

IBM Counter Fraud Management for Safer Payments

Version 5 Release 7

Application Programming Interface Reference



General Information about IBM Counter Fraud Management for Safer Payments Application Programming Interface: [click here](#)

1 ActivateKey	1
2 AddCaseActionAttachment	2
3 AddDefinedRiskEntry	3
4 AddTickerEntry	5
5 BulkDeleteDefinedRiskEntries	6
6 BulkEnableDisableDefinedRiskEntries	7
7 CanAccessAnyMasterdata	8
8 CancelGolive	9
9 CancelMasterKeyChange	10
10 CancelSimulationReport	11
11 CanChangeJobEncryptionSettings	12
12 ChangeMasterKey	13
13 ChangePassword	14
14 CheckCalendarComputations	15
15 CheckDataSelection	16
16 CheckIndex	18
17 CheckUserId	19
18 CleanoutRevisions	20
19 CloneSandboxRecord	21
20 CommitRule	22
21 CommitRuleCondition	23
22 CommitRules	24
23 CommitRuleScore	25
24 CommitRuleset	26

25 CompareRevisions	27
26 ComputeRuleGenerationDataSelection	29
27 ComputeSandboxRecords	30
28 ConfirmGolive	31
29 CopyMapping	32
30 CopyRevision	33
31 CountQuery	34
32 CreateCasesFromQuery	36
33 CreateDefinedRiskListEntryFromQuery	37
34 CreateDefinedRiskListImportSettings	38
35 CreateRulesScoring	39
36 DeactivatePrivateKeyPasswordSafe	41
37 Delete	42
38 DeleteAnalyses	43
39 DeleteAnalysis	44
40 DeleteDefinedRiskEntries	45
41 DeleteInstance	46
42 DeleteRulegenerationRules	47
43 DeleteTickerEntry	48
44 Detach	49
45 DisableAmi	50
46 DisableEci	51
47 DisableFli	52
48 DisableMci	53

49 DisableSimulation	54
50 EnableAmi	55
51 EnableApi	56
52 EnableBdi	57
53 EnableDisableCaseGroup	58
54 EnableDisableComplianceList	59
55 EnableDisableCpps	60
56 EnableDisableDefinedRiskEntries	61
57 EnableDisableDefinedRiskList	62
58 EnableEci	63
59 EnableFli	64
60 EnableMci	65
61 EnableSimulationCheckAndReport	66
62 EnableSimulationConfirmed	68
63 ExecuteBulkCaseTransition	69
64 ExecuteCaseTransition	70
65 ExecuteExternalQuery	71
66 ExecuteGroupByQuery	72
67 ExecuteJob	73
68 ExecuteQuery	74
69 ExecuteReport	76
70 ExecuteRuleQuery	77
71 ExecuteRuleReportQuery	78
72 FixMillisecondsMapping	79

73 GenerateRulesUndoRule	80
74 Get	81
75 GetActivatedComplianceLists	82
76 GetAdHocCheckComplianceListEntriesTable	83
77 GetAggregatedAuditTrails	85
78 GetAllProfilingsTable	86
79 GetAllRulesTable	87
80 GetAnalysedAttributes	88
81 GetAnalysisDataSelection	89
82 GetAnalysisProgress	91
83 GetAnalysisTable	92
84 GetAttributes	93
85 GetAuditLogTable	95
86 GetCaseAction	96
87 GetCaseActionPreview	98
88 GetCaseActions	99
89 GetCaseActionsFromCaseQueue	100
90 GetCaseActionsTable	101
91 GetCaseCloseCodes	102
92 GetCaseCloseCodesTable	103
93 GetCaseData	104
94 GetCaseGroupDefinitions	105
95 GetCaseGroupsTable	106
96 GetCaseHistory	107

97 GetCaseQueueReportsTable	109
98 GetCaseQueues	110
99 GetCaseQueuesTable	111
100 GetCasesTable	112
101 GetCaseStates	114
102 GetCaseStatesTable	115
103 GetCaseTransitions	116
104 GetCaseVariablesAsConditions	117
105 GetCaseWorkflows	118
106 GetCaseWorkflowsTable	119
107 GetChallengerStatus	120
108 GetChartColorSchema	121
109 GetCharts	122
110 GetChartsTable	123
111 GetCollusions	124
112 GetCollusionSimulationResults	125
113 GetCollusionsTable	126
114 GetCommonPointQueriesTable	127
115 GetCommonQuerySubsetResults	128
116 GetComplianceDataSourceExists	130
117 GetComplianceDefinitionTable	131
118 GetComplianceListDefinitions	132
119 GetConclusionExpressionValuePairs	133
120 GetCountersTable	134

121 GetCpp	135
122 GetCpps	136
123 GetCppTable	138
124 GetDashboardCharts	140
125 GetDashboardKeyPerformanceIndicatorData	141
126 GetDashboardKeyPerformanceIndicators	142
127 GetDefaultCapacities	143
128 GetDefaultMaskEncryptedValues	144
129 GetDefaults	145
130 GetDefinableCaseTransitions	147
131 GetDefinedRiskDefinitions	148
132 GetDefinedRiskListAuditTrailTable	149
133 GetDefinedRiskListDefinitionTable	150
134 GetDefinedRiskListEntriesTable	151
135 GetDefinedRiskListEntry	153
136 GetDefinedRiskListEntryAttribute	154
137 GetDefinedRiskListPrivileges	155
138 GetDefinedRiskListSettings	156
139 GetDeviceIdentificationTable	157
140 GetDonorInstances	158
141 GetElementPrintView	159
142 GetEnabledAnalyses	160
143 GetEventsTable	161
144 GetExecutables	162

145 GetExternalQueries	163
146 GetExternalQueriesTable	164
147 GetFastLinkStatusTable	165
148 GetFinalRulesets	166
149 GetFinalRulesetsTable	167
150 GetFittingRulesets	168
151 GetFollowupUsers	169
152 GetFormulasTable	170
153 GetFraudValues	171
154 GetGroupByQueriesTable	172
155 GetGroupByQueryAccountResults	173
156 GetGroupByQueryResult	174
157 GetGroupByQueryResultMetaInfo	176
158 GetGroupByQueryResults	177
159 GetIndexAspects	178
160 GetIndexAttribute	179
161 GetIndexBasedEvaluations	180
162 GetIndexBasedEvaluationsTable	181
163 GetIndexedAttributes	182
164 GetIndexes	183
165 GetIndexesTable	184
166 GetInputsTable	185
167 GetInstances	186
168 GetInstancesTable	187

169 GetInvestigationReportsTable	188
170 GetIsRulesetEnabled	189
171 GetKeyPerformanceIndicatorsTable	190
172 GetKeysTable	191
173 GetLastUrid	192
174 GetLastUserAction	193
175 GetListsTable	194
176 GetMandators	195
177 GetMandatorSelectionTable	197
178 GetMandatorSelectionTree	198
179 GetMandatorsTable	199
180 GetMappingsTable	200
181 GetMasterdataDeepInformation	201
182 GetMasterdatas	202
183 GetMasterKeysTable	203
184 GetMemoryManagementTable	204
185 GetMerchantMonitoringTemplate	205
186 GetMerchantMonitoringTemplates	206
187 GetMergingsTable	207
188 GetMessageReport	208
189 GetMessages	209
190 GetMessageReport	210
191 GetMessageTable	211
192 GetMetaAttributeAmount	212

193 GetMetaAttributeMaximum	213
194 GetMissedCasesReportsTable	214
195 GetModelComponents	215
196 GetModelComponentsTable	216
197 GetModelling	217
198 GetModellingsTable	219
199 GetMultipleRulesFiringStatistics	220
200 GetMyWorkingQueues	222
201 GetNextCase	223
202 GetNotifications	224
203 GetNotificationsTable	225
204 GetOptimizationAnalysis	226
205 GetOutgoingChannelConfigurations	227
206 GetOutgoingChannelConfigurationTable	228
207 GetOutputsTable	229
208 GetPasswordSafes	230
209 GetPasswordSafeTable	231
210 GetPatternsTable	232
211 GetPciDssReport	233
212 GetPossibleAmountAttributes	234
213 GetPrecedentsTable	235
214 GetPreference	236
215 GetPreProcessingRulesets	237
216 GetPreProcessingRulesetsTable	238

217 GetPrintView	239
218 GetProcessPriorities	240
219 GetProfiles	241
220 GetProfilesTable	242
221 GetPythonFunctions	243
222 GetQueries	244
223 GetQueryOptions	245
224 GetQueryResults	246
225 GetQuickSearchCasesTable	248
226 GetRealtimeInterceptionCodes	250
227 GetRealtimeInterceptionCodesTable	251
228 GetReferencesOfAttribute	252
229 GetReferencesOfCaseAction	253
230 GetReferencesOfCaseCloseCode	254
231 GetReferencesOfCaseQueue	255
232 GetReferencesOfCaseState	256
233 GetReferencesOfCaseWorkflow	257
234 GetReferencesOfChart	258
235 GetReferencesOfDefinedRiskList	259
236 GetReferencesOfExternalQuery	260
237 GetReferencesOfIndex	261
238 GetReferencesOfMasterdata	262
239 GetReferencesOfMessage	263
240 GetReferencesOfNotification	264

241 GetReferencesOfOutgoingChannelConfiguration	265
242 GetReferencesOfPasswordSafe	266
243 GetReferencesOfProfile	267
244 GetReferencesOfPythonModule	268
245 GetReferencesOfQuery	269
246 GetReferencesOfReminder	270
247 GetReferencesOfUserGroup	271
248 GetReminders	272
249 GetReminderTable	273
250 GetReportingAttributes	274
251 GetRevisionAuditTrail	275
252 GetRevisionGeneral	277
253 GetRevisions	279
254 GetRevisionSelectionTable	280
255 GetRevisionStatus	281
256 GetRoles	282
257 GetRolesTable	283
258 GetRuleGenerationAPrioriConditions	284
259 GetRuleGenerationDataSelection	285
260 GetRuleGenerationDesignTable	287
261 GetRuleGenerationExplore	288
262 GetRuleGenerationPerformance	289
263 GetRuleGenerationRules	291
264 GetRuleGenerationSettings	292

265 GetRuleGenerationStatus	293
266 GetRuleGenerationTargets	294
267 GetRulePerformance	295
268 GetRulePerformanceRulesets	297
269 GetRuleReport	298
270 GetRulesStatistics	300
271 GetRulesTable	301
272 GetRuleStatistics	302
273 GetRunningSimulations	303
274 GetSandboxTable	304
275 GetSearchComplianceListEntriesTable	305
276 GetSessionCountdownData	307
277 GetSettings	308
278 GetSimulationDataSelection	309
279 GetSimulationProgress	310
280 GetSimulationTable	311
281 GetStatistics	312
282 GetStatusAlarmIndicatorsDisplay	313
283 GetStatusAlarmIndicatorsTable	315
284 GetSystemInformation	316
285 GetSystemLogTable	317
286 GetTargetRevisions	318
287 GetTextModules	319
288 GetTextModulesTable	320

289 GetThreadPriorities	321
290 GetTickerEntries	322
291 GetTimestampAttributes	323
292 GetToAddressMeta	324
293 GetTransactionMessageReportTable	325
294 GetUserGroups	326
295 GetUserGroupsTable	327
296 GetUsers	328
297 GetUsersAndMandatorsOfRoles	329
298 GetUsersOfRole	330
299 GetUsersReportsTable	331
300 GetUsersTable	332
301 GetWorkingQueueCasesTable	333
302 GetWorkingQueues	335
303 GetWorkingQueuesTable	336
304 GetWorkingQueueStatistics	337
305 GoliveCheckAndReport	338
306 HasEntry	339
307 InterruptCase	340
308 InvestigateCase	341
309 IsCasesTableEncrypted	342
310 IsCppTableEncrypted	344
311 IsDdcEnabled	346
312 IsDdcEnabledForManualCaseCreation	347

313 IsDefinedRiskListEncrypted	348
314 IsGoliveRunning	349
315 IsGroupByQueryResultEncrypted	350
316 IsIndexSequenceMdcOutsideSimulation	351
317 IsMetaAttributeInChampions	352
318 IsOffline	353
319 IsPasswordValid	354
320 IsQueryResultEncrypted	355
321 IsRevisionDeletionEnabled	356
322 IsSamplingEnabled	357
323 IsSimulationValid	358
324 IsValidInstanceId	359
325 IsWorkingQueueCasesTableEncrypted	360
326 LoadCaseAuditTrailsFromDisk	361
327 Logout	362
328 MarkFraud	363
329 MoveRules	364
330 MultiDelete	365
331 ParentPathExists	366
332 PathExists	367
333 ReAttach	368
334 RebuildAllIndexes	369
335 RebuildIndex	370
336 RebuildIndexSequence	371

337 RecomputeAttributes	372
338 ReGoliveRevision	373
339 ReloadComplianceLists	374
340 ReloadPrivateKeys	375
341 ReloadPrivateKeysPasswordSafe	376
342 Reserve	377
343 ResetAllUserPreferences	378
344 ResetIndex	379
345 ResetOutgoingFli	380
346 ResetPreference	381
347 ResetSandboxRecords	382
348 ResetUserPreferences	383
349 RestartInstance	384
350 Restore	385
351 RetireChampion	386
352 RevertToChallenger	387
353 RevisionElementExists	388
354 RevokeKey	389
355 RewindFliBuffer	390
356 RuleReportQueryExists	391
357 Save	392
358 SaveCluster	394
359 SaveCpp	398
360 SaveSandboxEntry	400

361 SaveSettings	401
362 SendCaseActionFromPreview	420
363 SendCaseActions	422
364 Serialize	423
365 SetAllMyPreferencesAsDefault	424
366 SetAnalysis	425
367 SetAttachmentData	427
368 SetFinalRulesetsEnabled	428
369 SetKey	429
370 SetMasterdata	430
371 SetModelComponentsEnabled	431
372 SetModelling	432
373 SetMultiInterfaces	433
374 SetMultiModelling	434
375 SetOffline	435
376 SetOnline	436
377 SetPreference	437
378 SetQuickSearchCasesTablePreference	438
379 SetRuleGenerationCategoricIndicators	439
380 SetRuleGenerationDataSelection	440
381 SetRuleGenerationFollowingIndicators	442
382 SetRuleGenerationIndicator	443
383 SetRuleGenerationPrecedingIndicators	444
384 SetRuleGenerationSettings	445

385 SetRulesEnabled	447
386 SetSimulationDataSelection	448
387 SetUserExportPassword	450
388 SetXdcSizes	451
389 Shutdown	452
390 SimulateCollusion	453
391 StartAnalysis	454
392 StartAutomaticRuleGeneration	455
393 StopAnalysis	456
394 StopAutomaticRuleGeneration	457
395 StopCollusionSimulation	458
396 StopJob	459
397 StopQuery	460
398 StopRuleGeneration	461
399 TakeoverCase	462
400 TakeoverRevision	463
401 Test	464
402 TestLdapConnection	465
403 UnmarkFraud	466
404 Unreserve	467
405 UnreserveCase	468
406 UnreserveDefinedRiskListEntry	469
407 UpdateRiskListEntry	470

1 ActivateKey

Detailed Description

ActivateKey is used to activate a valid encryption key, and activate interfaces. To successfully activate, left and right key for the encryption key must have been provided by left and right user.

Parameters

HttpRequest: Holds the key uid and User object that have been passed to the server.

```
{
    "request": "activateKey",
    "uid": <uid>
}
```

- "uid": uid of the encryption key to be activated

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

2 AddCaseActionAttachment

Detailed Description

AddCaseActionAttachment is used to add an attachment to a case.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
    "request": "addCaseActionAttachment",
    "uid": <caseUid>
}
```

- "uid" (mandatory): Uid of the corresponding Case

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

3 AddDefinedRiskEntry

Detailed Description

AddDefinedRiskEntry is used to add an entry to a defined risk list

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "addDefinedRiskEntry",
  "uid": <definedRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "value": "<inputAttributeValue>",
    "outputAttributeCategoryValue": "<outputAttributeCategoryValue>" |
"undefined",
    "expiresAt": <expiresAt>,
    "startsAt": <startsAt>,
    "id": <entryId>,
    "comment": "<comment>",
    "label": "<label>",
    "active": true|false,
    "conditions": [<conditions>]
  }
}
```

- "uid": The uid of the risk list, in which the entry should be added
- "mandator": The uid of mandator, for which the risk list is defined
- "data:value": The value of the entry
- "data:outputAttributeCategoryValue": if a category is defined for the attribute the category value has to be send
- "data:expiresAt": If the entry should expire, the date can be assigned here
- "data:startsAt": If the entry should only be computed after a defined date, the date can be assigned here
- "data:id": The id of the entry in the list
- "data:comment": A comment for this entry can be assigned
- "data:label": The id of the entry in the list
- "data:active": Defines if the entry should be active or inactive
- "data:conditions": Additional conditions which have to be met for this entry to be applied in computation

Returns

A HTTP response with the following content:

```
{
  "comparisonResult": "Unchanged" | "ChangedNoImpact" | "ChangedComputationImpact",
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


4 AddTickerEntry

Detailed Description

AddTickerEntry is used to add entries to the ticker for the defined mandators

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "addTickerEntry",
  "data": {
    "userId": <userId>,
    "title": "<tickerTitle>",
    "note": "<tickerNote>",
    "mandatorUid": <mandatorUid>
  }
}
```

- "data:userId": The uid of the user, who adds the entry
- "data:title": Title of the entry
- "data:note": Text of the entry
- "data:mandatorUid": List of uids of mandators, for which the entry should be shown

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

5 BulkDeleteDefinedRiskEntries

Detailed Description

BulkDeleteDefinedRiskEntries deletes a list of risk list entries

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "bulkDeleteDefinedRiskEntries",
  "uid": <definedRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "items": [
      {
        "attributeValue": "<attributeValue>",
        "id": <entryId>
      },
      ... ..
    ]
  }
}
```

- "uid": The uid of the risk list, from which the entries should be deleted
- "mandator": Mandator identified by uid, where the risk list is defined.
- "data:items": List of items, to delete.
- "data:items:attributeValue": attributeValue of the entry (used to identify entry)
- "data:items:id": id of the entry (used to identify entry)

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

6 BulkEnableDisableDefinedRiskEntries

Detailed Description

BulkEnableDisableDefinedRiskEntries is used to enable/disable risk list entries

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "bulkEnableDisableDefinedRiskEntries",
  "uid": <defineRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "active": true|false,
    "items": [
      {
        "attributeValue": "<inputAttributeValue>",
        "id": <entryId>
      },
      ... ...
    ]
  }
}
```

- "uid": The uid of the risk list, from which the entries should be enabled/disabled
- "mandator": Mandator identified by uid, where the risk list is defined.
- "data:active": true if the entries should be enabled, false to disable
- "data:items": List of items, to enable/disable.
- "data:items:attributeValue": attributeValue of the entry (used to identify entry)
- "data:items:id": id of the entry (used to identify entry)

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

7 CanAccessAnyMasterdata

Detailed Description

CanAccessAnyMasterdata is used to determine if a user may access masterdata

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{  
    "request": "canAccessAnyMasterdata"  
}
```

Returns

A HTTP response with the following content:

```
{  
    "canAccessAnyMasterdata": true|false,  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

8 CancelGolive

Detailed Description

CancelGolive cancels a golive, when it is waiting for confirmation on the golive report.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "cancelGolive",
    "revision": <revisionUid>
}
```

- "revision": The revision (identified by uid) for which the golive report should be canceled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

9 CancelMasterKeyChange

Detailed Description

CancelMasterKeyChange cancels the ongoing change of the master key. Will invalidate the instance if that option was set in the request, otherwise instance status will be set to ok afterwards.

Parameters

HttpRequest: Holds the parameter invalidateInstance uid and user object.

```
{
  "request": "cancelMasterKeyChange",
  "data": {
    "invalidateInstance": <true/false>
  }
}
```

- "data:invalidateInstance": indicates if the instance status is "INVALIDATED" or "OK" after master key change

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

10 CancelSimulationReport

Detailed Description

CancelSimulationReport cancels a simulation report, before simulation is started/confirmed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "cancelSimulationReport",
    "revision": <revisionUid>
}
```

- "revision": The revision (identified by uid) for which the simulation report should be canceled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

11 CanChangeJobEncryptionSettings

Detailed Description

CanChangeJobEncryptionSettings checks if a user may change the encryption settings of a loadfile job.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "canChangeJobEncryptionSettings"
}
```

Returns

A HTTP response with the following content:

```
{
    "canChangeJobEncryptionSettings": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


12 ChangeMasterKey

Detailed Description

ChangeMasterKey changes the current master key. It triggers decryption and encryption of all data on all Safer Payments instances.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "changeMasterKey",
    "uid": <masterKeyUid>
}
```

- "uid" (mandatory): The new MasterKey (identified by its uid).

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

13 ChangePassword

Detailed Description

ChangePassword is used to change the user password.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "changePassword",
  "data": {
    "oldPassword": "<oldPassword>",
    "newPassword": "<newPassword>"
  }
}
```

- "data:oldPassword" (mandatory): The old password which should be changed
- "data:newPassword" (mandatory): The new password

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

14 CheckCalendarComputations

Detailed Description

CheckCalendarComputations checks index based evaluation calendar computation definitions and returns them as pseudo attributes. The server builds the calendar computations and checks them against the current champion Revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "checkCalendarComputations",
    "mandator": <revisionUid>
    "data": {
        "indexUid": <indexUid>,
        "calendarComputations": [
            ...
        ]
    }
}
```

- "mandator" (mandatory): The Mandator (identified by its uid) the calendar computations belong to.
- "data" (mandatory): The calendar computations to check.
 - "data:indexUid" (mandatory): The Index (identified by its uid) the selected calendar uses.
 - "data:calendarComputations" (mandatory): Calendar computation data streams.

Returns

A HTTP response with the following content:

```
{
    "attributes": [
        ...
    ],
    "responseStatus": [<status>, "<feedbackText>"],
    "reloadUserProfile": true|false
}
```

- "attributes": The calendar computations written like attributes to be usable by the conditions client-code.

15 CheckDataSelection

Detailed Description

CheckDataSelection is used to check if the data selection is valid. It returns minimumUrid, maximumUrid, minimumSystemtime and maximumSystemtime for the selection. So the user can see the boundaries for a possible selection.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "checkDataSelection",
  "type": "<selectionType>",
  "revision": <revisionUid>
  "data": {
    "mandators": [<mandatorUid>],
    "periodType": "RecordsAbsolute" | "RecordsRelative" | "TimeAbsolute" |
    "TimeRelative",
    "desired": {
      "fromUrid": <fromUrid>,
      "toUrid": <toUrid>,
      "fromRecord": <fromRecord>,
      "toRecord": <toRecord>,
      "fromTimestamp": <fromTimestamp>,
      "toTimestamp": <toTimestamp>,
      "fromDays": <fromDays>,
      "toDays": <toDays>
    }
  }
}
```

- "type": The type of the selection
- "revision": The revision (identified by uid) in which the data selection is defined
- "data:mandators": A list of mandators (identified by uid) which records should be included in the selection
- "data:periodType": How the records are selected (by time, recordId or relative by count)
- "data:desired": The desired selection, defined by following params
- "data:desired:fromUrid": starting Urid of selection (for absolute selection)
- "data:desired:toUrid": last Urid of selection (for absolute selection)
- "data:desired:fromRecord": starting number of selection (for relative selection)
- "data:desired:toRecord": last record number of selection (for relative selection)
- "data:desired:fromTimestamp": starting timestamp of selection (for absolute time)
- "data:desired:toTimestamp": end timestamp of selection (for absolute time)
- "data:desired:fromDays": How many days back records are included for selection (for relative time)
- "data:desired:toDays": How many days back records are excluded for selection (for relative time)

Returns

A HTTP response with the following content:

```
{
  "dataSelection": {
    <dataSelection>
  },
  "minimumUrid": <minimum urid>,
  "maximumUrid": <maximum urid>,
  "minimumSystemtime": <minimum systemtime>,
  "maximumSystemtime": <maximum systemtime>,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

16 CheckIndex

Detailed Description

CheckIndex checks the healthiness of an index. During the check, all other access to this index will be blocked and no transactions can be processed. The MCI, FLI and BDI interfaces should be disabled on this instance. The current user should have the global privilege of the maintenance function "Check health of index".

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "checkIndex",
    "uid": <indexUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): The uid of an index to be checked.
- "mandator" (mandatory): The mandator of an index.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

17 CheckUserId

Detailed Description

CheckUserId check if the userId in this request fits with the current user session. If the "userId" is set to a value other than the current user's UID, "reloadUserProfile" returns true.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "checkUserId",
    "data": {
        "userId": <userId>
    }
}
```

- "data" (mandatory):
- "data:userId" (mandatory): The user that has to be checked.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

18 CleanoutRevisions

Detailed Description

CleanoutRevisions is used to unload all non-champion revisions from IBM Safer Payments. It moves the file representations of all non-champion revisions from the "cfg" folder to the "arc" folder in all IBM Safer Payments instances. All changes in model and system configuration should be disabled while running this request on all IBM Safer Payments instances. It is thus recommended to disable the API interface on all IBM Safer Payments instances when run this request.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "cleanoutRevisions"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

Remarks

19 CloneSandboxRecord

Detailed Description

CloneSandboxRecord is used to clone one record in the sandbox. It returns the UID of the new record. Function checks, if revision is reserved for user, to apply changes.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "cloneSandboxRecord",
    "uid": <sourceSandboxRecordUid>,
    "revision": <revisionUid>,
    "data": {
        "name": "<targetSandboxRecordName>"
    }
}
```

- "uid": uid of record to be cloned.
- "revision": A Revision (identified by its uid) where the record will be created.
- "data:name": The name of the new sandbox record

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

20 CommitRule

Detailed Description

CommitRule is used to create a new rule with proposed conditions. After first creation, additional conditions can be added via CommitRuleCondition.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "commitRule",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): A Revision (identified by its uid) in which the rule will be created.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

21 CommitRuleCondition

Detailed Description

CommitRuleCondition is used in rule generator to add a proposed condition to a rule. If there is already a rule generated in rule generator, additional conditions can be added with this function.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "commitRuleCondition",
    "uid": <attributeUid>,
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): the uid of proposed condition, which should be added.
- "mandator" (mandatory): A Mandator (identified by its uid) in which the rule will be created.
- "revision" (mandatory): A Revision (identified by its uid) in which the rule will be created.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

22 CommitRules

Detailed Description

CommitRules is used in rule generator to add a generated rules to a ruleset. After generation of rules, one or more rules can be added with this function to an existing or new ruleset. This will also stop the rule generator.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "commitRules",
    "revision": <revisionUid>,
    "ruleset": <rulesetUid>,
    "data": {
        "rules": [<ruleUid>, ...]
    }
}
```

- "revision" (mandatory): A Revision (identified by its uid) in which the rule will be created.
- "ruleset" (optional): A ruleset (identified by its uid) where the rules should be added.
- "rules" (mandatory): In data, is a list of uids representing the rules to be added.
- "rulesetData" (optional if param ruleset is defined): all data necessary to create a new ruleset, where rules should be added.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

23 CommitRuleScore

Detailed Description

CommitRuleScore is used in analysis rule score to commit the proposed conclusion changes to the rules. After rule scoring, all or a subset of rules can be changed by applying the proposed score changes to their conclusions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "commitRuleScore",
  "revision": <revisionUid>,
  "data": {
    "allRules": true|false,
    "ruleUids": [<ruleUid>, ...]
  }
}
```

- "revision" (mandatory): A Revision (identified by its uid) the rules are defined in.
- "data:allRules" (mandatory): Indicates whether all rules that were scored should be committed or only a subset (see data:rules).
- "data:rules" (optional): If not all rules that were scored should be committed, a list of uids representing the rules to be committed.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

24 CommitRuleset

Detailed Description

CommitRuleset adds a new ruleset to a revision. After rule generation, this function is used to add all generated rules to a new or existing ruleset. This will also stop the rule generator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "commitRuleset",
    "revision": <revisionUid>,
    "ruleset": <rulesetUid>
}
```

- "revision" (mandatory): A Revision (identified by its uid) in which the ruleset will be created.
- "ruleset" (optional): A ruleset (identified by its uid) where the rules should be added.
- "data" (optional if param ruleset is defined): all data necessary to create a new ruleset, where rules should be added.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

25 CompareRevisions

Detailed Description

CompareRevisions is used to compare two revisions of the same mandator. It returns the comparison results as json. For every element, which is different, an entry is created to show the difference in comparisonResult. The changes are shown in comparisonResult:changes.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "compareRevisions",
  "uid": <sourceRevisionUid>,
  "revision": <targetRevisionUid>
}
```

- "uid" (mandatory): Source revision (identified by its uid) for the comparison.
- "revision" (mandatory): Target revision (identified by its uid) for the comparison.

Returns

A HTTP response with the following content:

```
{
  "showUnchangedElements": "<showUnchangedElements>",
  "source": "[<sourceRevisionId> ' <sourceRevisionName>' ",
  "target": "[<targetRevisionId> ' <targetRevisionName>' ",
  "sourceRevisionUid": <sourceRevisionUid>,
  "targetRevisionUid": <targetRevisionUid>,
  "sourceStatus": "<sourceRevisionStatus>",
  "targetStatus": "<targetRevisionStatus>",
  "sourceUser": "<sourceRevisionCurrentlyBeingEditedByUserName>",
  "targetUser": "<targetRevisionCurrentlyBeingEditedByUserName>",
  "comparisonResult": [{
    "listValueInAudittrailEncrypted": true|false,
    "notificationEntry": true|false,
    "userName": "<userName>",
    "userId": <userId>,
    "utc": <utcTimestamp>,
    "changedElementName": "<changedElementName>",
    "changedElementUid": <changedElementUid>,
    "revisionNumber": <revisionNumber>,
    "revisionName": "<revisionName>",
    "uid": <uid>,
    "comparisonResult": "<comparisonResult>",
    "type": "<type>",
```

```
    "name": "<name>",
    "origin": "<origin>",
    "changes": [{
      "type": "name",
      "status": "Unchanged",
      "needsEncryption": false,
      "sourceValue": "Primary Record ID",
      "targetValue": "Primary Record ID"
    },
    ... ..
  ],
  ... ..
],
"responseStatus": [<status>, "<feedbacktext>"],
"reloadUserProfile": true|false
}
```


26 ComputeRuleGenerationDataSelection

Detailed Description

ComputeRuleGenerationDataSelection start computation of rule generation data selection. It initializes the rule generator, sets dataselection and starts the generation process.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "computeRuleGenerationDataSelection",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) in which the rule generation should run.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

27 ComputeSandboxRecords

Detailed Description

ComputeSandboxRecords is used to start the output computation for all defined sandbox records.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "computeRuleGenerationDataSelection",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) in which the sandbox computation should run.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

28 ConfirmGolive

Detailed Description

ConfirmGolive is used to confirm the information stated in the golive report and with this starting the golive process.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "confirmGolive",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): A Revision (identified by its uid) which should execute golive.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

29 CopyMapping

Detailed Description

CopyMapping copies an attribute mapping from the source message to the target message.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "copyMapping",
  "revision": <revisionUid>,
  "data": {
    "messageUid": <sourceMessageUid>,
    "targetMessageUid": <targetMessageUid>,
    "attributeUid": <attributeUid>
  }
}
```

- "revision" (mandatory): Identifies the revision in which the mapping should be copied
- "data:messageUid" (mandatory): Identifies the source message from where the mapping should be copied
- "data:targetMessageUid" (mandatory): Identifies the target message where the mapping should be copied to
- "data:attributeUid" (mandatory): Identifies the attribute for which the mapping should be copied

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

30 CopyRevision

Detailed Description

CopyRevision is used to create a new challenger based on an existing revision of the same mandator

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "copyRevision",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): Identifies the revision which should be copied to create a new challenger

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

31 CountQuery

Detailed Description

CountQuery is used to count the records from a query result, without computing the whole result table.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "countQuery",
  "uid": <queryUid>,
  "type": "query" | "simulationQuery",
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "parameter": "<parameter>"
  }
}
```

- "uid" (mandatory): Identifies the query which should be executed
- "type" (mandatory): defines what kind of query it is. ["query" | "simulationQuery"]
- "mandator" (mandatory): Identifies the mandator for which the query is defined
- "revision" (optional): Identifies the revision in which the simulationQuery should be executed
- "data:parameter" (optional): additional parameter for hyperlink or index queries (holds the index value to query) to create a new challenger

Returns

A HTTP response with the following content:

```
{
  "amountDecimals": "<amountDecimals>",
  "amountDimension": "<amountDimension>",
  "statistics": {
    "transaction": {
      "genuine": {
        "records": <genuineRecords>,
        "amount": <amount>
      },
      "fraud": {
        "records": <fraudRecords>,
        "amount": <amount>
      }
    }
  }
}
```

```
    },  
    "queryUid": <queryUid>,  
    "recordsHit": <recordsHit>,  
    "parameter": "<parameter>",  
    "duration": <duration>,  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
  }  
}
```

- "statistics:transactions": shows how many records are genuine/fraud and how big the amount is for each

32 CreateCasesFromQuery

Detailed Description

CreateCasesFromQuery is used to create cases from a query result.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "createCasesFromQuery",
  "mandator": <mandatorUid>,
  "data": {
    "caseQueue": <caseQueueUid>,
    "urids": [<urid>, ...],
    "caseScore": <caseScore>,
    "memo": "<memo>",
    "cpp": "<cpp>" | "-1",
    "includeDdc": true|false
  }
}
```

- "mandator" (mandatory): The owning mandator for the newly created case.
- "data" (mandatory):
- "data:caseQueue": Cases will be created with this case queue.
- "data:urids": A list of unique record ids for which cases should be created.
- "data:caseScore": A numeric value indicating the score of a case.
- "data:memo": An short explanation or comment on creating cases.
- "data:cpp": If selected, cases will be assigned to this CPP.
- "data:includeDdc": If enabled, the DDC data will be available for the case creation.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


33 CreateDefinedRiskListEntryFromQuery

Detailed Description

CreateDefinedRiskListEntryFromQuery is used to create risk list entry from a query result.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "createDefinedRiskListEntryFromQuery",
  "uid": <definedRiskListUid>,
  "data": {
    "urid": [<queryResultRecordUrid>],
    "outputAttributeCategoryValue": "<outputAttributeCategoryValue>" |
"undefined",
    "expiresAt": <expiresAtTimestamp>,
    "startsAt": <startsAtTimestamp>,
    "comment": "<comment>",
    "label": "<label>",
    "active": true|false,
    "conditions": [ ... ]
  }
}
```

- "uid": The uid of a risk list for which a new entry should be created.
- "data":
 - "data:urid": Unique record id for which risk list entry will be created.
 - "data:outputAttributeCategoryValue": The value of output attribute.
 - "data:expiresAt": Expiry date for the risk list entry.
 - "data:startsAt": Can be used to define risk list entry for future use; only records past this date are checked against this risk list.
 - "data:comment": A comment to be added to newly created risk list.
 - "data:label": A label to be added to newly created risk list.
 - "data:active": If active, risk list will be enabled for rule actions.
 - "data:conditions": The risk list will be applied to transaction records satisfying these conditions.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

34 CreateDefinedRiskListImportSettings

Detailed Description

CreateDefinedRiskListImportSettings is used to create settings for defined risk list import.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "createDefinedRiskListImportSettings",
  "uid": <definedRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "overwrite": true|false,
    "value": "-1",
    "outputAttributeCategoryValue": "<outputAttributeCategoryValue>" |
"undefined",
    "expiresAt": <expiresAtTimestamp>,
    "startsAt": <startsAtTimestamp>,
    "id": -1,
    "comment": "<comment>",
    "label": "<label>",
    "active": true|false,
    "conditions": [...]
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "id": <id>,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

35 CreateRulesScoring

Detailed Description

CreateRulesScoring is used to create rules scoring.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "createRulesScoring",
  "revision": <revisionUid>,
  "data": {
    "analysisId": <analysisId>,
    "rulesets": [<rulesetUid>, ...],
    "scoreType": "Standard" | "SAPFA",
    "scoreConclusionAttribute": <scoreConclusionAttributeUid>
  }
}
```

- "revision" (mandatory): The revision for which rule scoring should be created.
- "data" (mandatory):
 - "data:analysisId" (mandatory): The analysis for which rule scoring should be created.
 - "data:rulesets" (mandatory): For each ruleset supplied a rule scoring will be created.
 - "data:scoreType" (optional): The type of the rule scoring, by default "Standard" is used.
 - "data:scoreConclusionAttribute": The conclusion attribute.

Returns

A HTTP response with the following content:

```
{
  "ruleScoring": [
    [
      <ruleUid>,
      <rulesetUid>,
      "<ruleName> (<rulesetName>)",
      <hitRate>,
      <falseAlarmRatio>,
      <savedAmountPerFalseAlarm>,
      <fraudAmountIntercepted>,
      <fraudTransactionsIntercepted>,
      <genuineAmountIntercepted>,
      <genuineTransactionsIntercepted>,
    ]
  ]
}
```

```
        "<rulesetName>",
        "<ruleName>",
        "<scoring>"
    ],
    "... .."
],
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

36 DeactivatePrivateKeyPasswordSafe

Detailed Description

DeactivatePrivateKeyPasswordSafe deactivates private key password safe. The user should have "change" privilege of "Password safes".

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "deactivatePrivateKeyPasswordSafe",
    "uid": <privateKeyPasswordSafeUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

37 Delete

Detailed Description

Delete is used to delete a configuration element.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "delete",
    "uid": <elementUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The uid of an element that has to be deleted.
- "revision": For revision elements, the owning revision has to be supplied. This variable is required if the element to delete is a revision element.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

38 DeleteAnalyses

Detailed Description

DeleteAnalyses deletes a list of analyses.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "deleteAnalyses",
  "revision": <revisionUid>,
  "data": {
    "analysisId": [<analysisId>, ...]
  }
}
```

- "revision" (mandatory): A Revision (identified by its uid) in which the analysis should be deleted.
- "data:analysisId" (mandatory): A list of uid representing analysis to be deleted.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

39 DeleteAnalysis

Detailed Description

DeleteAnalysis is used to delete a single analysis.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "deleteAnalysis",
    "uid": <analysisId>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The uid of the analysis to be deleted.
- "revision" (mandatory): The uid of the revision, from which the analysis should be deleted.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


40 DeleteDefinedRiskEntries

Detailed Description

DeleteDefinedRiskEntries is used to delete entries from a defined risk list. It can be used to delete a single entry or a list of entries of a risk list.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "deleteDefinedRiskEntries",
    "uid": <definedRiskListUid>,
    "mandator": <mandatorUid>,
    "data": {
        "items": [
            {
                "attributeValue": "<attributeValue>",
                "id": <entryId>
            },
            ... ..
        ]
    }
}
```

- "uid": The uid of the risk list, from which the entries should be deleted
- "mandator": Mandator identified by uid, where the risk list is defined.
- "data:items": List of items, to delete.
- "data:items:attributeValue": attributeValue of the entry (used to identify entry)
- "data:items:id": id of the entry (used to identify entry)

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

41 DeleteInstance

Detailed Description

DeleteInstance detaches FLI and SCI, removes the instance from the cluster and shuts it down.

Parameters

HttpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "deleteInstance",
    "uid": <instanceId>
}
```

- "uid" (mandatory): uid of the instance, which is to be deleted.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

42 DeleteRulegenerationRules

Detailed Description

DeleteRulegenerationRules is used to delete a generated rule from the rule generator. When rule generator created/proposed some rules, this function can be used to remove these from the proposed rules. This gives the opportunity to remove some rules, before creating them in a ruleset.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "deleteRulegenerationRules",
    "revision": <revisionUid>,
    "data": {
        "rules": [<ruleUid>, ...]
    }
}
```

- "revision": A revision, identified by uid, where rulegeneration is executed
- "data:rules": List of rules, to delete from generator.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

43 DeleteTickerEntry

Detailed Description

DeleteTickerEntry deletes a ticker entry. With this function a ticker entry can be deleted for all users.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "deleteTickerEntry",
    "data": {
        "entryId": <entryId>
    }
}
```

- "data:entryId": uid of the entry, which should be deleted.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

44 Detach

Detailed Description

Detach is used to detach an instance from the other instances. It stops the SCI and ECI.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "detach",
    "uid": <instanceUid>
}
```

- "uid" (mandatory): The uid of the instance, which should be detached from the cluster.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

45 DisableAmi

Detailed Description

DisableAmi disables alert message interface for instance.

Parameters

HttpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "disableAmi",
    "uid": <instanceId>
}
```

- "uid" (mandatory): uid of the instance, on which the AMI should be disabled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

46 DisableEci

Detailed Description

DisableEci disables encrypted communication interface for instance.

Parameters

HttpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "disableEci",
    "uid": <instanceId>
}
```

- "uid" (mandatory): uid of the instance, on which the ECI should be disabled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

47 DisableFli

Detailed Description

DisableFli disables fast link interface for instance.

Parameters

httpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "disableFli",
    "uid": <instanceId>
}
```

- "uid" (mandatory): uid of the instance, on which the FLI should be disabled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


48 DisableMci

Detailed Description

DisableMci disables message command interface for instance.

Parameters

HttpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "disableMci",
    "uid": <instanceId>
}
```

- "uid" (mandatory): uid of the instance, on which the MCI should be disabled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

49 DisableSimulation

Detailed Description

DisableSimulation stops a running simulation. Afterwards, already simulated values of attributes that only exist within challenger revision are not available anymore.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "disableSimulation",
    "data": {
        "revisions": [<revisionUids>]
    }
}
or
{
    "request": "disableSimulation",
    "revision": <revisionUids>
}
```

- "revision" (optional): uid for revision where simulation should be stopped
- "data:revisions" (optional): uids for revisions where simulation should be stopped
- "user" (mandatory): uid for user requesting the command. Used for privilege checks

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

50 EnableAmi

Detailed Description

EnableAmi enables alert message interface for instance.

Parameters

httpRequest: Holds the request variables that have been passed to the server

```
{  
    "request": "enableAmi",  
    "uid": <instanceId>  
}
```

- "uid" (mandatory): uid of the instance, on which the AMI should be enabled

Returns

A HTTP response with the following content:

```
{  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

51 EnableApi

Detailed Description

EnableApi enables application programming interface for instance.

Parameters

HttpRequest: Holds the request variables that have been passed to the server

```
{  
    "request": "enableApi",  
    "uid": <instanceId>  
}
```

- "uid" (mandatory): uid of the instance, on which the API should be enabled

Returns

A HTTP response with the following content:

```
{  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

52 EnableBdi

Detailed Description

EnableBdi enables batch data interface for instance.

Parameters

httpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "enableBdi",
    "uid": <instanceId>
}
```

- "uid" (mandatory): uid of the instance, on which the BDI should be enabled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

53 EnableDisableCaseGroup

Detailed Description

EnableDisableCaseGroup enables or disables a case group. CPPs cannot be assigned to disabled case groups.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "enableDisableCaseGroups",
    "uid": <caseGroupUid>,
    "mandator": <mandatorUid>,
    "data": {
        "activate": true|false
    }
}
```

- "uid" (mandatory): uid of the case group that should be enabled / disabled
- "mandator" (mandatory): uid of the mandator of the case group
- "data:activate" (mandatory): true if enable, false if disable

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

54 EnableDisableComplianceList

Detailed Description

EnableDisableComplianceList enables or disables a compliance list.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "enableDisableComplianceList",
    "uid": <complianceListUid>,
    "mandator": <mandatorUid>,
    "data": {
        "activate": true|false
    }
}
```

- "uid" (mandatory): uid of the compliance list that should be enabled / disabled
- "mandator" (mandatory): uid of the mandator of the case group
- "data:activate" (mandatory): true if enable, false if disable

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

55 EnableDisableCpps

Detailed Description

EnableDisableCpps Enables or disables a cpp (common point of purchase) of a case group.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "enableDisableCpps",
    "uid": <caseGroupUid>,
    "mandator": <mandatorUid>,
    "data": {
        "activate": true|false
        "id": <cppId>
    }
}
```

- "uid" (mandatory): uid of the case group that owns the cpp.
- "mandator" (mandatory): uid of the mandator of the case group
- "data:activate" (mandatory): true if enable, false if disable
- "data:id" (mandatory): id of the cpp that should be enabled / disabled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


56 EnableDisableDefinedRiskEntries

Detailed Description

EnableDisableDefinedRiskEntries enables or disables defined risk list entries for a defined risk list.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "enableDisableDefinedRiskListEntries",
  "uid": <definedRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "activate": true|false,
    "items": [<attributeValue>, <id>]
  }
}
```

- "uid" (mandatory): uid of the defined risk list that owns the entries
- "mandator" (mandatory): uid of the mandator of the case group
- "data:activate" (mandatory): true if enable, false if disable
- "data:items" (mandatory): holds attribute value and id of defined risk list entry that should be enabled / disabled

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

57 EnableDisableDefinedRiskList

Detailed Description

EnableDisableDefinedRiskList enabled or disables a defined risk list.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "enableDisableDefinedRiskList",
    "uid": <definedRiskListUid>,
    "mandator": <mandatorUid>,
    "data": {
        "activate": true|false
    }
}
```

- "uid" (mandatory): uid of the defined risk list that should be enabled / disabled
- "mandator" (mandatory): uid of the mandator of the case group
- "data:activate" (mandatory): true if enable, false if disable

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

58 EnableEci

Detailed Description

EnableEci enables encrypted communication interface for instance.

Parameters

httpRequest: Holds the request variables that have been passed to the server

```
{  
    "request": "enableEci",  
    "uid": <instanceId>  
}
```

- "uid" (mandatory): uid of the instance, on which the ECI should be enabled

Returns

A HTTP response with the following content:

```
{  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

59 EnableFli

Detailed Description

EnableFli enables fast link interface for instance.

Parameters

httpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "enableFli",
    "uid": <instanceId>
}
```

- "uid" (mandatory): uid of the instance, on which the FLI should be enabled

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

60 EnableMci

Detailed Description

EnableMic enables message command interface for instance.

Parameters

HttpRequest: Holds the request variables that have been passed to the server

```
{  
    "request": "enableMci",  
    "uid": <instanceId>  
}
```

- "uid" (mandatory): uid of the instance, on which the MCI should be enabled

Returns

A HTTP response with the following content:

```
{  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

61 EnableSimulationCheckAndReport

Detailed Description

EnableSimulationCheckAndReport requests a simulation report before actual simulation can be started. The report contains information about elements that will be simulated when simulation is started.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "enableSimulationCheckAndReport",
  "user": <userId>,
  "mandator": <mandatorId>,
  "revision": <revisionId>,
  "data": {
    "ruleId": <ruleId>
  }
}
```

- "user" (mandatory): uid of user who requests the simulation report. Privileges will be checked
- "mandator" (mandatory): uid of mandator that owns the revision
- "revision" (mandatory): uid of challenger revision for which the simulation report should be generated
- "data:ruleId" (optional): uid of rule for which a rule report should be started (only when started from rule)

Returns

A HTTP response with the following content:

```
{
  "simulationReport": [{
    "type": "simulationReportSimulateAttribute",
    "variables": ["<elementType>", "<elementName>", <value>]
  }, {
    "type": "simulationReportSimulate",
    "variables": ["<elementType>", "<elementName>", <value>]
  }, {
    "type": "simulationReportJustUseInput",
    "variables": ["<elementType>", "<elementName>"]
  }, {
    "type": "simulationReportJustUse",
    "variables": ["<elementType>", "<elementName>", <value>]
  }
],
```

```
"memoryRequired": <value>,  
"memoryAvailable": <value>,  
"userMemoryAvailable": true|false,  
"mandatorMemoryAvailable": true|false,  
"systemMemoryAvailable": true|false,  
"simulatedRecords": <value>,  
"userMemoryDefined": <value>,  
"canStartSimulation": true|false,  
"responseStatus": ["<status>", "<feedbacktext>"],  
"reloadUserProfile": true|false  
}
```

62 EnableSimulationConfirmed

Detailed Description

EnableSimulationConfirmed requests to start simulation for a challenger revision. It simulates the model in a revision for a data selection or simulates a rule.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "enableSimulationConfirmed",
    "revision": <revisionUid>,
    "data": {
        "ruleUid": <ruleUid>
    }
}
```

- "revision" (mandatory): uid of challenger revision for which the simulation should be started
- "data:ruleUid" (optional): uid of rule which should be simulated (only when started from rule)

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


63 ExecuteBulkCaseTransition

Detailed Description

ExecuteBulkCaseTransition executes a selected case transition on a number of selected case Uids, with the given case transition, justification, comment and case close code.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "executeBulkCaseTransition",
  "data": {
    "cases": [<caseUid>, <caseUid>, ...],
    "caseTransition": <caseTransitionUid>,
    "justification": "<justificationCodes>",
    "comment": "<comment>",
    "caseCloseCode": <caseCloseCodeUid>
  },
  "type": "<type>"
}
```

- "cases" (optional): An array of cases upon which transition should be performed (identified by their case uids)
- "caseTransition" (mandatory): The transition which should be performed on all the cases (identified by its uid)
- "justification" (optional): The justification (as a string) which will be used for the case transition (note that the justification is mandatory if the case transition requires it)
- "comment" (optional): A description which will be used for the transition (note that the comment is mandatory if the case transition requires it)
- "caseCloseCode" (optional): The case close code (identified by its uid) which will be used for the cases if the case transitions to a closed state (note that the caseCloseCode is mandatory if the case transitions to a closed state)
- "type" (optional): If the user wants to interrupt/intercept a case from another user(s) then the type needs to be set to "interrupt" in which case none of the other parameters except the cases array are used

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

64 ExecuteCaseTransition

Detailed Description

ExecuteCaseTransition executes a case transition for a single case.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "executeCaseTransition",
  "uid": <caseUid>,
  "data": {
    "caseTransition": <caseTransitionUid>,
    "justification": "<justificationCodes>",
    "comment": "<comment>",
    "caseCloseCode": <caseCloseCodeUid>
  }
}
```

- "uid" (mandatory): The case uid which should be transitioned
- "caseTransition" (mandatory): The transition which should be performed on the case (identified by its uid)
- "justification" (optional): The justification (as a string) which will be used for the case transition (note that the justification is mandatory if the case transition requires it)
- "comment" (optional): A description which will be used for the transition (note that the comment is mandatory if the case transition requires it)
- "caseCloseCode" (optional): The case close code (identified by its uid) which will be used for the cases if the case transitions to a closed state (note that the caseCloseCode is mandatory if the case transitions to a closed state)

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

65 ExecuteExternalQuery

Detailed Description

ExecuteExternalQuery is used to execute an external from a case. It returns an id to get the results with.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "executeExternalQuery",
  "uid": <externalQueryUid>,
  "data": {
    "caseUid": <caseUid>
  }
}
```

- "uid" (mandatory): The uid of the external query to be executed
- "data:caseUid" (mandatory): The uid of the case, from which the external query is executed

Returns

A HTTP response with the following content:

```
{
  "externalQuery": "<externalQueryName>",
  "values": [ ... ],
  "overwriteValues": 0|1,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

66 ExecuteGroupByQuery

Detailed Description

ExecuteGroupByQuery executes a groupByQuery. It returns an id to get the results with.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "executeGroupByQuery",
    "uid": <groupByQueryUid>
}
```

- "uid" (mandatory): The uid of the groupByQuery to be executed

Returns

A HTTP response with the following content:

```
{
    "resultId": <resultId>,
    "groupByQueryName": "<groupByQueryName>",
    "queryUid": <groupByQueryUid>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

67 ExecuteJob

Detailed Description

ExecuteJob is used to execute a job immediately. This request is used for execute job only (from the right click menu in the table), not for the save and execute job (icon in the job form). User privilege to execute jobs is needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "executeJob",
    "uid": <uid>
}
```

- "uid" (mandatory): uid of the job which should be started

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

68 ExecuteQuery

Detailed Description

ExecuteQuery is used to execute a query. The Query can be either an investigation query, a simulation query or a common point query. When executing a simulation query, the simulation has to be valid.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "executeQuery",
  "uid": <queryUid>,
  "type": <type>,
  "revision": <revisionUid>
  "data": {
    "parameter": "<queryIndexValue>",
    "urid": <urid>,
    "indexValues": ["<queryIndexValue>"]
  }
}
```

- "uid" (mandatory): The Query to execute.
- "type" (mandatory): The type of the Query to execute. One of ["query", "commonPointQuery", "simulationQuery"].
- "revision" (optional): Used by simulation queries. The Revision (identified by its uid) running a simulation in which to execute the Query.
- "data" (mandatory): Depending on "type" different values have to be set.
- "data:parameter" (optional): Used by investigation and simulation queries. It holds the index value to run the query on.
- "data:urid" (optional): Used by investigation queries.
- "data:indexValues" (optional): Used by common point queries.

Returns

A HTTP response with the following content:

```
{
  "resultId": <resultId>,
  "queryUid": <queryUid>,
  "attributes": <attributes>,
  "parameter": "<parameter>",
  "format": "<format>" | "None",
  "responseStatus": ["<status>", "<feedbackText>"],
  "reloadUserProfile": true|false
}
```

- "attributes": Only present after executing a common point query.
- "parameter": For simulation queries contains the parameter the query was called with, for investigation or common point queries contains the index value on which the query was run.

69 ExecuteReport

Detailed Description

ExecuteReport executes a report. The requesting User must have the privilege to view and generate already defined reports.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "executeReport",
    "uid": <reportUid>
}
```

- "uid" (mandatory): The Report to execute.

Returns

A HTTP response with the following content:

```
{
    "reportResult": [
        ...
    ],
    "type": <type>,
    "timeFrom": <fromTimestamp>,
    "timeTo": <toTimestamp>,
    "generatedOn": <generatedOnTimestamp>,
    "generatedBy": "<generatedByUserName>",
    "usedMandators": ["<mandatorName>", ...],
    "name": "<reportName>",
    "comment": "<comment>",
    "uid": <reportUid>,
    "reportingAttributesConditions": "<conditions>",
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


70 ExecuteRuleQuery

Detailed Description

ExecuteRuleQuery executes a SimulationQuery of type QUERY_TYPE_RULE. The requesting User must have the privilege to activate and configure modelling functions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "executeRuleQuery",
  "revision": <revisionUid>,
  "data": {
    "parameter": "<ruleName>",
    "ruleUid": <ruleUid>
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) running a simulation in which to execute the SimulationQuery.
- "data" (mandatory): Contains either "ruleUid" or "parameter" to identify the SimulationQuery by its uid or name respectively.

Returns

A HTTP response with the following content:

```
{
  "resultId": <resultId>,
  "queryUid": <queryUid>,
  "parameter": "<ruleName>",
  "format": "None",
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

71 ExecuteRuleReportQuery

Detailed Description

ExecuteRuleReportQuery executes a SimulationQuery of Type QUERY_TYPE_RULE_REPORT. The requesting User must have the privilege to activate and configure modelling functions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "executeRuleReportQuery",
  "revision": <revisionUid>,
  "data": {
    "parameter": "<ruleName>",
    "ruleUid": <ruleUid>
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) running a simulation in which to execute the SimulationQuery.
- "data" (mandatory): Either "ruleUid" or "parameter" have to be set.
- "data:parameter" (optional): The name of the rule the SimulationQuery should run on.
- "data:ruleUid" (optional): The uid of the rule the SimulationQuery should run on.

Returns

A HTTP response with the following content:

```
{
  "resultId": <resultId>,
  "queryUid": <queryUid>,
  "parameter": "<ruleName>",
  "format": "<format>"|"None",
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

72 FixMillisecondsMapping

Detailed Description

FixMillisecondsMapping fixes milliseconds mapping for an attribute. This is a maintenance function. During next golive, takes an integer Value in format YYYYMMDDhhmmssZZZ converts it to a timestamp value in milliseconds since 1970-01-01 00:00:00.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "executeJob",
    "mandator": <mandatorUid>,
    "revision": <revisionUid>,
    "uid": <attributeUid>
}
```

- "mandator" (optional): uid of the mandator of the revision which owns the attribute
- "revision" (mandatory): uid of the revision which owns the attribute
- "uid" (mandatory): uid of the attribute which should be changed

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

73 GenerateRulesUndoRule

Detailed Description

GenerateRulesUndoRule discard the generated conditions of a not-yet-completed rule and recompute statistics. With this function, it is possible to reset committed conditions before rule was generated.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "generateRulesUndoRule",
    "revision": " <revisionUid>"
}
```

- "revision" (mandatory): The Revision (identified by its uid) running a rulegeneration in which the conditions should be discarded.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "feedbacktext"],
    "reloadProfile": true|false
}
```

74 Get

Detailed Description

Get retrieves an element from the server. Depending on the type of element the requesting User must have the privilege to view different areas of Safer Payments.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "get",
    "uid": <elementUid>,
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The element (identified by its uid) to retrieve.
- "mandator" (optional): Identifies the mandator to look in. Usage depends on type of element.
- "revision" (optional): Identifies the revision to look in. Usage depends on type of element.

Returns

A HTTP response with the following content:

```
{
    "data": {
        ...
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

75 GetActivatedComplianceLists

Detailed Description

GetActivatedComplianceLists retrieves a report about which compliance lists are currently activated. The requesting User must have the privilege to view compliance lists (global or role privilege).

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getActivatedComplianceLists",
    "mandator": <mandatorUid>
}
```

- "mandator" (optional): The Mandator (identified by its uid) used to check the role privilege if the requesting User does not have the global one.

Returns

A HTTP response with the following content:

```
{
    "activatedLists": {
        "PoliticalExposedPersonList": true|false,
        "OfacSdn": true|false,
        "UnitedNationsList": true|false,
        "GlobalWatchList": true|false,
        "EuropeanTerrorList": true|false,
        "RussianTerrorList": true|false
    },
    "currentlyReloading": true|false,
    "readonly": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

76 GetAdHocCheckComplianceListEntriesTable

Detailed Description

GetAdHocCheckComplianceListEntriesTable performs ad-hoc check and returns a table containing the result.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getAdHocCheckComplianceListEntriesTable",
  "data": {
    "selectedLists": [<complianceListUid>],
    "name": "<name>",
    "street": "<street>",
    "city": "<city>",
    "country": "<country>",
    "passport": "<passport>",
    "birthdate": "<birthDate>"
  }
}
```

- "data" (mandatory):
- "data:selectedLists": The selected compliance lists.
- "data:name": The name of the user.
- "data:street": The street of the user.
- "data:city": The city of the user.
- "data:country": The country of the user.
- "data:passport": The passport number of the user.
- "data:birthdate": The birthdate of the user.

Returns

A HTTP response with the following content:

```
{
  "tableContent": [
    {
      "id": <listId>,
      "type": "<typeOfHit>",
      "entityType": "<entityType>",
      "entryUniqueId": <entryUniqueId>,

```

```
        "name": "<name>",
        "streetAddress": "<street>",
        "city": "<city>",
        "country": "<country>",
        "blacklistType": "<blackListType>",
        "passportNumber": "<passportNumber>",
        "birthdate": "<birthDate>",
        "title": "<title>",
        "otherInformation": "<otherInformation>",
        "negativeNewsAlert": true|false,
        "computedScore": <computedScore>,
        "blacklistHitType": "<blackListHitType>"
    },
    ... ..
],
"resultId": <resultId>,
"numberOfEntriesShown": <numberOfEntriesShown>,
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```


77 GetAggregatedAuditTrails

Detailed Description

GetAggregatedAuditTrails retrieves the aggregated audit trail for an attribute in case investigation.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getAggregatedAuditTrails",
    "uid": <caseUid>,
    "data": {
        "attributeUid": <reportingAttributeUid>
    }
}
```

- "uid" (mandatory): The Case (identified by its uid) for which the audit trail is requested.
- "data:attributeUid" (mandatory): The ReportingAttribute (identified by its uid) for which the aggregated audit trail is requested.

Returns

A HTTP response with the following content:

```
{
    "auditTrails": [
        {
            "UTC": <timestamp>,
            "caseQueueId": <caseQueueId>,
            "caseQueueName": "<caseQueueName>",
            "caseUid": <caseUid>,
            "hits": <numberOfHits>,
            "score": <score>,
            "status": "<status>",
            "caseActionName": "<caseActionName>",
            "attachmentName": "<attachmentName>",
            "user": "<userName>",
            "userId": <userId>,
            "followUpUser": "<followUpUser>",
            "followUpTimestamp": <followUpTimestamp>,
            "comment": "<comment>",
            "cppName": "<cppName>",
            "caseCloseCode": "<caseCloseCode>",
            "fraudStatus": "<fraudStatus>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

78 GetAllProfilingsTable

Detailed Description

GetAllProfilingsTable retrieves the table of all profilings from a given Revision. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getAllProfilingsTable",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) to retrieve the profilings table from.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

79 GetAllRulesTable

Detailed Description

GetAllRulesTable retrieves all rules and final rules for a certain revision including inherited rules.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getAllRulesTable",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the all rules table is requested.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <rulesetUid>,
            "<rulesetName>",
            <rulesetPriority>,
            <rulesetEnabled>"Yes" | "No",
            "<rulesetConditions>",
            <ruleUid>,
            <revisionUid>,
            <rulePriority>,
            "<ruleName>",
            "<ruleComment>",
            <ruleEnabled>"Yes" | "No",
            "<ruleConditions>",
            "<ruleConclusion>",
            "<actions>",
            "<mandatorName>",
            <mandatorUid>,
            <inherited>"Yes" | "No",
            <finalRuleset>"Yes" | "No"
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

80 GetAnalysedAttributes

Detailed Description

GetAnalysedAttributes retrieves all own and inherited attributes that are enabled for Analysis in Modelling.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getAnalysedAttributes",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) of the attribute analysis.

Returns

A HTTP response with the following content:

```
{
    "attributes": [
        {
            "uid": <attributeUid>,
            "name": "<attributeName>",
            "mandator": <mandatorUid>,
            "revision": <revisionUid>,
            "mdcStored": true|false,
            "origin": "<attributOrigin>",
            "type": "<attributeType>",
            "encrypted": true|false,
            "length": <attributeLength>,
            "decimals": <decimals>,
            "dimension": "<dimension>",
            "categoriesEnabled": true|false,
            "categories": [...],
            "usedInMergingConclusion": true|false,
            "metaAttribute": "<metaAttributeName>",
            "format": "<format>" | "None"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

81 GetAnalysisDataSelection

Detailed Description

GetAnalysisDataSelection retrieves the data selection in the analysis definition, and it also retrieves general settings of analysis.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getAnalysisDataSelection",
  "uid": <analysisId>,
  "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the analysis is defined.
- "data:analysisId" (mandatory): The Analysis (identified by its id) for which the data selection is requested.

Returns

A HTTP response with the following content:

```
{
  "analysis": {
    "enabled": true|false,
    "name": "<analysisName>",
    "comment": "<comment>",
    "optimization": true|false,
    "adoptSimulationDataSelection": true|false,
    "dataSelection": {
      "mandators": [<mandatorUid>],
      "periodType": "<periodType>",
      "actual": {
        "fromUrid": <fromUrid>,
        "fromRecord": <fromRecord>,
        "fromTimestamp": <fromTimestamp>,
        "fromDays": <fromDays>,
        "toUrid": <toUrid>,
        "toRecord": <toRecord>,
        "toTimestamp": <toTimestamp>,
        "toDays": <toDays>
      },
      "desired": {
        "fromUrid": <fromUrid>,
        "fromRecord": <fromRecord>,
        "fromTimestamp": <fromTimestamp>,

```

```
        "fromDays": <fromDays>,
        "toUrid": <toUrid>,
        "toRecord": <toRecord>,
        "toTimestamp": <toTimestamp>,
        "toDays": <toDays>
    },
    "incDdc": true|false,
    "conditions": [<conditions>]
},
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

82 GetAnalysisProgress

Detailed Description

GetAnalysisProgress retrieves the analysis progress on an analysis. It is used by various analysis dependent functionalities in the model.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getAnalysisProgress",
  "revision": <revisionUid>,
  "data": {
    "analysisId": <analysisId>
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the analysis is defined.
- "data:analysisId" (mandatory): The Analysis (identified by its id) for which the progress is requested.

Returns

A HTTP response with the following content:

```
{
  "data": {
    "simulationValid": true|false,
    "simulationEnabled": true|false,
    "isDataInIris": true|false,
    "simulationStatus": "<simulationStatus>",
    "simulationProgress": <simulationProgress>,
    "progress": <progress>,
    "recordsProcessed": <recordsProcessed>,
    "optimizationProgress": <optimizationProgress>,
    "manRecordsProcessed": <manRecordsProcessed>,
    "finished": true|false,
    "optimizing": true|false,
    "uridFrom": <uridFrom>,
    "uridTo": <uridTo>,
    "refresh": <autoRefreshSetting>,
    "analysisResultsValid": true|false,
    "analysisStatus": "<analysisStatus>"
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

83 GetAnalysisTable

Detailed Description

GetAnalysisTable retrieves own and inherited analyses.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getAnalysisTable",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) for which the analysis table is requested.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <analysisId>,
            <enabled>true|false,
            "<name>",
            "<comment>",
            <ruleOptimization>true|false,
            "<currentStatus>"
        ],
        ... ..
    ],
    "refresh": <autoRefreshSetting>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


84 GetAttributes

Detailed Description

GetAttributes gets attribute uids in the context of a reservable. These can be input, or output attributes, or for special elements reporting attributes. Takes mandator into account and checks privileges for user accordingly. For elements like case selection takes filter settings into account.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getAttributes",
    "uid": <uid>,
    "type": <type>,
    "mandator": <mandatorUid>,
    "revision": <revisionUid>,
    "data": { "index": <indexUid> }
}
```

- "uid" (optional): uid of an monitoring list, index based evaluation or attribute, depending on type
- "type" (optional): type of the requests which filters the attribute for some specific use case. The type defines, which other elements of the requests are mandatory and which internal logic is used to filter the attributes. Possible types are:

```
"compromisedConditionsCollusions"
"complianceList"
"fraudFlaggingTimestamp"
"doubletDetection"
"mandator"
"query"
"groupByQuery"
"definedRiskListImport"
"definedRiskList"
"definedRiskListRuleAction"
"definedRiskListImport"
"aPrioriConditions"
"counterEvaluation"
"counterEvaluationIncludingCurrent"
"counterReference"
"counterReferenceIncludingCurrent"
"mergingConclusion"
"mergingTarget"
"mergingTermination"
"deviceIdentification"
"deviceIdentificationInputs"
"eventIncludingCurrent"
"indexComputation"
"indexInsertion"
"complianceList"
```

```

"mergingSource"
"pattern"
"profilePre"
"ruleConclusion"
"event"
"ruleset"
"ruleConclusionExpression"
"ruleCondition"
"profilePost"
"modelTrainingDataSelection"
"modelVerificationDataSelection"
"connivanceConditionsCollusions"
"counterpartyConditionsCollusions"
"stencil"
"precedent"
"precedentsReference"
"queryDataSelection"
"groupByQueryDataSelection"
"analysisDataSelection"
"simulationQuery"
"simulationQueryDataSelection"
"simulation"
"stencil"
"precedent"
"precedentsReference"
"caseQueue"
"createRulesScoring"
"definedRiskInput"
"definedRiskOutput"
"reporting"
"formula"
"transaction"
"mergingConclusionExpression"

```

- "mandator" (optional): mandator, which is used for the request
- "revision" (optional): revision of the request
- "data: index" (optional) : the index uid which is used for specific request types

Returns

A HTTP response with the following content:

```

{
  "attributes": [{
    "uid":<uid>,
    "name":<name>,
    "type": "Numeric" | "Text" | "Hexadecimal" | "Boolean" | "IP" | ...,
    "metaAttribute":<metaAttributeType>,
    "mdcStored":true|false},
    ....
  ], ...],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}

```

85 GetAuditLogTable

Detailed Description

GetAuditLogTable retrieves the audit log table. With the request parameters the audit log table can be filtered by date, log level, instance and user. When exporting the table the parameter lastEntry is used to export only the currently shown table.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getAuditLogTable",
  "data": {
    "from": <fromTimestamp>,
    "to": <toTimestamp>,
    "filter": ["A", "C", "D", "X", "E", "F", "I", "N", "W"],
    "instances": [<instanceUids>]
    "users": [<userUids>]:
    "lastEntry": -1|<maxTimestamp>
  }
}
```

- "data:from" (mandatory): The start time to filter the audit log entries.
- "data:to" (mandatory): The end time to filter the audit log entries.
- "data:filter" (mandatory): A concatenated string containing the log levels to filter the audit log entries.
- "data:instances" (mandatory): The IrisInstances (identified by their uids) to filter the audit log entries.
- "data:users" (mandatory): The Users (identified by their uids) to filter the audit log entries.
- "data:lastEntry" (mandatory): The maximum timestamp of an audit log entry to be included or -1 for all entries.

Returns

A HTTP response with the following content:

```
{
  "recordLimit": <recordLimitSetting>,
  "tableContent": [...],
  "from": <fromTimestamp>,
  "to": <toTimestamp>,
  "synced": true|false,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

86 GetCaseAction

Detailed Description

GetCaseAction shows a case action in a case.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCaseAction",
  "uid": <caseActionUid>,
  "mandator": <mandatorUid>,
  "data": {
    "caseUid": <caseUid>
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "caseAction": {
    "uid": <caseActionUid>,
    "enabled": true|false,
    "mandator": {
      "uid": <mandatorUid>,
      "name": "<mandatorName>",
      "maxMemoryForSimulationGB": <maxMemoryForSimulationInGB>
    },
    "name": "<caseActionName>",
    "comment": "<comment>",
    "targetType": "<targetType>",
    "messageAddress": "<messageAddress>",
    "messagePort": <messagePort>,
    "userName": "<userName>",
    "contentType": "<contentType>",
    "targetPath": "<targetPath>",
    "useSsl": true|false,
    "certificatePath": "<certificatePath>",
    "useClientCertificate": true|false,
    "clientCertificatePath": "<clientCertificatePath>",
    "clientCertificateKeyPath": "<clientCertificateKeyPath>",
    "clientCertificateKeyType": "<clientCertificateKeyType>",
    "clientCertificateKeyPasswordPath": "<clientCertificateKeyPasswordPath>",
    "basicAuthUsername": "<userName>",
    "basicAuth": true|false,
    "fromAddress": "<fromAddress>",
  }
}
```

```
"addressType": "<addressType>",
"constantToAddress": "<constantToAddress>",
"formatAttributeValues": true|false,
"maskAttributeValues": true|false,
"subjectTemplate": "<template>",
"bodyTemplate": "<template>",
"messageTemplate": "<template>",
"attachment": {
    "name": "<name>",
    "comment": "<comment>",
    "size": <size>,
    "lastChangedBy": "<lastChangedByUserName>",
    "lastChangedOn": <lastChangedOnTimestamp>,
    "data": "<data>"
},
"lastChangedBy": "<lastChangedByUserName>",
"lastChangedTimestamp": <lastChangedTimestamp>,
"emailTo": "<emailTo>"
},
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

87 GetCaseActionPreview

Detailed Description

GetCaseActionPreview previews a case action. This request is a POST request. It supports the case action types in SMTP and ODBC SQL.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCaseActionPreview",
  "uid": <caseUid>,
  "mandator": <caseMandatorUid>,
  "data": {
    "caseActionUid": <caseActionUid>
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "preview": {
    ...
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

88 GetCaseActions

Detailed Description

GetCaseActions lists all case actions for a mandator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseActions",
    "mandator": <mandatorUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "caseActions": [
        {...},
        {...},
        ... ..
    ],
    "responseStatus": ["<status>", "<feedback>"],
    "reloadUserProfile": true|false
}
```

89 GetCaseActionsFromCaseQueue

Detailed Description

GetCaseActionsFromCaseQueue lists all case actions for a case.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseActionsFromCaseQueue",
    "uid": <caseUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "caseActions": [
        {
            "uid": <caseActionUid>,
            "enabled": true|false,
            "name": "<caseActionName>",
            "targetType": "<targetType>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


90 GetCaseActionsTable

Detailed Description

GetCaseActionsTable returns a table of all case actions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseActionsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [... ...],
        [... ...],
        ... ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

91 GetCaseCloseCodes

Detailed Description

GetCaseCloseCodes lists all available case close codes for specified case and mandators.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCaseCloseCodes",
  "type": "bulkClose" | "",
  "data": {
    "transitionUid": <transitionUid>,
    "mandators": [<mandatorUid>, ...]
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "caseCloseCodes": [
    {
      "uid": <caseCloseCodeUid>,
      "name": "<name>",
      "comment": "<comment>",
      "caseActions": [
        ... ..
      ],
      "fraudStatus": <fraudStatus>,
      "lastChangedBy": "<lastChangedByUserName>",
      "lastChangedTimestamp": <lastChangedTimeStamp>
    },
    ... ..
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

92 GetCaseCloseCodesTable

Detailed Description

GetCaseCloseCodesTable returns a table of all case close codes the user can access.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseCloseCodesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            ... ..
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

93 GetCaseData

Detailed Description

GetCaseData returns all data of a case.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseData",
    "uid": <caseUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "caseData": [
        {
            "uid": <caseUid>,
            "Queue": "<caseQueueName>",
            "Generated": <generatedAtTimestamp>,
            "Score": <score>,
            "Hits": <numberOfHits>,
            "Status": "<status>",
            "FraudStatus": "<fraudStatus>",
            "CloseCodeName": "<caseCloseCodeName>",
            "User": <userId>,
            "UserName": "<userName>",
            "LastUserAction": <lastUserActionTimestamp>,
            "MandatorName": "<mandatorName>",
            "Mandator": <mandatorUid>,
            "CppName": "<cppName>",
            "Cpp": "<cpp>",
            "<attributeUid>": "<attributeValue>",
            ... ..
        }
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

94 GetCaseGroupDefinitions

Detailed Description

GetCaseGroupDefinitions lists all case groups.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseGroupDefinitions"
}
```

Returns

A HTTP response with the following content:

```
{
    "casaeGroups": [
        {
            "streamtype": "caseGroup",
            "uid": <caseGroupUid>,
            "name": "<caseGroupName>",
            "comment": "<comment>",
            "mandatorUid": <mandatorUid>,
            "inherited": true|false,
            "enabled": true|false,
            "evaluationAttributeUid": <evaluationAttributeUid>,
            "lastUsedId": <lastUsedId>,
            "cpps": [ ... ]
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

95 GetCaseGroupsTable

Detailed Description

GetCaseGroupsTable returns a table of all case groups the user can access.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseGroupsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <caseGroupUid>,
            <mandatorUid>,
            "<mandatorName>",
            "<caseGroupName>",
            "<comment>",
            <enabled>"Yes" | "No",
            <inheritToSubMandators>"Yes" | "No",
            "<evaluationAttributeName>"
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

96 GetCaseHistory

Detailed Description

GetCaseHistory returns a table of case history.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCaseHistory",
  "uid": <caseUid>,
  "data": {
    "attributeUid": <reportingAttributeUid>
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "header": [
    {
      "uid": "<uid>",
      "name": "<name>",
      "align": "<align>",
      "type": "<type>",
      "format": "<format>"
    },
    ... ..
  ],
  "tableContent": [
    {
      "uid": <caseUid>,
      "Queue": "<caseQueueName>",
      "Generated": <generatedOnTimestamp>,
      "Score": <score>,
      "Hits": <numberOfHits>,
      "Status": "<status>",
      "FraudStatus": "<fraudStatus>",
      "CloseCodeName": "<closeCodeName>",
      "User": <userId>,
      "UserName": "<userName>",
      "LastUserAction": <lastUserActionTimestamp>,
      "MandatorName": "<mandatorName>",
      "Mandator": <mandatorUid>,
      "CppName": "<cppName>",
      "Cpp": "<cpp>",

```

```
        "<attributeUid>": "<attributeValue>",  
        ... ..  
    },  
    ... ..  
],  
"responseStatus": ["<status>", "<feedbacktext>"],  
"reloadUserProfile": true|false  
}
```


97 GetCaseQueueReportsTable

Detailed Description

GetCaseQueueReportsTable returns a table of all case queue reports.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseQueueReportsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <caseQueueReportUid>,
            "<mandatorName>",
            "<reportName>",
            "<comment>",
            "<lastChangedByUserName>",
            "<lastChangedTimestamp>"
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

98 GetCaseQueues

Detailed Description

GetCaseQueues lists all available case queues for the specified type.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseQueues",
    "type": "<type>",
    "mandator": <mandatorUid>
}
```

- "type": Available values this parameter are: "caseFilter", "kpiFilter", "conclusion", "query", "collusion" (requires "revision") , "caseCreation", "report","userReport".

Returns

A HTTP response with the following content:

```
{
    "caseQueues": [
        {
            "uid": <caseQueueUid>,
            "id": <id>,
            "name": "<caseQueueName>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

99 GetCaseQueuesTable

Detailed Description

GetCaseQueuesTable returns a table of all case queues the user can view.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseQueuesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            ... ..
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

100 GetCasesTable

Detailed Description

GetCasesTable returns a table of all filtered cases the user can view. This request is similar to the request "getQuickSearchCasesTable".

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCasesTable",
  "data": {
    "mandators": [
      <mandatorUid>,
      ... ..
    ],
    "caseQueues": [
      <caseQueueUid>,
      ... ..
    ],
    "dateRestriction": {
      "from": <fromTimestamp>,
      "to": <toTimestamp>
    },
    "status": {
      "New": true|false,
      "Investigated": true|false,
      "Followup": true|false,
      "Due": true|false,
      "Closed": true|false,
      "Reopened": true|false
    },
    "users": [<userId>],
    "scoreFrom": <caseScoreFromValue>,
    "scoreTo": <caseScoreToValue>,
    "cpps": ["<cppUid_caseGroupUid>", ...]
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "header": [
    {
      "uid": "<uid>",
      "name": "<name>",
      "align": "<align>",
```

```

        "type": "<type>",
        "format": "<format>"
    },
    ... ..
],
"tableContent": [
    {
        "uid":<caseId>,
        "Queue": "<caseQueueName>",
        "Generated": <generatedTimestamp>,
        "Score": <scoreValue>,
        "Hits": <numberOfHits>,
        "Status": "<status>",
        "FraudStatus": "<fraudStatus>",
        "CloseCodeName": "<caseCloseCodeName>",
        "User": <userId>,
        "UserName": "<userName>",
        "LastUserAction": <lastUserActionTimestamp>,
        "MandatorName": "<mandatorName>",
        "Mandator": <mandatorId>,
        "CppName": "<cppName>",
        "Cpp": "<cppCode>",
        ... ..
    },
    ... ..
],
"responseStatus": ["<status>", "feedbacktext"],
"reloadProfile": true|false
}

```

101 GetCaseStates

Detailed Description

GetCaseStates lists available case states for a type.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCaseStates",
  "uid": <targetCaseQueueUid>,
  "type": "caseQueue"|"kpiFilter"|"sai"|"caseFilter"|"caseWorkflow",
  "mandator": <mandatorUid>,
  "data": {
    "mandators": [<mandatorUid>, ...]
  }
}
```

- "uid": This is required only if the "type" is "caseQueue".
- "mandator": This is required only if the "type" is "caseWorkflow".
- "data:mandators": This is required only if the "type" is "kpiFilter" or "sai".

Returns

A HTTP response with the following content:

```
{
  "caseStates": [
    {
      "uid": <caseStateUid>,
      "name": "<caseStateName>",
      "exclusiveState": true|false,
      "meta": "<meta>"
    },
    ... ..
  ],
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

102 GetCaseStatesTable

Detailed Description

GetCaseStatesTable returns a table of defined case states.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseStatesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <caseStateUid>,
            <mandatorUid>,
            "<mandatorName>",
            "<caseStateName>",
            "<comment>",
            "<lastChangedByUserName>",
            <lastChangedOnTimestamp>,
            <defaultState>"Yes"|"No"
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

103 GetCaseTransitions

Detailed Description

GetCaseTransitions lists available case transitions for the specified case(s).

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

There are two ways to run this request.

Option 1:

```
{
    "request": "getCaseTransitions",
    "uid": <caseUid>
}
```

Option 2:

```
{
    "request": "getCaseTransitions",
    "type": "caseSelection",
    "data": {
        "cases": [<caseUid>, ...]
    }
}
```

Returns

A HTTP response with the following content:

```
{
    "caseTransitions": [
        {
            "uid": <caseTransitionUid>,
            "name": "<caseTransitionName>",
            "justificationCodes": [<justificationCode>, ...],
            "justificationMandatory": true|false,
            "commentAllowed": true|false,
            "commentMandatory": true|false,
            "targetStateExclusive": true|false,
            "targetStateMeta": "<targetStateMeta>"
        },
        ... ..
    ],
    "responseStatus": [<status>, "feedbacktext"],
    "reloadProfile": true|false
}
```


104 GetCaseVariablesAsConditions

Detailed Description

GetCaseVariablesAsConditions returns all case variables (such as case age, case mandator, etc.) All case variables are wrapped in condition JSON format so that case variables can be treated / used as conditions in the User Interface.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseVariablesAsConditions"
}
```

Returns

A HTTP response with the following content:

```
{
    "caseVariables": [
        {
            "caseVariable": true|false,
            "uid": "<uid>",
            "name": "[<name>]",
            "type": "<type>",
            "metaAttribute": "None"|"<metaAttribute>",
            "categories": [ ... ]
        },
        ... ..
    ],
    "responseStatus": ["<status>", "feedbacktext"],
    "reloadProfile": true|false
}
```

105 GetCaseWorkflows

Detailed Description

GetCaseWorkflows lists all case workflows available for the case queues in the specified mandator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseWorkflows",
    "type": "caseQueue",
    "mandator": <mandatorUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "caseWorkflows": [
        {
            "uid": <caseWorkflowUid>,
            "name": "<caseWorkflowName>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "feedbacktext"],
    "reloadProfile": true|false
}
```

106 GetCaseWorkflowsTable

Detailed Description

GetCaseWorkflowsTable returns a table of all defined case workflows.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCaseWorkflowsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <caseWorkflowUid>,
            <mandatorUid>,
            "<mandatorName>",
            <enabled>"Yes"|"No",
            "<caseWorkflowName>",
            "<comment>",
            "<lastChangedByUserName>",
            <lastChangedOnTimestamp>,
            <defaultWorkflow>"Yes"|"No",
            <numberOfStates>,
            <numberOfTransitions>
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "feedbacktext"],
    "reloadProfile": true|false
}
```

107 GetChallengerStatus

Detailed Description

GetChallengerStatus shows the status of a challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getChallengerStatus",
    "revision": <revisionUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "challengerStatus": "Simulate" | "Analyze" | "Ready" | "GenerateRules",
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


109 GetCharts

Detailed Description

GetCharts lists all charts the user can view.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCharts"
}
```

Returns

A HTTP response with the following content:

```
{
    "charts": [
        {
            "streamType": "chart",
            "uid": <chartUid>,
            "position": <position>,
            "name": "<chartName>",
            "comment": "<comment>",
            "mandator": {
                "uid": <mandatorUid>,
                "name": "<mandatorName>",
                "maxMemoryForSimulationGB": <maximumMemoryForSimulationInGB>
            },
            "explanation": "<explanation>",
            "tooltip": "<tooltip>",
            "onlineHelpType": "NoOnlineHelp" | "DefaultOnlineHelp" |
"CustomAndDefaultOnlineHelp" | "CustomOnlineHelp",
            "onlineHelp": "<onlineHelp>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

110 GetChartsTable

Detailed Description

GetChartsTable returns a table of all charts the user can view.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getChartsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [... ...],
        ... ...
    ],
    "nextPosition": <nextPosition>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

111 GetCollusions

Detailed Description

GetCollusions lists all collusions in a revision.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCollusions",
    "revision": <revisionUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "collusions": [
        {
            "uid": <collusionUid>,
            "name": "<collusionName>",
            "comment": "<comment>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


112 GetCollusionSimulationResults

Detailed Description

GetCollusionSimulationResults returns the simulation result of a collusion.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCollusionSimulationResults",
    "uid": <collusionUid>,
    "revision": <revisionUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "simulationResults": [<simulationResults>],
    "progress": <progress>,
    "simulationFinished": true|false,
    "simulationRunning": true|false,
    "refresh": <autoRefreshSetting>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

113 GetCollusionsTable

Detailed Description

GetCollusionsTable returns a table of all collusions in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCollusionsTable",
    "type": "own" | "inherited",
    "revision": <revisionUid>
}
```

- "type" (mandatory): to decide whether to get collusions of own revision or inherited mandators
- "revision" : uid of the revision to get collusions for

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <revisionUid>,
            <collusionUid>,
            "<collusionName>",
            "<comment>",
            "<mandatorName>",
            "<firstPartyName>",
            "<counterPartyName>",
            <counterPartyTimeFrom>,
            <counterPartyTimeTo>,
            "<counterPartyTimeUnit>",
            <connivanceTimeBefore>,
            <connivanceTimeAfter>,
            "<connivanceTimeUnit>",
            "<caseQueueName>",
            <thresholdFirstParties>,
            "<counterPartyConditions>",
            "<connivanceConditions>",
            "<compromisedConditions>",
            []
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

114 GetCommonPointQueriesTable

Detailed Description

GetCommonPointQueriesTable returns a table of all common point queries.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCommonPointQueriesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <commonPointQueryUid>,
            <mandatorUid>,
            "<queryType>",
            <indexUid>,
            "<indexName>",
            "<mandatorName>",
            "<commonPointQueryName>",
            "<comment>",
            <enabled>"Yes" | "No",
            <columnsForCommonPoint>,
            "<lastChangedByUserName>",
            <lastChangedTimestamp>
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

115 GetCommonQuerySubsetResults

Detailed Description

GetCommonQuerySubsetResults returns all common points for a common point attribute value in the common point query result.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCommonQuerySubsetResults",
  "uid": <commonQueryUid>,
  "data": {
    "resultId": <resultId>,
    "urid" : [<urid>]
  }
}
```

- "uid" (mandatory): uid of the query that was run

Returns

A HTTP response with the following content:

```
{
  "queryResult": {
    "name": "<commonPointQueryName>",
    "comment": "<commonPointQueryComment>",
    "header": [{
      "name": "<attributeName>",
      "attributeUid": <attributeUid>,
      "align": "<align>",
      "decimals": <decimals>,
      "hyperlinkQueryUid": [{
        "name": "<hyperlinkQueryName>",
        "comment": "<comment>",
        "uid": <hyperlinkQueryUid>
      }
    ]
  },
  "commonPointQueryUid": [{
    "name": "<commonPointQueryName>",
    "comment": "<commonPointQueryComment>",
    "uid": <commonPointQueryUid>
  }
],
  "type": "<attributeType>",
  "format": "<attributeFormat>"
}
```

```
    },
    ... ..
  ],
  "tableContent": [
    ... ..
  ],
  },
  "indexValueUrid": <indexValueUrid>|-1,
  "running": true|false,
  "recordsTotal": <recordsTotal>,
  "recordsFinished": <recordsFinished>,
  "duration": <duration>,
  "amountDimension": "<amountDimension>",
  "statistics": {
    "transaction": {
      "genuine": {
        "records": <numberOfRecords>,
        "amount": <amount>
      },
      "fraud": {
        "records": <numberOfRecords>,
        "amount": <amount>
      }
    }
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

116 GetComplianceDataSourceExists

Detailed Description

GetComplianceDataSourceExists performs a check if compliance list data file exists or not. This request is a POST request.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getComplianceDataSourceExists"
}
```

Returns

A HTTP response with the following content:

```
{
    ["euslPath" | "ruslPath" | "ofacPath" | "unPath" | "pepPath"]: true|false,
    "allDataSourcesAvailable": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

117 GetComplianceDefinitionTable

Detailed Description

GetComplianceDefinitionTable returns a table of all compliance lists the user can view.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getComplianceDefinitionTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <complianceListUid>,
            <mandatorUid>,
            "<mandatorName>",
            "<complianceListName>",
            "<comment>",
            <priority>,
            "<outputAttribute>",
            "<typeOfList>",
            <searchTreeSize>,
            "<lastChangedByUserName>",
            <lastChangedOnTimestamp>,
            <enabled>["Yes" | "No"]
        ],
        ... ..
    ],
    "currentlyReloading": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

118 GetComplianceListDefinitions

Detailed Description

GetComplianceListDefinitions returns compliance list definitions for which the user has view privilege.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getComplianceListDefinitions"
}
```

Returns

A HTTP response with the following content:

```
{
    "complianceLists": [
        { ... },
        ...
    ],
    "currentlyReloading": true|false,
    "readonly": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


119 GetConclusionExpressionValuePairs

Detailed Description

GetConclusionExpressionValuePairs runs maintenance method of creating conclusion expression pair list.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getConclusionExpressionValuePairs",
    "uid": <uid>,
    "data": {
        "expressionAttribute": <attributeUid>
    }
}
```

Returns

A HTTP response with the following content:

```
{
    "values": [<values>],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

120 GetCountersTable

Detailed Description

GetCountersTable returns a table of all counters in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCountersTable",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): uid of the revision the counters are to be fetched for

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

121 GetCpp

Detailed Description

GetCpp gets a Common Purchase Point (CPP) definition

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCountersTable",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): uid of the revision the cpp is fetched from

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

122 GetCpps

Detailed Description

GetCpps gets Common Purchase Point (CPP) definitions for a mandator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getCpps",
    "uid": <caseUid>,
    "type": "caseCreation" | "caseInvestigation" | "caseFilter",
    "mandator": <mandatorUid>
}
```

- "uid" (optional): case uid, when request started from case investigation
- "type" (mandatory): to decide from which section the request was sent
- "mandator" (optional): uid of mandator, when request was sent from case creation or case investigation

Returns

A HTTP response with the following content:

```
{
    "selectedCpp": <selectedCpp>,
    "selectedCaseGroup": <selectedCaseGroup>,
    "cpps": [{
        "streamType": "cpp",
        "caseGroupUid": <caseGroupUid>,
        "mandatorUid": <mandatorUid>,
        "id": <cppId>,
        "name": "<cppName>",
        "status": "<status>",
        "comment": "<comment>",
        "reportingAttributes": [{
            "name": "<attributeName>",
            "type": "<type>",
            "format": "<format>",
            "attributeId": <attributeUid>,
            "encrypted": true|false,
            "value": <value>,
            "decimals": <decimals>,
            "category": [<category>]
        },
        ... ..
    ]
}
```

```
    ],
    "createdBy": "<createdByUserName>",
    "createdTimestamp": <createdTimestamp>,
    "lastChangedBy": "<lastChangedByUserName>",
    "lastChangedTimestamp": <lastChangedTimestamp>,
    "active": true|false,
    "inherit": true|false
  },
  ... ..
],
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

123 GetCppTypeTable

Detailed Description

GetCppTypeTable return a table of all Common Purchase Points that meet the filter criteria.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getCppTypeTable",
  "data": {
    "showAllEntries": true|false,
    "caseGroups": [<caseGroupUid>],
    "dateRestriction": {
      "all": true|false,
      "from": <fromTimestamp>,
      "to": <toTimestamp>
    },
    "status": {
      "New": true|false,
      "Investigated": true|false,
      "Followup": true|false,
      "Due": true|false,
      "Closed": true|false,
      "Reopened": true|false
    },
    "active": {
      "Active": true|false,
      "Inactive": true|false
    }
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "header" : [
    {
      "uid" : "<uid>",
      "name" : "<name>",
      "align" : "<align>",
      "type" : "<type>",
      "decimals" : <decimals>,
      "format" : "<format>"
    },
    ... ..
  ]
}
```

```

],
"tableContent" : [
    {
        "id" : <id>,
        "caseGroupUid" : <caseGroupUid>,
        "caseGroup" : "<caseGroupName>",
        "mandatorUid" : <mandatorUid>,
        "mandator" : "<mandatorName>",
        "name" : "<cppName>",
        "status" : "<status>",
        "comment" : "<comment>",
        "active" : "Yes" | "No",
        "inheritable" : "Yes" | "No",
        "createdBy" : "<createdByUsername>",
        "createdAt" : <createdAtTimestamp>,
        "lastChangedBy" : "<lastChangedByUsername>",
        "lastChangedAt" : <lastChangedAtTimestamp>,
        "numberOfAssociatedCases" : <numberOfAssociatedCases>,
        "numberOfAssociatedFraudulentCases" : <numberOfAssociatedFraudulentCases>,
        "numberOfAssociatedGenuineCases" : <numberOfAssociatedGenuineCases>,
        "rateOfFalseAlarmCases" : <rateOfFalseAlarmCases>,
        "evaluationAttribute" : {
            "value" : "<value>"
        },
        "<attributeUid>" : "<attributeValue>",
        ... ..
    },
    ... ..
],
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}

```

124 GetDashboardCharts

Detailed Description

GetDashboardCharts retrieves a list of charts. This request searches through every Mandator for which the requesting User has the "View dashboard" privilege and retrieves all charts of that Mandator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDashboardCharts"
}
```

Returns

A HTTP response with the following content:

```
{
    "dashboardCharts": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


125 GetDashboardKeyPerformanceIndicatorData

Detailed Description

GetDashboardKeyPerformanceIndicatorData returns for a given time interval and instance the key performance indicator data which will be displayed on the dashboard.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getDashboardKeyPerformanceIndicatorData",
  "uid": "<kpiUid>",
  "data": {
    "from": <fromTimestamp>,
    "to": <toTimestamp>
  }
}
```

- "uid" (mandatory): The uid of KPI for which data has to be retrieved.
- "data" (mandatory):
- "data:from": Data points that have been recorded after this value will be returned.
- "data:to": Data points that have been recorded before this value will be returned.

Returns

A HTTP response with the following content:

```
{
  "keyPerformanceIndicatorData": {
    "instance": <instanceId>,
    "timestamps": [ <x1>, <x2>, <x3>, ... , <xn> ],
    "values": [ <y1>, <y2>, <y3>, ... , <yn> ]
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

126 GetDashboardKeyPerformanceIndicators

Detailed Description

GetDashboardKeyPerformanceIndicators returns key performance indicators for a chart to be displayed on the dashboard. The response of this request does not include any actual data. The actual data is obtained by the request "getDashboardKeyPerformanceIndicatorData".

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDashboardKeyPerformanceIndicators",
    "uid": <chartUid>
}
```

- "uid" (mandatory): The uid of a chart.

Returns

A HTTP response with the following content:

```
{
    "keyPerformanceIndicators": [
        {
            "uid": <kpiUid>,
            "position": <position>,
            "name": <kpiName>,
            "unit": "<unit>",
            "tooltip": "<tooltip>",
            "instance": <instanceId>,
            "autoScale": true|false,
            "minimum": <minimum>,
            "maximum": <maximum>,
            "showLine": true|false,
            "showBars": true|false,
            "showPoints": true|false
        },
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

127 GetDefaultCapacities

Detailed Description

Fill response with default settings of mdc / ddc capacity as set in system configuration

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefaultCapacities"
}
```

Returns

A HTTP response with the following content:

```
{
    "ddcCapacity": <ddcCapacity>,
    "mdcCapacity": <mdcCapacity>,
    "mdcStored": <mdcStored>true|false,
    "ddcStored": <ddcStored>true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

128 GetDefaultMaskEncryptedValues

Detailed Description

GetDefaultMaskEncryptedValues returns the default setting for the mask encrypted values for the selected type. The default setting for masking encrypted values is used when defining new case actions, notifications or outgoing channel configurations.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefaultMaskEncryptedValues",
    "type": "CaseAction" | "Notification" | "OutgoingChannelConfiguration"
}
```

- "type" (mandatory): The type of an element. One of ["CaseAction", "Notification", "OutgoingChannelConfiguration"]

Returns

A HTTP response with the following content:

```
{
    "defaultMaskEncryptedValues": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

129 GetDefaults

Detailed Description

GetDefaults returns the default settings as json

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefaults"
}
```

Returns

A HTTP response with the following content:

```
{
    "data": {
        "streamType": "defaultSettings",
        "settings": {
            "application": {
                "name": "<applicationName>"
            },
            "pageAutoRefresh": {
                "dashboard": <valueInSeconds>,
                "investigation": <valueInSeconds>,
                "analysis": <valueInSeconds>,
                "jobSchedule": <valueInSeconds>,
                "elementGeneration": <valueInSeconds>,
                "golive": <valueInSeconds>,
                "cluster": <valueInSeconds>,
                "keys": <valueInSeconds>,
                "query": <valueInSeconds>,
                "memoryManagement": <valueInSeconds>,
                "sessionCountdown": <valueInSeconds>
            },
            "interfaces": {
                "applicationProgramming": {
                    "lockTimeoutSeconds": <lockTimeoutInSeconds>,
                    "lockTimeoutCountdownThresholdSeconds":
<lockTimeoutCountdownThresholdInSeconds>,
                    "enableCsrf": true|false,
                    "useGzip": true|false,
                    "useKeepAlive": true|false,
                    "downloadRequests": true|false,
                    "defaultLanguage": "<defaultLanguage>",
                    "defaultDateTimeFormat": "ISO" | "US" | "EUR",
                    "defaultDecimal": "<defaultDecimalSeparator>",
```

```
        "defaultDigitGroup": "<defaultDigitGroupSeparator>",
        "defaultFieldSeparator": "<defaultFieldSeparator>",
        "defaultExtendedSelectDialog": true|false,
        "defaultSearchInSelections": true|false,
        "defaultShowExplanation": true|false,
        "defaultShowUids": true|false
    }
}
},
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

130 GetDefinableCaseTransitions

Detailed Description

GetDefinableCaseTransitions lists all definable case transitions in a case workflow.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefinableCaseTransitions",
    "uid": <caseWorkflowUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "definableCaseTransitions": [
        {
            "uid": <caseTransitionUid>,
            "name": "<caseTransitionName>",
            "targetState": <caseStateUid>,
            "targetStateExclusive": true|false,
            "targetStateMeta": "<caseStateMeta>",
            "targetCaseQueue": "Current"|"Definable"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

131 GetDefinedRiskDefinitions

Detailed Description

GetDefinedRiskDefinitions lists all defined risk lists.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getDefinedRiskDefinitions",
  "type": "ruleAction" | "createEntryFromQuery",
  "mandator": <mandatorUid>
}
```

- "type" (optional): For "ruleAction" type, the "mandator" is required.

Returns

A HTTP response with the following content:

```
{
  "definedRisks": [
    {
      "streamType": "definedRiskList",
      "reencryptAuditTrailValues": true|false,
      "uid": <definedRiskListUid>,
      "name": "<definedRiskListName>",
      "comment": "<comment>",
      "priority": <priority>,
      "explanation": "<explanation>",
      "starts": true|false,
      "expires": true|false,
      "expiresByDefault": true|false,
      "defaultLifeTime": <defaultLifeTime>,
      "label": "<label>",
      "active": true|false,
      "lastUsedId": <lastUsedId>,
      "mandator": <mandatorUid>,
      "enableForRuleAction": true|false,
      "ruleAction": {
        "comment": "<comment>",
        "label": "<label>",
        "active": true|false,
        "overwrite": true|false,
        "conditions": [<conditions>]
      }
    },
    ... ..
  ],
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


132 GetDefinedRiskListAuditTrailTable

Detailed Description

GetDefinedRiskListAuditTrailTable returns all audit trail for a defined risk list.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefinedRiskListAuditTrailTable",
    "uid": <definedRiskListUid>,
    "mandator": <mandatorUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "definedRiskListName": "<definedRiskListName>",
    "auditTrail": [{
        "reencryptAuditTrailValues": true|false,
        "notificationEntry": true|false,
        "userName": "<userName>",
        "userId": <userId>,
        "createdByRuleAction": true|false,
        "utc": <timestamp>,
        "action": "<action>",
        "encrypted": true|false,
        "changedElementName": "<changedElementName>",
        "riskListName": "<riskListName>",
        "riskListUid": <riskListUid>,
        "comparisonResult": "<comparisonResult>",
        "type": "<riskListType>",
        "changes": [{
            ... ...
        }]
    }],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

133 GetDefinedRiskListDefinitionTable

Detailed Description

GetDefinedRiskListDefinitionTable returns a table of filtered defined risk lists.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getDefinedRiskListDefinitionTable",
  "data": {
    "selectFilter": "all" | "active" | "inactive"
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "allowDelete": true|false,
  "tableContent": [[
    <definedRiskListUid>,
    <mandatorUid>,
    "<mandatorName>",
    "<definedRiskListName>",
    "<comment>",
    <priority>,
    "<inputAttributeName>",
    "<outputAttributeName>",
    "<outputAttributeValue>",
    <numberOfEntries>,
    "<lastChangedByUserName>",
    <lastChangedOnTimestamp>,
    <enabled>"Yes"|"No",
    <enabledForRuleActions>"Yes"|"No"
  ],
  ... ..
],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

The value for "allowDelete" is decided by the system setting "Permanent deletion of defined risk lists".

134 GetDefinedRiskListEntriesTable

Detailed Description

GetDefinedRiskListEntriesTable is used to get entries of defined risk lists. It may also use filters selections for specific users. This request is a POST request. To search for risk list entries last changed by "rule action", the "ruleAction" filter should be true.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getDefinedRiskListEntriesTable",
  "uid": <definedRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "label": "<labelFilterValue>",
    "comment": "<commentFilterValue>",
    "value": "<riskListInputAttributeValue>",
    "showEnabled": true|false,
    "showDisabled": true|false,
    "showAllEntries": true|false,
    "lastChangeRestriction": {
      "all": false,
      "from": <lastChangedOnFrom>,
      "to": <lastChangedOnTo>
    },
    "validRestriction": {
      "all": false,
      "from": <startsAtFrom>,
      "to": <startsAtTo>
    },
    "expiresRestriction": {
      "all": false,
      "from": <expiresFrom>,
      "to": <expiresTo>
    },
    "users": [<userId>, ...],
    "ruleAction": true|false
  }
}
```

Returns

A HTTP response with the following content:

```

{
  "tableContent": [[
    "<inputAttributeValue>",
    "<entryId>",
    "<outputCategoryAttributeValue>",
    "<startsAt>",
    "<expiresAt>",
    "<comment>",
    "<lastChangedBy>",
    "<lastChangedOn>",
    "<filterCriteria>",
    "<label>",
    "<enabled>\"Yes\" | \"No\"",
  ],
  ...
],
"name": "<definedRiskListName>",
"attributeName": "<inputAttributeName>",
"expiresEnabled": true|false,
"expiredByDefault": true|false,
"defaultLifeTime": <defaultLifeTime>,
"startsAtEnabled": true|false,
"explanation": "<explanation>",
"inputAttributeIsEncrypted": true|false,
"deleteEnabled": true|false,
"activateEnabled": true|false,
"showAuditTrail": true|false,
"numberOfEntriesShown": <numberOfEntriesShown>,
"outputAttributeCategoriesEnabled": true|false,
"outputAttributeIsEncrypted": true|false,
"attributeFormat": "<attributeFormat>",
"attributeType": "<attributeType>",
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false,
}

```

135 GetDefinedRiskListEntry

Detailed Description

GetDefinedRiskListEntry gets information of a single defined risk list entry. This request is a POST request.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getDefinedRiskListEntry",
  "uid": <definedRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "attributeValue": "<entryInputAttributeValue>",
    "id": <entryId>
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "entry": {
    "value": "<value>",
    "attributeName": "<attributeName>",
    "attribute": {
      "type": "<type>",
      "format": "<format>" | "None",
      "decimals": <decimals>,
      "length": <length>
    },
    "outputAttributeCategoryValue": "<value>",
    "expiresAt": <expiresAtTimestamp>,
    "startsAt": <startsAtTimestamp>,
    "id": <entryId>,
    "comment": "<comment>",
    "label": "<label>",
    "active": true|false,
    "conditions": [ <conditions> ],
    "expires": true|false,
    "starts": true|false,
    "lastChangedBy": "<lastChangedByUserName>",
    "lastChangedTimestamp": <lastChangedTimestamp>
  },
  "canEdit": true|false,
  "responseStatus": [ "<status>", "<feedbacktext>" ],
  "reloadUserProfile": true|false
}
```

136 GetDefinedRiskListEntryAttribute

Detailed Description

GetDefinedRiskListEntryAttribute shows the input attribute for a defined risk list

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefinedRiskListEntryAttribute",
    "uid": <definedRiskListUid>,
    "mandator": <mandatorUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "attributeName": "<inputAttributeName>",
    "attribute": {
        "type": "<attributeType>",
        "format": "<attributeFormat>",
        "decimals": <decimals>,
        "encrypted": true|false,
        "length": <attributeLength>
    },
    "outputAttributeCategoriesEnabled": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

137 GetDefinedRiskListPrivileges

Detailed Description

GetDefinedRiskListPrivileges shows user privileges for the defined risk lists.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefinedRiskListPrivileges",
    "mandator": <mandatorUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "viewAuditTrail": true|false,
    "allowDelete": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

The value for "viewAuditTrail" is decided by "view defined risk list audit trail" privilege setting in user's role. The value for "allowDelete" is decided by "Permanent deletion of defined risk list" setting in System Configuration.

138 GetDefinedRiskListSettings

Detailed Description

GetDefinedRiskListSettings shows "Starts" and "Expires" settings of a defined risk list.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDefinedRiskListSettings",
    "uid": <definedRiskListUid>,
    "mandator": <mandatorUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "starts": true|false,
    "expires": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


139 GetDeviceIdentificationTable

Detailed Description

GetDeviceIdentificationTable returns a table of Device Identifications in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDeviceIdentificationTable",
    "type": "own" | "inherited",
    "revision": <revisionUid>
}
```

- "type" (mandatory): type of table to show. Own is for elements defined in this mandator. Inherited is for elements defined in head mandators.
- "revision" (mandatory): Revision, identified by its uid, for which the table should be shown.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <revisionUid>,
            <deviceIdentificationUid>,
            "<deviceIdentificationName>",
            "<comment>",
            "<mandatorName>",
            "<indexName>",
            "<timestampAttribute>",
            "<cookie>",
            "<fingerPrint>",
            "<conditions>",
            [
                "<attributeNameA>",
                "<attributeNameB>"
                ... ..
            ],
            [
                "<computationForAttributeA>",
                "<computationForAttributeB>",
                ... ..
            ],
            <numberOfAttributes>
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

140 GetDonorInstances

Detailed Description

GetDonorInstances retrieves a list of IrisInstance objects usable to restore another one. All instances with Status "OK" that are not the one to be restored are returned.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getDonorInstances",
    "uid": <instanceUid>
}
```

- "uid" (mandatory): The IrisInstance (identified by its uid) to restore.

Returns

A HTTP response with the following content:

```
{
    "donorInstances": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

141 GetElementPrintView

Detailed Description

GetElementPrintView is used to get a print view in json format of a single revision element.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getElementPrintView",
  "uid": <revisionElementUid>,
  "revision": <revisionUid>
}
```

Returns

A HTTP response with the following content:

```
{
  "data": {
    "uid": <revisionUid>,
    "mandator": <mandatorUid>,
    "mandatorName": "<mandatorName>",
    "general": {
      "revision": <revisionId>,
      "name": "<revisionName>",
      "comment": "<comment>",
      "editedByUser": <editedByUserUid>,
      "editedSince": <editedSinceTimestamp>,
      "lastChangedBy": "<lastChangedByUserName>",
      "lastChangedTimestamp": <lastChangedTimestamp>,
      "goliveInitiationUser": <goliveInitiationUserUid>|-1,
      "setLifeBy": "<setLifeBy>",
      "setLifeTimestamp": <setLifeTimestamp>,
      "retiredBy": "<retiredBy>",
      "retiredTimestamp": <retiredTimestamp>,
      "status": "<revisionStatus>"
    },
    "elementType": "<elementType>",
    "<elementType>": [
      {
        ... ..
      }
    ]
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

142 GetEnabledAnalyses

Detailed Description

GetEnabledAnalyses returns a list of enabled analyses in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getEnabledAnalyses",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): Revision for which the analyses should be listed

Returns

A HTTP response with the following content:

```
{
    "enabledAnalyses": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

143 GetEventsTable

Detailed Description

GetEventsTable return a table of all Events in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getEventsTable",
    "type": "inherited" | "own",
    "revision": <revisionUid>
}
```

- "type" (mandatory): defines if the table should show the inherited elements or only the elements defined for this mandator. "inherited" is for Events defined in head mandators. "own" is for events defined in this mandator
- "revision" (mandatory): Revision, identified by its uid, for which the events should be shown

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

144 GetExecutables

Detailed Description

GetExecutables list all executables for the specified mandator(s). It returns executables such as investigation reports, group by queries and merchant monitoring rules for a report generation job.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getExecutables",
    "data": {
        "mandators": [<mandatorUid>]
    }
}
```

- "data" (mandatory):
- "data:mandators" (mandatory): Executables that belong to the selected mandators will be retrieved.

Returns

A HTTP response with the following content:

```
{
    "executables": [
        {
            "streamType": "<streamType>",
            "uid": <uid>,
            "name": "<name>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

145 GetExternalQueries

Detailed Description

GetExternalQueries is used to request available external queries. Privilege to view/change case queues or to investigate a case is needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getExternalQueries",
    "mandator": <mandatorUid>
}
```

- "mandator" (mandatory): uid of mandator to check privileges and available external queries

Returns

A HTTP response with the following content:

```
{
    "externalQueries": [...],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

146 GetExternalQueriesTable

Detailed Description

GetExternalQueriesTable is used to get the data for external queries table. Privilege to view external queries is needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getExternalQueriesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [{
        "uid": <uid>,
        "enabled": true | false,
        "comment": "<comment>",
        "name": "<name>"
    },
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


147 GetFastLinkStatusTable

Detailed Description

GetFastLinkStatusTable returns fast link status table to be shown on the cluster page.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getFastLinkStatusTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        {
            "instanceName": "<instanceName>",
            "instanceStatus": [
                {
                    ... ..
                }
            ]
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

148 GetFinalRulesets

Detailed Description

GetFinalRulesets lists all resultsets for a given revision, which includes inherited and own resultsets.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getFinalRulesets",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): Revision in which the resultsets are listed

Returns

A HTTP response with the following content:

```
{
    "rulesets": [
        {
            "uid": <rulesetUid>,
            "name": "<rulesetName>",
            "comment": "<comment>",
            "revision": <revisionUid>
        },
        ... ..
    ],
    "mayDefineFinalRulesets": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

149 GetFinalRulesetsTable

Detailed Description

GetFinalRulesetsTable returns a table of all final rulesets in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getFinalRulesetsTable",
    "type": "own" | "inherited",
    "revision": <revisionUid>
}
```

- "type": It defines if table should list inherited rulesets or only rulesets from own mandator.
- "revision" (mandatory): Revision, identified by uid, for which the table should be returned

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <rulesetUid>,
            <revisionUid>,
            <priority>,
            "<rulesetName>",
            "<comment>",
            <enabled>"Yes" | "No",
            <enabledRules>,
            <definedRules>,
            "<mandatorName>",
            "<conditions>"
        ],
        ... ..
    ],
    "mayDefineFinalRulesets": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

150 GetFittingRulesets

Detailed Description

GetFittingRulesets lists all possible target rulesets, to move a rule. All rulesets, that have the same conditions as the original ruleset of the rule or that have less to none conditions, where all defined conditions are also defined in original ruleset are returned.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getFittingRulesets",
    "uid": <ruleUid>,
    "revision": <revisionUid>
}
```

- "revision" (mandatory): Revision, identified by uid, in which the ruleset should be searched

Returns

A HTTP response with the following content:

```
{
    "rulesets": [
        {
            "uid": <rulesetUid>,
            "name": "<rulesetName>",
            "comment": "<comment>",
            "revision": <revisionUid>
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

151 GetFollowupUsers

Detailed Description

GetFollowupUsers retrieves a list of User objects that can follow up a Case. The request is either used to determine the potential users to follow up a specific Case or cases or to get a list of candidates when setting up an automatic CaseTransition to an exclusive CaseState within a CaseQueue. Depending on the context the requesting User must have the privilege to investigate cases, execute bulk transitions or administrate case classes.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getFollowupUsers",
  "type": <type>,
  "uid": <caseUid>,
  "data": {
    "mandators": [<mandatorUid>]
  }
}
```

- "type" (optional): Describes the context of the request. One of ["caseSelection", "caseQueue"]. Not used when a valid "uid" is provided.
- "uid" (optional): The Case (identified by its uid) to search potential followup users for.
- "data" (optional): Only used when "type" is "caseSelection". -"data:mandators" (optional): A list of Mandator objects (identified by their uids) from which users should be retrieved.

Returns

A HTTP response with the following content:

```
{
  "users": [
    {
      "streamType": "user",
      "uid": <userId>,
      "enabled": true|false,
      "login": <userLoginName>,
      "name": "<userName>"
    },
    ...
  ],
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

152 GetFormulasTable

Detailed Description

GetFormulasTable retrieves a table of Formula Objects from a given Revision. Either inherited or own formulas can be retrieved. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getFormulasTable",
    "type": <type>,
    "revision": <revisionUid>
}
```

- "type" (optional): When not set or set to a value different from "inherited" only formulas owned by the Revision are retrieved; otherwise only inherited formulas are retrieved.
- "revision" (mandatory): The Revision (identified by its uid) to retrieve formulas from.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

153 GetFraudValues

Detailed Description

GetFraudValues retrieves all categories of meta attribute "Fraud" from a given Mandator. The requesting User must have the privilege to manually flag fraudulent transactions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getFraudValues",
    "mandator": <mandatorUid>
}
```

- "mandator" (mandatory): The Mandator (identified by its uid) to get the fraud values from.

Returns

A HTTP response with the following content:

```
{
    "categories": [<categoriesForMetaAttributeFraud>],
    "manualFraudValue": <manualFraudValue>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

154 GetGroupByQueriesTable

Detailed Description

GetGroupByQueriesTable returns all group by queries for all mandators the user has privilege for.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getGroupByQueriesTable"
    "user": <userId>
}
```

- "user": User identified by uid, who sent the request.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <groupByQueriesId>,
            <mandatorId>,
            "<mandatorName>",
            "<groupByQueriesName>",
            "<comment>",
            "<attributeName>",
            <attributeId>,
            <lastChangedTimestamp>,
            "<lastChangedByUserName>"
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


155 GetGroupByQueryAccountResults

Detailed Description

GetGroupByQueryAccountResults lists all accounts from a group by query result. If a group by query has account analysis enabled, this function can be used, to get all accounts, which had transactions containing the group by value.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getGroupByQueryAccountResults",
  "uid": <groupByQueryUid>,
  "data": {
    "resultId": <resultId>,
    "period": <period>,
    "groupingValue": "<groupByAttributeValue>"
  }
}
```

- "uid" (mandatory): Group by query, identified by uid, for which the results should be returned.
- "data:resultId" (mandatory): resultId, which was returned by "executeGroupByQuery", to get the exact result of this execution.
- "data:period" (optional): when timing analysis is enabled, the time period can be passed here.
- "data:groupingValue" (mandatory): value which should be in transaction.

Returns

A HTTP response with the following content:

```
{
  "accountAttribute": {
    "name": "<attributeName>",
    "format": "<format>" | "None",
    "decimals": <decimals>,
    "dimension": "<dimension>",
    "type": "<type>"
  },
  "accounts": [
    "<accountName>",
    ... ..
  ],
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

156 GetGroupByQueryResult

Detailed Description

GetGroupByQueryResult returns the query result for a passed resultId. Sorting can be applied by passing a sorting type in "data:sortedBy".

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getGroupByQueryResult",
  "uid": <groupByQueryUid>,
  "data": {
    "resultId": <resultId>,
    "maxRecords": <maximumNumberOfRecords>,
    "sortedBy": "none" | "falseAlarms" | "fraudAmount" | "fraudRatio" |
    "genuineAmount" | "totalAmount"
  }
}
```

- "uid" (mandatory): uid to identify the group by query, to get the result from.
- "data:resultId" (mandatory): resultId to get information about.
- "data:maxRecords": maximum number of records to print.
- "data:sortedBy": if sorting should be applied, the kind of sorting can be chosen here.

Returns

A HTTP response with the following content:

```
{
  "dimension": "<dimension>",
  "decimals": <decimals>,
  "groupByQueryName": "<groupByQueryName>",
  "running": true|false,
  "queryResult": [
    [
      "<groupByAttributeValue>",
      "groupByAttributeValue_CATEGORY",
      <groupingUID>,
      <timePeriod>,
      "<timePeriodDisplay>",
      <totalNumber>,
      <totalAmount>,
      <totalAverage>,

```

```
        <genuineNumber>,
        <genuineAmount>,
        <genuineAverage>,
        <fraudNumber>,
        <fraudAmount>,
        <fraudAverage>,
        <falseAlarms>,
        <fraudRatio>,
        <accounts>
    ],
    ... ..
],
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

157 GetGroupByQueryResultMetaInfo

Detailed Description

GetGroupByQueryResultMetaInfo returns information about the group by query result to define how the results should be shown. This is necessary if you want to show an already executed/computed query result, where you don't know what parameters were used and how the group by query was defined for it.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getGroupByQueryResultMetaInfo",
    "uid": <groupByQueryUid>,
    "data": {
        "resultId": <resultId>
    }
}
```

- "uid" (mandatory): group by query (identified by uid), to execute function in
- "resultId" (mandatory): result id to get meta information about

Returns

A HTTP response with the following content:

```
{
    "groupByQueryName": "<groupByQueryName>",
    "groupingAttribute": {
        "name": "<groupByAttributeName>",
        "format": "<format>" | "None",
        "dataType": "<dataType>",
        "masked": 0|1
    },
    "timestampAnalysisEnabled": 0|1,
    "timestampAttribute": {
        ... ..
    },
    "accountAnalysisEnabled": 0|1,
    "rulePerformanceEnabled": 0|1,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

158 GetGroupByQueryResults

Detailed Description

GetGroupByQueryResults returns a list with all results, which are available for this group by query, in available Results. This can be used to show the user already computed results. With the resultId, getGroupByQueryResult can be used, to get all necessary information to show the result.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getGroupByQueryResults",
    "uid": <groupByQueryUid>
}
```

- "uid" (mandatory): Uid to identify the group by query, to get the results from.

Returns

A HTTP response with the following content:

```
{
    "availableResults": [
        [
            <resultId>,
            <startedOnTimestamp>,
            <finished>"Yes" | "No",
            "<dataSelectionConditions>",
            <dataRangeFromUrid>,
            <dataRangeToUrid>,
            <timingAnalysis>true|false,
            <accountAnalysis>true|false
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

159 GetIndexAspects

Detailed Description

GetIndexAspects returns the source and target attribute of peer indexes.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getIndexAspects",
  "uid": <indexUid>,
  "revision": <revisionUid>
}
```

- "uid" (mandatory): uid of index, which aspects should be returned
- "revision" (mandatory): Revision (identified by uid) where index is defined

Returns

A HTTP response with the following content:

```
{
  "indexAspects": [
    {
      "uid": <attributeUid>,
      "name": "<attributeName>"
    }
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

160 GetIndexAttribute

Detailed Description

GetIndexAttribute returns all details about the attribute which is used in the passed index. If no mandator is passed, a revision has to be passed and vice versa.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getIndexAttribute",
    "uid": <indexUid>,
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): Index (identified by uid) for which the index attribute should be shown.
- "mandator" (mandatory if no revision): Mandator (identified by uid) where the index is defined.
- "revision" (mandatory if no mandator): Champion Revision (identified by uid) where the index is defined.

Returns

A HTTP response with the following content:

```
{
    "attribute": {
        "uid": <attributeUid>,
        "name": "<attributeName>",
        "mandator": <mandatorUid>,
        "revision": <revisionUid>,
        "mdcStored": true|false,
        "origin": "<origin>",
        "type": "<type>",
        "encrypted": true|false,
        "length": <attributeLength>,
        "decimals": <decimals>,
        "dimension": "<dimension>",
        "categoriesEnabled": true|false,
        "categories": [ ... ],
        "usedInMergingConclusion": true|false,
        "metaAttribute": "<metaAttribute>" | "None",
        "format": "<format>" | "None"
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

161 GetIndexBasedEvaluations

Detailed Description

GetIndexBasedEvaluations retrieves a list of IndexBasedEvaluation objects for a given list of Mandator objects.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getIndexBasedEvaluations",
    "data": {
        "mandators": [<mandatorUid>]
    }
}
```

- "data" (mandatory): See below.
- "data:mandators" (mandatory): A list of Mandator objects (identified by their uids) to retrieve index based evaluations from.

Returns

A HTTP response with the following content:

```
{
    "indexBasedEvaluations": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


162 GetIndexBasedEvaluationsTable

Detailed Description

GetIndexBasedEvaluationsTable retrieves the table of all IndexBasedEvaluation objects.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getIndexBasedEvaluationsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

163 GetIndexedAttributes

Detailed Description

GetIndexedAttributes returns all indexes, which are used in any masterdata for all mandators, where the user has privilege to view masterdata entries. Indexes on encrypted attributes are excluded, if the user has no privilege to view unmasked data.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getIndexedAttributes"
}
```

Returns

A HTTP response with the following content:

```
{
    "indexAttributes": [
        {
            "uid": <indexUid>,
            "revision": <revisionUid>,
            "mandator": <mandatorUid>,
            "name": "<indexName>",
            "attribute": {
                "type": "<attributeType>",
                "format": "<format>" | "None",
                "decimals": <decimals>,
                "length": <length>
            },
            ... ..
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

164 GetIndexes

Detailed Description

GetIndexes retrieves a list of Index objects from the server. The request parameters can be used to limit the Mandator and/or the Revision to be considered. The results also depend on the privileges of the requesting User.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getIndexes",
  "type": <type>,
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "selectedMandators": [<mandatorUid>]
  }
}
```

- "type" (mandatory): Controls the general behaviour of the retrieval. One of ["caseQueue", "checkIndex", "collusion", "complianceList", "counter", "counterGeneration", "doubletDetection", "fraudFlag", "indexBasedEvaluation", "job", "masterdataAssociated", "merchantMonitoringTemplate", "merging", "pattern", "precedent", "query", "rebuildIndex", "resetIndex", "sai", "setXdcSizes"].
- "mandator" (optional): A Mandator (identified by its uid) from which indexes should be retrieved. Usage depends on "type".
- "revision" (optional): A Revision (identified by its uid) from which indexes should be retrieved. Usage depends on "type".
- "data" (optional): Only used for type "job".
- "data:selectedMandators" (optional): A list of Mandator objects (identified by their uids) to retrieve indexes from.

Returns

A HTTP response with the following content:

```
{
  "indexes": [
    ...
  ],
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

165 GetIndexesTable

Detailed Description

GetIndexesTable retrieves a table of Index objects from a given Revision. Either inherited or own indexes can be retrieved. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request variables that have been passed to the server

```
{
    "request": "getIndexesTable",
    "type": <type>,
    "revision": <revisionUid>
}
```

- "type" (optional): When not set or set to a value different from "inherited" only indexes owned by the revision are retrieved; otherwise only inherited indexes are retrieved
- "revision" (mandatory): The Revision (identified by its uid) to retrieve indexes from.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

166 GetInputsTable

Detailed Description

GetInputsTable retrieves a table of Attribute objects used as input attributes in a given Revision. Either inherited or own inputs can be retrieved. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getInputsTable",
    "type": <type>,
    "revision":<revision>
}
```

- "type" (optional): When not set or set to a value different from "inherited" only inputs owned by the Revision are retrieved; otherwise only inherited inputs are retrieved.
- "revision" (mandatory): The Revision (identified by its uid) to retrieve inputs from.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

167 GetInstances

Detailed Description

GetInstances retrieves a list of IrisInstance objects. The list contains all instances of the Safer Payments cluster. When using the request parameter "checkInstanceStatus" the output is limited to instances that are reachable.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getInstances",
    "data": {
        "checkInstanceStatus": true|false
    }
}
```

- "checkInstanceStatus": limit the retrieved instances to only those that are reachable.

Returns

A HTTP response with the following content:

```
{
    "instances": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

168 GetInstancesTable

Detailed Description

GetInstancesTable retrieves a table of all IrisInstance objects in the Safer Payments cluster.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getInstancesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "autoRefresh": <autoRefreshInterval>,
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

169 GetInvestigationReportsTable

Detailed Description

GetInvestigationReportsTable retrieves a table of all InvestigationReport objects. The requesting User must have the privilege to view and generate already defined reports.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getInvestigationReportsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <investigatoInReportUid>,
            "<mandatorName>",
            "<reportName>",
            "<comment>",
            "<lastChangedByUserName>",
            <lastChangedTimestamp>
        ],
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


170 GetIsRulesetEnabled

Detailed Description

GetIsRulesetEnabled checks if a ruleset is enabled.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getIsRulesetEnabled",
    "uid": <rulesetUid>,
    "revision": <revisionUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "rulesetEnabled": 0|1,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

171 GetKeyPerformanceIndicatorsTable

Detailed Description

GetKeyPerformanceIndicatorsTable returns a table of all key performance indicators the user can view.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getKeyPerformanceIndicatorsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

172 GetKeysTable

Detailed Description

GetKeysTable returns the table data for encryption keys. It also returns the status of the masterkey and if it is changeable.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getKeysTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ...
    ],
    "autoRefresh": <autoRefreshSetting>,
    "changeMasterKeyStatus": <changeMasterKeyStatus>,
    "masterKeyChangeable": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

173 GetLastUrid

Detailed Description

GetLastUrid returns the last URID created in the instance.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{  
    "request": "getLastUrid"  
}
```

Returns

A HTTP response with the following content:

```
{  
    "urid": [<urid>],  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

174 GetLastUserAction

Detailed Description

GetLastUserAction returns the timestamp of the last action performed by the logged on user. The timestamp value is in epoch time.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getLastUserAction"
}
```

Returns

A HTTP response with the following content:

```
{
    "lastUserAction": <timestamp>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

175 GetListsTable

Detailed Description

GetListsTable returns a table of all Lists in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getListsTable",
    "type": "inherited" | "own",
    "revision": <revisionUid>
}
```

- "type" (mandatory): for inherited or revisions' own lists table
- "revision" (mandatory): uid of revision containing the lists

Returns

A HTTP response with the following content:

```
{
    "tableContent": [{
        "uid": <uid>,
        "name": "<name>"
    },
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

176 GetMandators

Detailed Description

GetMandators is used to retrieve mandators for various request types.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMandators",
    "type": "<type>",
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "type": The type of request. Available values for "streamType" are:

```
"analysis" (requires "mandator")
"groupByQueryDataSelection" (requires "mandator")
"queryDataSelection" (requires "mandator")
"simulation" (requires "mandator")
"simulationQuery" (requires "mandator")
"ruleGeneration" (requires "mandator" and "revision")
"counterGeneration" (requires "mandator" and "revision")
"user"
"reportGeneration"
"grants"
"reportUseMandatorData"
"chartSubMandator"
"statusAlarmIndicatorSubMandator"
"reportGenerationJob"
"commonPointQuery"
"ticker"
"caseromQuery"
"mandator"
"headMandators"
"query"
"groupByQuery"
"report"
"caseQueue"
"roles"
"notification"
"textModule"
"caseAction"
"externalQuery"
"reminder"
"caseCloseCode"
"statusAlarmIndicator"
"chart"
"caseFilter"
```

```
"caseCreation"  
"caseGroup"  
"definedRisk"  
"compliance"  
"merchantMonitoringRule"  
"memoryManagement"  
"rebuildIndex"  
"resetIndex"  
"checkIndex"  
"setXdcSizes"  
"fixMillisecondsMapping"
```

- "mandator": Members of the selected mandator will be returned.
- "revision": The revision should be supplied for revision components.

Returns

A HTTP response with the following content:

```
{  
  "mandators": [{  
    "uid": <mandatorUid>,  
    "name": "<name>",  
    "maxMemoryForSimulationGB": <simulationMemorySize>  
  },  
  ...  
],  
  "responseStatus": ["<status>", "<feedbacktext>"],  
  "reloadUserProfile": true|false  
}
```


177 GetMandatorSelectionTable

Detailed Description

GetMandatorSelectionTable returns a table of all mandators the user may access revisions from.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMandatorSelectionTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

178 GetMandatorSelectionTree

Detailed Description

GetMandatorSelectionTree returns a tree of all mandators the user may access.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMandatorSelectionTree",
    "type": "adminMandators" | ""
}
```

- "type" (optional): switch privilege check and if "conditions" are also added to stream

Returns

A HTTP response with the following content:

```
{
    "tree": [{
        "uid": <mandatorUid>,
        "label": "<mandatorLabel>",
        "champion": true|false,
        "revisions": <totalNumberOfRevisions>,
        "title": "<mandatorTitle>",
        "conditions": "<mandatorConditions>",
        "expanded": true|false,
        "depth": <mandatorDepth>,
        "selectable": 0|1,
        "children": [
            ... ..
        ]
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

179 GetMandatorsTable

Detailed Description

GetMandatorsTable returns a table of all mandators that the user can access.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMandatorsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        {
            <mandatorUid>,
            "<headMandatorName>" | "ROOT",
            "<mandatorName>",
            "<comment>",
            <numberOfRevisions>,
            <hasChampion>"Yes" | "No",
            <simulationMemoryInGB>,
            "<lastChangedByUserName>",
            <lastChangedOnTimestamp>,
            "<conditions>"
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

180 GetMappingsTable

Detailed Description

GetMappingsTable returns a table of all mappings in a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMappingsTable",
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "mandator" (mandatory): uid of mandator that owns the revision. Used to get the champion to access attributes.
- "revision" (mandatory): uid of revision that owns the mappings.

Returns

A HTTP response with the following content:

```
{
    "header": [{
        "name": "<name>",
        "type": "<type>",
        "messageUid": -1
    },
    ...
    ],
    "dataSource": [{
        "key": "<value>"
    },
    ...
    ],
    "tableContent": [
        [ ... ],
        ...
    ],
    "mappingsUids": [{
        "attributeUid": <attributeUid>,
        "messageUid": <messageTypeUid>,
        "mappingUid": <mappingUid>
    },
    ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

181 GetMasterdataDeepInformation

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getMasterdataDeepInformation",
  "uid": <uid>
  "mandator": <mandatorUid>
  "data": {
    "parameter": <indexValue>,
    "urid": <record of indexValue>,
    "isTargetValue": true | false
  }
}
```

- "uid" (mandatory): uid of the index
- "mandator" (mandatory): uid of mandator
- "data: parameter" (optional): value of the index as text
- "data: urid" (optional): record which contains value
- "data: isTargetVale" (optional): search on target or source index

Returns

A HTTP response with the following content:

```
{
  "uid": <indexUid>,
  "mandator": <mandatorUid>,
  "parameter": "<parameter>",
  "indexAttribute": <indexAttribute>,
  "type": "<indexAttributeType>",
  "format": <indexAttributeFormat>,
  "tableHeader": [ ... ... ],
  "tableContent": [ ... ... ],
  "multipleValues": [ ... ... ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

182 GetMasterdatas

Detailed Description

GetMasterdatas is used to receive a list of masterdatas defined in a current champion

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMasterdatas",
    "type": <type>,
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "type" (mandatory): caseQueue | caseSelection | ruleAction
- "mandator" (optional): the uid of the mandator
- "revision" (optional): the uid of the revision

Returns

A HTTP response with the following content:

```
{
    "uid": <uid>,
    "name": "<name>",
    "comment": "<comment>",
    "index": <index uid>,
    "conditions": [<Condition>, ...],
    "deletionConditions": [<Condition>, ...],
    "insertTransaction": true | false,
    "multipleValues": true | false,
    "multipleValuesCapacity": <multipleValuesCapacity>,
    "associatedIndexUid": <associatedIndexUid>,
    "relationshipAttributes": [<Attribute>, ...],
    "attribute": <attribute uid>,
    "attributeData": {<attribute data>},
    "targetAttribute": <attribute uid>,
    "responseStatus": [<status>, "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

183 GetMasterKeysTable

Detailed Description

GetMasterKeysTable is used to retrieve the content of master keys table.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMasterKeysTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            <id>,
            "<status>",
            <numberOfKeys>,
            <activationTimestamp>,
            "<enteredByUserName>"
        ],
        ... ..
    ],
    "autoRefresh": <autoRefreshSetting>,
    "lastActiveMasterKey": <lastActiveMasterKeyId>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

184 GetMemoryManagementTable

Detailed Description

GetMemoryManagementTable retrieves a table with memory usage information for a given list of Mandator objects. The requesting User must have the privilege to view the memory consumption of simulations.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getMemoryManagementTable",
  "data": {
    "mandators": [<mandatorUid>],
    "users": [<userId>]
  }
}
```

- "data" (optional): Contains mandators and users for which to retrieve information about memory consumption.
- "data:mandators" (optional): A list of Mandator objects for which to retrieve information about memory consumption.
- "data:users" (optional): A list of User objects for which to retrieve information about memory consumption.

Returns

A HTTP response with the following content:

```
{
  "memoryConsumption": [[
    "<mandatorName>",
    "<memoryUsedInGB>",
    "<memoryLimitInGB>"
  ],
  ...
],
  "totalUsedMemoryMandators": <totalUsedMemoryMandators>,
  "totalAvailableMemoryMandators": <totalAvailableMemoryMandators>,
  "tableContent": [
    ...
  ],
  "totalUsedMemoryUsers": <totalUsedMemoryUsers>,
  "totalAvailableMemoryUsers": <totalAvailableMemoryUsers>,
  "refresh": <autoRefreshSetting>,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


185 GetMerchantMonitoringTemplate

Detailed Description

GetMerchantMonitoringTemplate retrieves a MerchantMonitoringTemplate.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getMerchantMonitoringTemplate",
  "data": {
    "templateKey": "<templateKey>"
  }
}
```

- "data" (mandatory): Contains the key of the MerchantMonitoringTemplate.
- "data:templateKey" (mandatory): Identifies the MerchantMonitoringTemplate to retrieve.

Returns

A HTTP response with the following content:

```
{
  "merchantMonitoringTemplate": {
    "name": "<templateName>",
    "index": <indexUid>|-1,
    "mandator": <mandatorUid>|-1,
    "periodsBack": "<numberOfPeriodsBack>",
    "chargebackProfile": <chargebackProfile>,
    "totalSalesProfile": <totalSalesProfile>,
    "minimumNumberOfChargebacks": <minimumNumberOfChargebacks>,
    "minimumNumberOfTotalSales": <minimumNumberOfTotalSales>,
    "minimumRatioChargebackTotalBp": <minimumRatioChargebackTotalBp>,
    "numberOfConsecutivePeriodsToMeet": <numberOfConsecutivePeriodsToMeet>
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

186 GetMerchantMonitoringTemplates

Detailed Description

GetMerchantMonitoringTemplates retrieves a list of all MerchantMonitoringTemplate objects.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMerchantMonitoringTemplates"
}
```

Returns

A HTTP response with the following content:

```
{
    "merchantMonitoringTemplates": [{
        "templateKey": "<templateKey>",
        "name": "<templateName>"
    },
    ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

187 GetMergingsTable

Detailed Description

GetMergingsTable retrieves a table of Merging objects from a given Revision. Either inherited or own mergings can be retrieved. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMergingsTable",
    "type": <type>,
    "revision":<revisionUid>
}
```

- "type" (optional): When not set or set to a value different from "inherited" only mergings owned by the Revision are retrieved, otherwise only inherited mergings are retrieved.
- "revision" (mandatory): The Revision (identified by its uid) to retrieve mergings from.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

188 GetMessageReport

Detailed Description

GetMessageReport retrieves the report for a given Message. The requesting User must have the privilege to view messages.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMessageReport",
    "uid": <messageUid>
}
```

- "uid" (mandatory): The Message (identified by its uid) to retrieve the report for.

Returns

A HTTP response with the following content:

```
{
    "messageReport": {
        "name": "<messageTypeName>",
        "comment": "<comment>",
        "mtid": <messageTypeId>,
        "messageType": "<messageType>",
        ["offline" | "online"]: {
            ...
        }
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

189 GetMessages

Detailed Description

GetMessages lists all messages for the specified type.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMessages",
    "type": "Revision" | "Job" | "Queue" | "Reminder",
    "revision": <revisionUid>
}
```

- "type" (mandatory): The type of request.
- "revision" (optional): For request type "Revision" the revision has to be supplied.

Returns

A HTTP response with the following content:

```
{
    "messages": [{
        "streamType": "message",
        "uid": <messageUid>,
        "name": "<messageName>",
        "mtid": <mtidValue>,
        "mayCreateTransactionRecords": true|false
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

190 GetMessagesReport

Detailed Description

GetMessagesReport retrieves reports for all Message objects. The requesting User must have the privilege to view messages.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMessagesReport"
}
```

Returns

A HTTP response with the following content:

```
{
    "messagesReport": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

191 GetMessagesTable

Detailed Description

GetMessagesTable retrieves a table containing all Message objects. The requesting User must have the privilege to view messages.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMessagesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <messageUid>,
        "<messageName>",
        "<comment>",
        <MTID>,
        <createTransactionRecords>true|false,
        "<type>",
        "<lastChangedByUserName>",
        "<lastChangedOnTimestamp>"
    ],
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

192 GetMetaAttributeAmount

Detailed Description

GetMetaAttributeAmount retrieves meta Attribute "Amount" from root Mandator's champion Revision. The unit and amount of decimals of the meta Attribute are retrieved. The requesting User must have the privilege to view the model tab.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMetaAttributeAmount"
}
```

Returns

A HTTP response with the following content:

```
{
    "metaAttributeAmount": {
        "dimension": "<unit>",
        "decimals": <decimals>
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


193 GetMetaAttributeMaximum

Detailed Description

GetMetaAttributeMaximum retrieves the maximum value of a given meta Attribute. The meta Attribute is either taken from the root Mandator's champion Revision or from the champion Revision of the Mandator supplied in the request.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMetaAttributeMaximum",
    "type": <type>,
    "mandator": <mandatorUid>,
    "data": {
        "metaAttribute": <metaAttribute>
    }
}
```

- "type" (optional): Can be left out or one of ["", "caseQueue", "intercept"].
- "mandator" (optional): The Mandator (identified by its uid) to retrieve the meta Attribute's maximum value from. Only used when "type" is left out or empty.
- "data" (mandatory): Contains the name of the meta Attribute.
- "data:metaAttribute" (mandatory): Name of the meta Attribute e.g. "Account", "Amount", "Fraud" etc.

Returns

A HTTP response with the following content:

```
{
    "metaAttributeMax": <maxValue>,
    "responseStatus": [ "<status>", "<feedbacktext>" ],
    "reloadUserProfile": true|false
}
```

194 GetMissedCasesReportsTable

Detailed Description

GetMissedCasesReportsTable retrieves a table of all MissedCasesReport objects. The requesting User must have the privilege to view and generate already defined reports.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMissedCasesReportsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <missedCasesReportUid>,
        "<mandatorName>",
        "<reportName>",
        "<comment>",
        "<lastChangedByUserName>",
        <lastChangedOnTimestamp>
    ],
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

195 GetModelComponents

Detailed Description

GetModelComponents retrieves model components for a given revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getModelComponents",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which model components should be retrieved.

Returns

A HTTP response with the following content:

```
{
    "rulesets": [{
        "uid": <rulesetUid>,
        "name": "<rulesetName>"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

196 GetModelComponentsTable

Detailed Description

GetModelComponentsTable returns a table of all model components for a given revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getModelComponentsTable",
    "type": "inherited|",
    "revision": <revisionUid>
}
```

- "type": Indicates whether or not inherited components should be returned.
- "revision": The revision for which model components should be returned.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

197 GetModelling

Detailed Description

GetModelling retrieves the modelling information from a given Attribute. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getModelling",
  "uid": <attributeUid>,
  "revision": <revisionUid>
}
```

- "uid" (mandatory): The Attribute (identified by its uid) to retrieve modelling information from.
- "revision" (mandatory): The Revision (identified by its uid) to retrieve the Attribute's modelling information from.

Returns

A HTTP response with the following content:

```
{
  "modelling": {
    "attribute": <attributeUid>,
    "name": "<attributeName>",
    "origin": "<origin>",
    "type": "<type>",
    "dimension": "<dimension>",
    "decimals": <decimals>,
    "simulationRequired": true|false,
    "testEnabled": true|false,
    "simulationEnabled": true|false,
    "analysisEnabled": true|false,
    "counterGenerationEnabled": true|false,
    "ruleGenerationEnabled": true|false,
    "attributeUsage": "<attributeUsage>",
    "maxComputedIndicators": <maxComputedIndicators>,
    "maxSelectedIndicators": <maxSelectedIndicators>,
    "linearFrom": <linearFrom>,
    "linearStep": <linearStep>,
    "linearTo": <linearTo>,
    "customCategories": [
      ... ..
    ],
  },
}
```

```
    "customIntervalFrom": [
        ... ..
    ],
    "customIntervalTo": [
        ... ..
    ]
},
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

198 GetModellingsTable

Detailed Description

GetModellingsTable retrieves the table of all Modelling objects for a given Revision. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getModellingsTable",
  "revision": <revisionUid>,
  "data": {
    "showInheritedAttributes": true|false,
    "selectedOrigins": [<origin>],
    "selectedMessages": [<messageUid>]
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) to retrieve the modellings table from.
- "data" (optional): Additional values to filter the displayed attributes.
- "data:showInheritedAttributes" (optional): When set to "false" only attributes owned by the Revision are retrieved; otherwise inherited Attributes are retrieved as well.
- "data:selectedOrigins" (optional): A list of attribute origins. Limits displayed attributes to those that have at least one of the defined origins. Possible values are: ["List", "DeviceIdentification", "Precedent", "Profile", "Pattern", "Event", "Counter", "Collusion", "Formula", "Output", "Input"].
- "data:selectedMessages" (optional): A list of Message objects (identified by their uid). Limits displayed input and output attributes to those that have a mapping in at least one the messages. Only used when the respective origin was not selected.

Returns

A HTTP response with the following content:

```
{
  "tableContent" :[
    ... ..
  ],
  "totalAttributes": <totalNumberOfAttributes>,
  "testEnbled": <totalNumberOfAttributesWithTestEnabled>,
  "simulationEnabled": <totalNumberOfAttributesWithSimulationEnabled>,
  "analysisEnabled": <totalNumberOfAttributesWithAnalysisEnabled>,
  "ruleGenerationEnabled": <totalNumberOfAttributesWithRuleGenerationEnabled>,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

199 GetMultipleRulesFiringStatistics

Detailed Description

GetMultipleRulesFiringStatistics retrieves Rule firing statistics from a given Revision to analyse overlap. The retrieved statistics show the number of transactions that fulfilled the conditions of multiple rules, indicating an overlap in the Rule conditions which should generally be avoided.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getMultipleRulesFiringStatistics",
  "revision": <revisionUid>,
  "data": {
    "maxNumberOfRules": <maxNumberOfRules>,
    "analysisId": <analysisId>
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) to retrieve Rule firing statistics from.
- "data" (mandatory): Additional values to limit the maximum number of displayed rules and select an Analysis to use.
- "data:maxNumberOfRules" (mandatory): The maximum number of rules for which to retrieve firing statistics.
- "data:analysisId" (mandatory): The Analysis (identified by its uid) to use as a base for the Rule firing statistics.

Returns

A HTTP response with the following content:

```
{
  "multipleRulesFiringNames": [
    ...
  ],
  "multipleRulesFiring": [
    ...
  ],
  "multipleRulesInterceptNames": [
    ...
  ],
  "multipleRulesIntercept": [
    ...
  ],
  "multipleRulesGeneratingAlarmsNames": [
    ...
  ]
}
```



```
],
"multipleRulesGeneratingAlarms": [
    ...
],
"multipleRulesGeneratingNotificationsNames": [
    ...
],
"multipleRulesGeneratingNotifications": [
    ...
],
"multipleRulesGeneratingRemindersNames": [
    ...
],
"multipleRulesGeneratingReminders": [
    ...
],
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

200 GetMyWorkingQueues

Detailed Description

GetMyWorkingQueues returns all public and private working queues the requesting user is allowed to see.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getMyWorkingQueues"
}
```

Returns

A HTTP response with the following content:

```
{
    "workingQueues": {
        "public": [{
            "uid": <workingQueueUid>,
            "name": "<workingQueueName>"
        },
        ... ..
    ],
        "private": [{
            "streamType": "user",
            "uid": <userId>,
            "enabled": true|false,
            "login": "<loginName>",
            "name": "<userName>"
        },
        ... ..
    ]
},
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

201 GetNextCase

Detailed Description

GetNextCase retrieves the next Case for the requesting User. The requesting User must have the privilege to investigate cases. Cases are retrieved from the User's private WorkingQueue first, then from the public working queues in order of the mandators and within a Mandator depending on the priority of the queues themselves.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getNextCase"
}
```

Returns

A HTTP response with the following content:

```
{
    "uid": <caseUid>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

202 GetNotifications

Detailed Description

GetNotifications is used to request available notifications.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getNotifications",
    "type": <type>,
    "mandator": <mandator uid>
}
```

- "type" (mandatory): "query" | "caseSelection"
- "mandator" (mandatory): the uid of the mandator

Returns

A HTTP response with the following content:

```
{
    "notifications": [{
        "uid": <uid>,
        "name": "<name>"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

203 GetNotificationsTable

Detailed Description

GetNotificationsTable is used to get the data for notifications table. Privilege to view/edit notifications is needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getNotificationsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [{
        "uid": <uid>,
        "id": <id>,
        "enabled": true | false,
        "name": "<name>"
    },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

204 GetOptimizationAnalysis

Detailed Description

GetOptimizationAnalysis retrieves the Rule optimization analysis from a given Revision. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getOptimizationAnalysis",
    "revision": <revisionUid>,
    "data": {
        "analysisId": <analysisId>
    }
}
```

- "revision" (mandatory): The Revision (identified by its uid) from which to retrieve the Rule optimization analysis.
- "data" (optional): Identifies the Analysis to use.
- "data:analysisId" (optional): The Analysis (identified by its uid) to use as a base for the Rule optimization analysis.

Returns

A HTTP response with the following content:

```
{
    "statistics": [
        ... ..
    ],
    "responseStatus": [<status>, "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

205 GetOutgoingChannelConfigurations

Detailed Description

GetOutgoingChannelConfigurations is used to request available outgoing channel configurations for a specific outgoing channel configuration template (e.g. case action, notification, external query, generate report job). Privilege to change outgoing channel configuration template is needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getOutgoingChannelConfigurations",
    "type": "<type>"
}
```

- "type" (mandatory): "caseAction" | "externalQuery" | "reportGenerationJob" | "notification"

Returns

A HTTP response with the following content:

```
{
    "outgoingChannelConfigurations": [{
        "uid": <uid>,
        ... ..
        "name": "<name>"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

206 GetOutgoingChannelConfigurationTable

Detailed Description

GetOutgoingChannelConfigurationTable is used to get the data for outgoing channel configuration table. Privilege to view/edit notifications is needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getOutgoingChannelConfigurationTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [{
        "uid": <uid>,
        ... ..
        "name": "<name>"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


207 GetOutputsTable

Detailed Description

GetOutputsTable returns a table of all output attributes in a revision. It can be chosen, if own outputs (defined in this mandator) or inherited outputs (defined in head mandators) should be returned.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getOutputsTable",
    "type": "inherited" | "own",
    "revision": <revisionUid>
}
```

- "type" (optional): defines if inherited or own outputs are returned. If empty "own" will be set.
- "revision" (mandatory): Revision (identified by uid) for which the outputs should be listed

Returns

A HTTP response with the following content:

```
{
    "tableContent": [ [
        ... ..
    ],
    ... ..
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

208 GetPasswordSafes

Detailed Description

GetPasswordSafes returns all password safes.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPasswordSafes"
}
```

Returns

A HTTP response with the following content:

```
{
    "passwordSafes": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

209 GetPasswordSafeTable

Detailed Description

GetPasswordSafeTable: Return a table of all password safes.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPasswordSafeTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

210 GetPatternsTable

Detailed Description

GetPatternsTable returns a table of Patterns in a revision. It can be chosen, if own patterns (defined in this mandator) or inherited patterns (defined in head mandators) should be returned.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPatternsTable",
    "type": "own" | "inherited",
    "revision": <revisionUid>
}
```

- "type" (optional): defines if inherited or own patterns are returned. If empty "own" will be set.
- "revision" (mandatory): Revision (identified by uid) for which the patterns should be listed

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <revisionUid>,
        <patternUid>,
        "<patternName>",
        "<comment>",
        "<mandatorName>",
        "<indexName>",
        "<computationConditions>",
        <numberOfStencils>
    ]],
    ... ..
],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

211 GetPciDssReport

Detailed Description

GetPciDssReport returns the PCI DSS report for the current configuration. The PCI DSS report will contain, relevant to the current configuration, information of how to make the configuration and model compliant with PCI DSS requirements.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPciDssReport "
}
```

Returns

A HTTP response with the following content:

```
{
    "pciDssReport": [ { "type": "<type>", "variables": [ ... ... ] }, ... ... ],
    "generatedOn": <now as timestamp>,
    "responseStatus": [ "<status>", "<feedbacktext>" ],
    "reloadUserProfile": true|false
}
```

212 GetPossibleAmountAttributes

Detailed Description

GetPossibleAmountAttributes lists all attributes that may be used as amount attributes.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPossibleAmountAttributes",
    "revision": <revisionUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "passibleAmountAttributes": [{
        "uid": <attributeUid>,
        "name": "<attributeName>",
        "mandator": <mandatorUid>,
        "revision": <revisionUid>,
        "mdcStored": true|false,
        "origin": "Input" | "Output",
        "type": "Numeric",
        "encrypted": true|false,
        "length": <attributeLength>,
        "decimals": <decimals>,
        "dimension": "<dimension>",
        "categoriesEnabled": true|false,
        "categories": [
            ... ..
        ],
        "usedInMergingConclusion": true|false,
        "metaAttribute": "None" | "<metaAttributeType>",
        "format": "None" | "<format>"
    },
    ... ..
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

213 GetPrecedentsTable

Detailed Description

GetPrecedentsTable retrieves a table of Precedent objects from a given Revision. Either inherited or own precedents can be retrieved. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPrecedentsTable",
    "type": <type>,
    "revision": <revisionUid>
}
```

- "type" (optional): When not set or set to a value different from "inherited" only precedents owned by the Revision are retrieved; otherwise only inherited precedents are retrieved.
- "revision" (mandatory): The Revision (identified by its uid) to retrieve precedents from.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

214 GetPreference

Detailed Description

GetPreference returns a preference setting of a certain type for the current user. User preference settings store user specific settings like table view options.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPreference",
    "type": "<type>"
}
```

- "type" (mandatory): lookup key for preference value

Returns

A HTTP response with the following content:

```
{
    "preference": ... ..,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


215 GetPreProcessingRulesets

Detailed Description

GetPreProcessingRulesets lists all preprocessing resultsets for a given revision, which includes inherited and own rulesets.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPreProcessingRulesets",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): Revision in which the rulesets are listed

Returns

A HTTP response with the following content:

```
{
    "rulesets": [{
        "uid": <rulesetUid>,
        "name": "<rulesetName>",
        "comment": "<comment>",
        "revision": <revisionUid>
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

216 GetPreProcessingRulesetsTable

Detailed Description

GetPreProcessingRulesetsTable returns a table of all preprocessing rulesets for a given revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "GetPreProcessingRulesetsTable",
    "type": "inherited|",
    "revision": <revisionUid>
}
```

- "type": Indicates whether or not inherited components should be returned.
- "revision": The revision for which model components should be returned.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [ ... ... ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

217 GetPrintView

Detailed Description

GetPrintView is used to get a print view in json format of various revision elements. The normal use case is to get a complete revision model containing all its elements and inherited elements.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPrintView",
    "uid": <revisionUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "data": {
        ... ..
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

218 GetProcessPriorities

Detailed Description

GetProcessPriorities retrieves a list of all available process priorities. The available process priorities depend on the operating system for which IBM Safer Payments was build. The requesting User must have the privilege to view the system configuration.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getProcessPriorities"
}
```

Returns

A HTTP response with the following content:

```
{
    "data": [
        "<priority1>",
        "<priority2>",
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

219 GetProfiles

Detailed Description

GetProfiles retrieves a list of Profile objects from a given Revision. The requesting User must have either the privilege to view merchant monitoring rules or to view index based evaluations.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getProfiles",
    "uid": <indexUid>,
    "revision": <revisionUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (optional): An Index (identified by its uid) to limit the retrieved calendar profiles to those belonging to it.
- "revision" (optional): The Revision (identified by its uid) from which to retrieve calendar profiles. If not used, the Mandator's next champion is taken.
- "mandator" (mandatory): The Mandator (identified by its uid) whose next champion Revision is used if no Revision

Returns

A HTTP response with the following content:

```
{
    "profiles": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

220 GetProfilesTable

Detailed Description

GetProfilesTable retrieves a table of Profile objects from a given Revision. Either inherited or own calendar profiles can be retrieved. The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getProfilesTables",
    "type": <type>,
    "revision":<revisionUid>
}
```

- "type" (optional): When not set or set to a value different from "inherited" only calendar profiles owned by the Revision are retrieved; otherwise only inherited calendar profiles are retrieved.
- "revision" (mandatory): The Revision (identified by its uid) to retrieve calendar profiles from.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

221 GetPythonFunctions

Detailed Description

GetPythonFunctions lists all available Python functions for the specified type. The available values for the "type": "conclusion".

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getPythonFunctions",
    "type": "<type>",
    "revision": <revisionUid>
}
```

- "type" (mandatory): "ruleConclusion" | "ruleConclusionPreProcessing" | "formula"
- "revision" (mandatory): uid of revision

Returns

A HTTP response with the following content:

```
{
    "pythonFunctions": [{
        "name": "<pythonFunctionName>",
        "helpText": "<pythonFunctionHelpText>",
        "arguments": ["argument1", "argument2", ...]
    },
    ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

222 GetQueries

Detailed Description

GetQueries is used to get a list of queries, for example to use them in drop downs.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getQueries",
    "mandator": <mandatorUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "queries": [
        {
            "streamType": "investigationQuery",
            "uid": <queryUid>,
            "name": <queryName>
        },
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


223 GetQueryOptions

Detailed Description

GetQueryOptions shows settings for Query.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getQueryOptions"
}
```

Returns

A HTTP response with the following content:

```
{
    "queryOptions": {
        "maxRecordsWarning": <maxRecordsWarning>,
        "maxRecordsLimit": <maxRecordsLimit>
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

224 GetQueryResults

Detailed Description

GetQueryResults is used to fetch a computed query result, if it is already finished.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getQueryResults",
  "uid": <queryUid>,
  "data": {
    "resultId": <resultId>
  }
}
```

- "resultId": it is generated by the request "executeQuery".

Returns

A HTTP response with the following content:

```
{
  "queryResult": {
    "name": "<queryName>",
    "comment": "<comment>",
    "header": [
      {
        "name": "<attributeName>",
        "attributeUid": <attributeUid>,
        "align": "<align>",
        "decimals": <decimals>,
        "hyperlinkQueryUid": [],
        "commonPointQueryUid": [],
        "type": "<attributeType>",
        "format": "<format>"
      },
      ... ..
    ],
    "tableContent": [
      ... ..
    ]
  },
  "indexValueUrid": <urid>,
  "running": true|false,
  "recordsTotal": <recordsTotal>,
  "recordsFinished": true|false,
```

```
"duration": <duration>,
"refresh": true|false,
"highlightCppAttributes": <highlightCppAttributes>,
"amountDimension": "<amountDimension>",
"amountDecimals": "<amountDecimals>",
"statistics": {
  "transaction": {
    "genuine": {
      "records": <numberOfRecords>,
      "amount": <amount>
    },
    "fraud": {
      "records": <numberOfRecords>,
      "amount": <amount>
    }
  }
},
>falsePositives": <falsePositiveRate>,
"numberOfAccounts": <numberOfAccounts>,
"extractTemplate": "<extractTemplate>",
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

225 GetQuickSearchCasesTable

Detailed Description

GetQuickSearchCasesTable returns a table of filtered cases the user can view. Requires a previous setQuickSearchCasesTablePreference that sets the search value. This request is similar to the request "getCasesTable". The cases are filtered by the quick search criteria delivered in the "data" object of the request.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getQuickSearchCasesTable",
  "data": {
    "caseSearch": true|false
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "header": [
    {
      "uid": "<uid>",
      "name": "<name>",
      "align": "<align>",
      "type": "<type>",
      "decimals": <decimals>,
      "format": "<format>"
    },
    ... ..
  ],
  "tableContent": [
    {
      "uid":<caseId>,
      "Queue": <caseQueueName>,
      "Generated": <generatedTimestamp>,
      "Score": <scoreValue>,
      "Hits": <numberOfHits>,
      "Status": "<status>",
      "FraudStatus": "<fraudStatus>",
      "CloseCodeName": "<caseCloseCodeName>",
      "User": <userId>,
      "UserName": "<userName>",
      "LastUserAction": <lastUserActionTimestamp>,
      "MandatorName": "<mandatorName>",
      "Mandator": <mandatorId>,
    }
  ]
}
```

```
        "CppName": "<cppName>",
        "Cpp": "cppCode",
        ... ..
    },
    ... ..
],
"responseStatus": ["<status>", "feedbacktext"],
"reloadProfile": true|false
}
```

226 GetRealtimeInterceptionCodes

Detailed Description

GetRealtimeInterceptionCodes returns all realtime interception codes.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRealtimeInterceptionCodes"
}
```

Returns

A HTTP response with the following content:

```
{
    "realtimeInterceptionCodes": [{
        "uid": <realtimeInterceptionCodeUid>,
        "name": "<realtimeInterceptionCodeName>",
        "from": <fromValue>,
        "to": <toValue>
    },
    ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

227 GetRealtimeInterceptionCodesTable

Detailed Description

GetRealtimeInterceptionCodesTable returns a table of all realtime interception codes.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRealtimeInterceptionCodesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        [
            ... ..
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

228 GetReferencesOfAttribute

Detailed Description

GetReferencesOfAttribute retrieves a list of all elements referencing a given Attribute. Attributes can be referenced by elements within the same Revision (own references) or by elements outside the Revision (foreign references). The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfAttribute",
    "uid": <attributeUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The Attribute (identified by its uid) to retrieve references of.
- "revision" (mandatory): The Revision (identified by its uid) to search the Attribute in.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ...
    ],
    "foreignReferences": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


229 GetReferencesOfCaseAction

Detailed Description

GetReferencesOfCaseAction retrieves a list of CaseAction references. The request parameters must be used to identify CaseAction and Mandator. Privilege to view case actions needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfCaseAction",
    "uid": <uid>,
    "mandator": <mandatoruid>
}
```

- "uid" (mandatory): uid of CaseAction
- "mandator" (mandatory): mandator of the (outgoing channel configuration) mandator.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [ ... ... ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

230 GetReferencesOfCaseCloseCode

Detailed Description

GetReferencesOfCaseCloseCode retrieves a list of CaseCloseCode references and a flag indicating if cases exist that are associated to the case close code. The request parameters must be used to identify CaseCloseCode and Mandator. Privilege to view case close codes needed.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
    "request": "getReferencesOfCaseCloseCode",
    "uid": <caseCloseCodeUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): Uid of the CaseCloseCode.
- "mandator" (mandatory): Uid of the Mandator.

Returns

A HTTP response with the following content:

```
{
    "hasCases": true (optional),
    "ownReferences": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

231 GetReferencesOfCaseQueue

Detailed Description

GetReferencesOfCaseQueue retrieves a list of all elements referencing a given case queue and a flag indicating if cases exist that are associated to the case queue. The requesting User must have the privilege to view case queues.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
  "request": "getReferencesOfCaseQueue",
  "uid": <caseQueueUid>,
  "mandator": <mandatorUid>,
}
```

- "uid" (mandatory): Uid of the case queue.
- "mandator" (mandatory): Uid of the Mandator.

Returns

A HTTP response with the following content:

```
{
  "hasCases": true (optional),
  "ownReferences": [
    ...
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

232 GetReferencesOfCaseState

Detailed Description

GetReferencesOfCaseState retrieves a list of CaseState references and a flag indicating if cases exist that are associated to the case state. The request parameters must be used to identify CaseState and Mandator. Privilege to view case states needed.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
  "request": "getReferencesOfCaseState",
  "uid": <caseStateUid>,
  "mandator": <mandatorUid>,
}
```

- "uid" (mandatory): Uid of the CaseState.
- "mandator" (mandatory): Uid of the Mandator.

Returns

A HTTP response with the following content:

```
{
  "hasCases": true (optional),
  "ownReferences": [
    ...
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

233 GetReferencesOfCaseWorkflow

Detailed Description

GetReferencesOfCaseWorkflow retrieves a list of CaseWorkflow references. The request parameters must be used to identify CaseWorkflow and Mandator. Privilege to view case workflows needed.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
  "request": "getReferencesOfCaseWorkflow",
  "uid": <caseWorkflowUid>,
  "mandator": <mandatorUid>
}
```

- "uid" (mandatory): Uid of the CaseWorkflow.
- "mandator" (mandatory): Uid of the Mandator.

Returns

A HTTP response with the following content:

```
{
  "ownReferences": [
    ...
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

234 GetReferencesOfChart

Detailed Description

GetReferencesOfChart retrieves a list of all key performance indicators referencing a given Chart. The requesting User must have the privilege to view the administration pages for charts and key performance indicators.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfChart",
    "uid": <chartUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): The Chart (identified by its uid) to retrieve references of.
- "mandator" (mandatory): The Mandator (identified by its uid) the Chart belongs to. Used to check the User's privilege.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

235 GetReferencesOfDefinedRiskList

Detailed Description

GetReferencesOfDefinedRiskList retrieves a list of all rules referencing a given DefinedRiskList in their RuleAction. The requesting User must have the privilege to view defined risk list definitions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfDefinedRiskList "
    "uid": <definedRiskListUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): The DefinedRiskList (identified by its uid) to retrieve references of.
- "mandator" (mandatory): The Mandator (identified by its uid) the DefinedRiskList belongs to.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

236 GetReferencesOfExternalQuery

Detailed Description

GetReferencesOfExternalQuery is used to get references of an external query.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
    "request": "getReferencesOfExternalQuery",
    "uid": <externalQueryUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): Uid of the ExternalQuery.
- "mandator" (mandatory): Uid of the Mandator.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


237 GetReferencesOfIndex

Detailed Description

GetReferencesOfIndex retrieves a list of all elements referencing a given Index. Indexes can be referenced by elements within the same Revision (own references) or by elements outside the Revision (foreign references). The requesting User must have the privilege to view models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfIndex",
    "uid": <indexUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The Index (identified by its uid) to retrieve references of.
- "revision" (mandatory): The Revision (identified by its uid) to search the Index in.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ... ..
    ],
    "foreignReferences": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

238 GetReferencesOfMasterdata

Detailed Description

GetReferencesOfMasterdata retrieves a list of references for a masterdata.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfMasterdata",
    "uid": <masterdataUid>,
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) which is used as base for the Masterdata reference search.
- "uid" (mandatory): uid of the Masterdata whose references are requested.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [<reference>],
    "foreignReferences": [<reference>],
    "responseStatus": [<status>, <message>],
    "reloadUserProfile": true | false
}
```

239 GetReferencesOfMessage

Detailed Description

GetReferencesOfMessage retrieves a list of references for a Message.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfMessage",
    "uid": <messageUid>,
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) which is used as base for the Message reference search.
- "uid" (mandatory): uid of the Message whose references are requested.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [<reference>],
    "responseStatus": [<status>, <message>],
    "reloadUserProfile": true | false
}
```

240 GetReferencesOfNotification

Detailed Description

GetReferencesOfNotification retrieves a list of references for a Notification.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfNotification",
    "uid": <notificationUid>,
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) which is used as base for the Notification reference search.
- "uid" (mandatory): uid of the Notification whose references are requested.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [<reference>],
    "responseStatus": [<status>, <message>],
    "reloadUserProfile": true | false
}
```

241 GetReferencesOfOutgoingChannelConfiguration

Detailed Description

GetReferencesOfOutgoingChannelConfiguration is used to get references of an outgoing channel configuration. Could be e.g. notifications, case actions, external queries or generate report jobs. Privilege to view outgoing channel configuration is needed.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
    "request": "getReferencesOfOutgoingChannelConfiguration",
    "uid": <outgoingChannelConfigurationUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): Uid of the OutgoingChannelConfiguration.
- "mandator" (mandatory): Uid of the Mandator.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

242 GetReferencesOfPasswordSafe

Detailed Description

GetReferencesOfPasswordSafe retrieves a list of references for a PasswordSafe.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfPasswordSafe"
    "uid": <PasswordSafe Uid>
}
```

- "uid" (mandatory): uid of the PasswordSafe

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [ <references> ],
    "responseStatus": [ "<status>", "<feedbacktext>" ],
    "reloadUserProfile": true|false
}
```

243 GetReferencesOfProfile

Detailed Description

GetReferencesOfProfile retrieves a list of references for a Profile.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfMessage",
    "uid": <Message Uid>,
    "revision": <Revision Uid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) which is used as base for the Profile reference search.
- "uid" (mandatory): uid of the Profile whose references are requested.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [<reference>],
    "responseStatus": [<status>, <message>],
    "reloadUserProfile": true|false
}
```

244 GetReferencesOfPythonModule

Detailed Description

GetReferencesOfPythonModule retrieves a list of Python module references. The request parameters must be used to identify Python module and Mandator. Privilege to view mandators needed.

Parameters

HttpRequest: Holds the request variables that have been passed to the server:

```
{
    "request": "getReferencesOfPythonModule",
    "uid": <pythonModuleUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): Uid of the PythonModule.
- "mandator" (mandatory): Uid of the Mandator.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


245 GetReferencesOfQuery

Detailed Description

GetReferencesOfQuery retrieves a list of references for a query.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfQuery"
    "uid": <Query Uid>
    "mandator": <Mandator Uid>
}
```

- "uid" (mandatory): Uid of Query
- "mandator" (mandatory): Uid of Mandator

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [<reference>],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

246 GetReferencesOfReminder

Detailed Description

GetReferencesOfReminder retrieves the references of an existing reminder.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfReminder",
    "uid": <reminderUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): The Reminder (identified by its uid) to request the references for.
- "mandator" (mandatory): The Mandator (identified by his uid) of the Reminder.

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

247 GetReferencesOfUserGroup

Detailed Description

GetReferencesOfUserGroup retrieves a list of references for a UserGroup.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReferencesOfUserGroup"
    "uid": <UserGroup Uid>
    "mandator": <Mandator Uid>
}
```

- "uid" (mandatory): UserGroup uid
- "mandator" (mandatory): Mandator uid

Returns

A HTTP response with the following content:

```
{
    "ownReferences": [<reference>],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

248 GetReminders

Detailed Description

GetReminders retrieves reminders for conditions/conclusions and rule actions. If the type is "caseSelection", no mandator or revision parameter is needed; all reminders are returned that belong to a mandator of a case queue selected in case selection. Otherwise, the reminders will be returned based on the revision parameter (returns the reminders of the revision mandator) or mandator parameter (returns the reminders of the mandator). If the type is "query" it will also return the reminders of the members.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getReminders",
  "type": "query" | "ruleAction" | "caseSelection",
  "mandator": <mandatorUid>,
  "revision": <revisionUid>
}
```

- "type" (optional): The request type to identify the origin of the request.
- "mandator" (optional): The Mandator (identified by his uid) of the element the condition/conclusion/rule action belongs to.
- "revision" (optional): The Revision (identified by its uid) where the element the condition/conclusion/rule action belongs to is defined.

Returns

A HTTP response with the following content:

```
{
  "reminders": [{
    "uid": <reminderUid>,
    "id": <reminderId>,
    "name": "<reminderName>"
  },
  ...
],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

249 GetReminderTable

Detailed Description

GetReminderTable returns a table of all reminders the user has view privilege for.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReminderTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

250 GetReportingAttributes

Detailed Description

GetReportingAttributes retrieves all reporting attributes for a case.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getReportingAttributes",
    "uid": <caseUid>
}
```

- "uid" (mandatory): The uid of a case for which reporting attributes has to be retrieved.

Returns

A HTTP response with the following content:

```
{
    "reportingAttributes": [{
        "uid": <attributeUid>,
        "name": "<attributeName>",
        "mandator": <mandatorUid>,
        "revision": <revisionUid>,
        "mdcStored": true|false,
        "origin": "<attributeOrigin>",
        "type": "<type>",
        "encrypted": true|false,
        "length": <length>,
        "decimals": <decimals>,
        "dimension": "<dimension>",
        "categoriesEnabled": true|false,
        "categories": [ ... ],
        "usedInMergingConclusion": true|false,
        "metaAttribute": "<metaAttribute>" | "None",
        "format": "<format>" | "None"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

251 GetRevisionAuditTrail

Detailed Description

GetRevisionAuditTrail is used to request all audit trail entries for a particular revision within IBM Safer Payments that match the provided selection criteria or to retrieve the audit trail of a specific model element

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRevisionAuditTrail",
  "uid": <elementUid>,
  "type": "element" | "filter",
  "revision": <revisionUid>,
  "data": {
    "value": "<value>",
    "comparisonResult": "Added" | "all" | "ChangedComputationImpact" |
"ChangedNoImpact" | "Removed",
    "entryType": "Attribute" | "Collusion" | "Counter" | "DeviceIdentification" |
"Event" | "Formula" | "Index" | "List" | "Mapping" | "Masterdata" | "Merging" | "Notification" |
"NumericCategory" | "Pattern" | "Precedent" | "Profile" | "Rule" | "Ruleset" | "PmmlModel" |
"Stencil" | "TextCategory" | "UnknownType" | "Value" | "all",
    "changeRestriction": {
      "from": <fromTimestamp>,
      "to": <toTimestamp>
    }
    "users": [<userId>, ...]
  }
}
```

- "uid": The uid of the element the audit trail belongs to
- "type": "element" to request an audit trail for a single element specified by uid or "filter" for the general revision audit trail
- "revision": The uid of the revision of the audit trail
- "data:value" (optional): A search value to filter the general revision audit trail when using type "filter"
- "data:comparisonResult" (optional): which type of change should be retrieved when using type "filter"
- "data:entryType" (optional): which entry types should be retrieved when using type "filter"
- "data:changeRestriction:from" (optional): A start timestamp to restrict the audit trail entries by time when using type "filter"
- "data:changeRestriction:to" (optional): An end timestamp to restrict the audit trail entries by time when using type "filter"
- "data:users" (optional): A list of users to filter the audit trail when using type "filter"

Returns

A HTTP response with the following content:

```
{
  "revisionNumber": "<revisionNumber>",
  "editingUser": <userId>,
  "auditTrail": <auditTrail>
  "responseStatus": [ "<status>", "<feedbacktext>" ],
  "reloadUserProfile": true|false
}
```


252 GetRevisionGeneral

Detailed Description

GetRevisionGeneral returns generation information about a model revision.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRevisionGeneral",
  "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which information should be retrieved.

Returns

A HTTP response with the following content:

```
{
  "revisionGeneral": {
    "name": "<revisionName>",
    "comment": "<revisionComment>",
    "status": "<status>",
    "editedBy": <editedByUserId>|-1,
    "editedByName": "<editedByUserName>" | "",
    "editedSince": <editedSinceTimestamp>,
    "lastChangedBy": <lastChangedByUserId>|-1,
    "lastChangedTimestamp": <lastChangedTimestamp>,
    "setLifeBy": "<setLifeByUserName>" | "",
    "setLifeTimestamp": <setLifeTimestamp>,
    "retiredBy": "<retiredByUserName>" | "",
    "retiredTimestamp": <retiredTimestamp>
  },
  "ddcStatistics": {
    "inputsMemory": "<inputsMemory>",
    "numberOfInputs": "<numberOfInputs>",
    "outputsMemory": "<outputsMemory>",
    "numberOfOutputs": "<numberOfOutputs>",
    "listsMemory": "<listsMemory>",
    "numberOfLists": "<numberOfLists>",
    "indexesMemory": "<indexesMemory>",
    "numberOfIndexes": "<numberOfIndexes>",
    "masterdataMemory": "<masterdataMemory>",
    "numberOfMasterdatas": "<numberOfMasterdatas>",
    "deviceIdentificationsMemory": "<deviceIdentificationsMemory>",
    "numberOfDeviceIdentifications": "<numberOfDeviceIdentifications>",
  }
}
```

```

    "precedentsMemory": "<precedentsMemory>",
    "numberOfPrecedents": "<numberOfPrecedents>",
    "profilesMemory": "<profilesMemory>",
    "numberOfProfiles": "<numberOfProfiles>",
    "patternsMemory": "<patternsMemory>",
    "numberOfEvents": "<numberOfPatterns>",
    "eventsMemory": "<eventsMemory>",
    "numberOfEvents": "<numberOfEvents>",
    "countersMemory": "<countersMemory>",
    "numberOfCounters": "<numberOfCounters>",
    "collusionsMemory": "<collusionsMemory>",
    "numberOfCollusions": "<numberOfCollusions>",
    "formulasMemory": "<formulasMemory>",
    "numberOfFormulas": "<numberOfFormulas>",
    "totalMemory": "<totalMemory>"
  },
  "mdcStatistics":{
    "inputsMemory": "<inputsMemory>",
    "numberOfInputs": "<numberOfInputs>",
    "outputsMemory": "<outputsMemory>",
    "numberOfOutputs": "<numberOfOutputs>",
    "listsMemory": "<listsMemory>",
    "numberOfLists": "<numberOfLists>",
    "indexesMemory": "<indexesMemory>",
    "numberOfIndexes": "<numberOfIndexes>",
    "masterdatasMemory": "<masterdataMemory>",
    "numberOfMasterdatas": "<numberOfMasterdatas>",
    "deviceIdentificationsMemory": "<deviceIdentificationsMemory>",
    "numberOfDeviceIdentifications": "<numberOfDeviceIdentifications>",
    "precedentsMemory": "<precedentsMemory>",
    "numberOfPrecedents": "<numberOfPrecedents>",
    "profilesMemory": "<profilesMemory>",
    "numberOfProfiles": "<numberOfProfiles>",
    "patternsMemory": "<patternsMemory>",
    "numberOfEvents": "<numberOfPatterns>",
    "eventsMemory": "<eventsMemory>",
    "numberOfEvents": "<numberOfEvents>",
    "countersMemory": "<countersMemory>",
    "numberOfCounters": "<numberOfCounters>",
    "collusionsMemory": "<collusionsMemory>",
    "numberOfCollusions": "<numberOfCollusions>",
    "formulasMemory": "<formulasMemory>",
    "numberOfFormulas": "<numberOfFormulas>",
    "totalMemory": "<totalMemory>"
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}

```

253 GetRevisions

Detailed Description

GetRevisions returns all revisions for a given mandator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRevisions",
    "mandator": <mandatorUid>
}
```

- "mandator" (mandatory): The mandator for which revisions should be fetched.

Returns

A HTTP response with the following content:

```
{
    "revisions": [{
        "name": "[<revisionId>] \'<revisionName>\'",
        "uid": <revisionUid>,
        "status": <status>
    },
    ...
    ],
    "responseStatus": [<status>, "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

Revision status value mapping:

```
0 means revision status of "Challenger"
1 means revision status of "Golive"
2 means revision status of "Champion"
3 means revision status of "Retired"
4 means revision status of "Invalidated"
5 means revision status of "Golive waiting for confirmation"
```

254 GetRevisionSelectionTable

Detailed Description

GetRevisionSelectionTable returns a table of all revisions for a given mandator. The results are filtered according to the user preference.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRevisionSelectionTable",
    "mandator": <mandatorUid>
}
```

- "mandator" (mandatory): The mandator for which revisions should be fetched.

Returns

A HTTP response with the following content:

```
{
    "mandatorName": "<mandatorName>",
    "tableContent": [[
        <revisionUid>,
        <editingUserUid>,
        <revisionNumber>,
        "<revisionName>",
        "<revisionComment>",
        "<revisionStatus>",
        "<editingUserName>",
        <editingSinceTimestamp>,
        "<lastChangedBy>",
        <lastChangedTimestamp>,
        "<setLifeBy>",
        <setLifeTimestamp>,
        "<retiredBy>",
        <retiredTimestamp>,
        <ifNotContainSubMandator>>true|false
    ],
    ...
    ],
    "refresh": <autoRefreshSettingInSecondsForModelGolive>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

If the revision is not edited by any user, the "editingUserUid" is set to -1. Available values for "revisionStatus" are: "Challenger", "Golive", "Champion", "Retired", "Invalidated", "Waiting".

255 GetRevisionStatus

Detailed Description

GetRevisionStatus returns the status of a given revision.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRevisionStatus",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which the status should be checked.

Returns

A HTTP response with the following content:

```
{
    "revisionStatus": {
        "reserved": true|false,
        "isChallenger": true|false,
        "challengerStatus": "Simulate" | "Analyze" | "Ready" | "GenerateRules",
        "status": "Challenger" | "Golive" | "Champion" | "Retired" | "Invalidated"
    },
    "Waiting",
    "name": "<revisionName>",
    "revision": <revisionNumber>,
    "mandatorName": "<mandatorName>",
    "mayDefineFinalRulesets": true|false,
    "autoRefreshAnalysis": <autoRefreshSetting>,
    "simulationValid": true|false,
    "averageSimulationPerformance": <averageSimulationPerformance>
},
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

256 GetRoles

Detailed Description

GetRoles returns roles for a certain mandator. A role stores different privileges a user is granted / not granted for safer payments.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRoles",
    "mandator": <mandatorUid>
}
```

- "mandator": The uid of mandator, for which the roles are retrieved

Returns

A HTTP response with the following content:

```
{
    "roles": [<role>, ...],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

257 GetRolesTable

Detailed Description

GetRolesTable returns a table of all roles that the user can view.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRolesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <roleUId>,
        <mandatorUId>,
        "<mandatorName>",
        "<roleName>",
        "<roleComment>",
        <numberOfPrivileges>,
        <numberOfUsers>,
        "<lastChangedByUserName>",
        <lastChangedOnTimestamp>
    ],
    ... ..
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

258 GetRuleGenerationAPrioriConditions

Detailed Description

GetRuleGenerationAPrioriConditions returns rule generation predefined conditions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRuleGenerationAPrioriConditions",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which predefined conditions should be fetched.

Returns

A HTTP response with the following content:

```
{
    "aPrioriConditions": [ ... ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


259 GetRuleGenerationDataSelection

Detailed Description

GetRuleGenerationDataSelection returns rule generation data selection for a given revision.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRuleGenerationDataSelection",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which rule generation data selection should be fetched.

Returns

A HTTP response with the following content:

```
{
    "simulationAttributesAvailable": true|false,
    "missingAttributes": [ ... ... ],
    "performanceTableProgress": "<performanceTableProgress>",
    "verify": true|false,
    "generationScenario": "FromScratch" | "CurrentRevisionOnly" |
    "CurrentRevisionAndHistory",
    "trainingDataSelectionStream": {
        "mandators": [ ... ... ],
        "periodType": "RecordsAbsolute" | "RecordsRelative" | "TimeAbsolute" |
    "TimeRelative",
        "actual": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
            "toTimestamp": <toTimestamp>,
            "toDays": <toDays>
        },
        "desired": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
        }
    }
}
```

```

        "toTimestamp": <toTimestamp>,
        "toDays": <toDays>
    },
    "incDdc": true|false,
    "conditions": [ ... ... ]
},
"verificationDataSelectionStream": {
    "mandators": [<mandatorUid>],
    "periodType": "RecordsAbsolute" | "RecordsRelative" | "TimeAbsolute" |
    "TimeRelative",
    "actual": {
        "fromUrid": <fromUrid>,
        "fromRecord": <fromRecord>,
        "fromTimestamp": <fromTimestamp>,
        "fromDays": <fromDays>,
        "toUrid": <toUrid>,
        "toRecord": <toRecord>,
        "toTimestamp": <toTimestamp>,
        "toDays": <toDays>
    },
    "desired": {
        "fromUrid": <fromUrid>,
        "fromRecord": <fromRecord>,
        "fromTimestamp": <fromTimestamp>,
        "fromDays": <fromDays>,
        "toUrid": <toUrid>,
        "toRecord": <toRecord>,
        "toTimestamp": <toTimestamp>,
        "toDays": <toDays>
    },
    "incDdc": true|false,
    "conditions": [ ... ... ]
},
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}

```

260 GetRuleGenerationDesignTable

Detailed Description

GetRuleGenerationDesignTable returns rule generation design table for a given challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRuleGenerationDesignTable",
  "revision": <revisionUid>,
  "data": {
    "showNonSelectedIndicatorAttributes": true|false
  }
}
```

- "revision" (mandatory): The revision for which rule generation design table should be returned.
- "data" (optional):
- "data:showNonSelectedIndicatorAttributes" (optional): A boolean determining if non-selected indicator attributes should be retrieved. Default value is false.

Returns

A HTTP response with the following content:

```
{
  "stopCondition": "<stopCondition>",
  "noConditionProposedReason": "<noConditionProposedReason>",
  "decimalsAmount": <decimalsAmount>,
  "ignoredConditionContainingEncryptedAttribute": true|false,
  "numberOfGeneratedConditions": <numberOfGeneratedConditions>,
  "numberOfUsedAttributes": <numberOfUsedAttributes>,
  "numberOfGeneratedRules": <numberOfGeneratedRules>,
  "tableContent": [[
    ... ..
  ],
  ... ..
],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

261 GetRuleGenerationExplore

Detailed Description

GetRuleGenerationExplore returns, for a given attribute, all indicators to be shown in the rule designer table.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRuleGenerationExplore",
  "uid": <attributeUid>,
  "type": "all" | "",
  "revision": <revisionUid>
}
```

- "uid" (mandatory): The uid of an attribute for which indicators should be returned.
- "type" (mandatory): Filter type: if "all", all indicators are returned. If not set, only indicators for which training results contain fraudulent records are returned.
- "revision" (mandatory): The revision of an attribute.

Returns

A HTTP response with the following content:

```
{
  "exploreAttribute": {
    "name": "<attributeName>",
    "modelUsage": "NoUsage" | "Categoric" | "Quantise" | "Logarithmic" |
    "Interval" | "Integer" | "CustomInterval" | "CustomCategoric",
    "type": "attributeType",
    "dimension": "attributeDimension",
    "decimals": <decimals>
  },
  "amountAttribute": {
    "dimension": "<dimension>",
    "decimals": <decimals>
  },
  "tableContent": [
    [ ... ... ],
    ... ...
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

262 GetRuleGenerationPerformance

Detailed Description

GetRuleGenerationPerformance returns rule generation performance for a given challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRuleGenerationPerformance",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which the generation performance should be returned.

Returns

A HTTP response with the following content:

```
{
    "generationScenario": "FromScratch" | "CurrentRevisionOnly" |
    "CurrentRevisionAndHistory",
    "training": {
        "actual": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
            "toTimestamp": <toTimestamp>,
            "toDays": <toDays>
        },
        "desired": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
            "toTimestamp": <toTimestamp>,
            "toDays": <toDays>
        },
        "minimumUrid": <minimumUrid>,
        "maximumUrid": <maximumUrid>,
        "minimumSystemtime": <minimumSystemTimestamp>,
        "maximumSystemtime": <maximumSystemTimestamp>
    }
}
```

```

    },
    "verification": {
        "actual": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
            "toTimestamp": <toTimestamp>,
            "toDays": <toDays>
        },
        "desired": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
            "toTimestamp": <toTimestamp>,
            "toDays": <toDays>
        },
        "minimumUrid": <minimumUrid>,
        "maximumUrid": <maximumUrid>,
        "minimumSystemtime": <minimumSystemTimestamp>,
        "maximumSystemtime": <maximumSystemTimestamp>
    },
    "numberOfGeneratedRules": <numberOfGeneratedRules>,
    "numberOfGeneratedConditions": <numberOfGeneratedConditions>,
    "tableContent": [
        [ ... ... ],
        ... ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}

```

263 GetRuleGenerationRules

Detailed Description

GetRuleGenerationRules returns a table of generated rules for a given challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRuleGenerationRules",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which a table of generated rules should be retrieved.

Returns

A HTTP response with the following content:

```
{
    "amountDimension": "<amountDimension>",
    "verify": "true|false",
    "tableContent": [
        [ ... ... ],
        .. ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

264 GetRuleGenerationSettings

Detailed Description

GetRuleGenerationSettings returns rule generation settings for a given challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRuleGenerationSettings",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which rule generation settings should be returned.

Returns

A HTTP response with the following content:

```
{
    "performanceTableProgress": "<performanceTableProgress>",
    "balanceFraudFalsePositives": <balanceFraudFalsePositives>,
    "lowerBoundFalsePositives": <lowerBoundFalsePositives>,
    "upperBoundFraudHitRate": <upperBoundFraudHitRate>,
    "stopRuleFalsePositives": <stopRuleFalsePositives>,
    "maxNumberConditions": <maxNumberConditions>,
    "minPopulationQuantise": <minPopulationQuantise>,
    "minSampleSizePercent": <minSampleSizePercent>,
    "minSelectHitRate": <minSelectHitRate>,
    "minConditionImprovement": <minConditionImprovement>,
    "relaxAllUpAutoThreshold": <relaxAllUpAutoThreshold>,
    "maxNumberRules": <maxNumberRules>,
    "maxFraudHit": <maxFraudHit>,
    "rulesName": "<rulesName>",
    "rulesComment": "<rulesComment>",
    "aPrioriConditions": [ <conditions> ],
    "ruleConclusion": {
        "interceptValue": <interceptValue>
    },
    "responseStatus": [ "<status>", "<feedbacktext>" ],
    "reloadUserProfile": true|false
}
```


265 GetRuleGenerationStatus

Detailed Description

GetRuleGenerationStatus returns the status of rule generation for a given challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRuleGenerationStatus",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which the status should be retrieved.

Returns

A HTTP response with the following content:

```
{
    "simulationAttributesAvailable": true|false,
    "missingAttributes": [
        ... ..
    ],
    "verify": true|false,
    "performanceTableProgress": true|false,
    "designerTableProgress": true|false,
    "automaticRulegenerationRunning": true|false,
    "refresh": <autoRefreshSettingInSecondsForModelElementGeneration>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

266 GetRuleGenerationTargets

Detailed Description

GetRuleGenerationTargets returns all target rulesets of generated rules for a given model revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRuleGenerationTargets",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision for which target rulesets should be returned.

Returns

A HTTP response with the following content:

```
{
    "rulesets": [{
        "uid": <rulesetUid>,
        "name": "<rulesetName>",
        "comment": "<comment>",
        "revision": <revisionUid>
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

267 GetRulePerformance

Detailed Description

GetRulePerformance retrieves the results of rule performance based on valid analysis results.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRulePerformance",
  "uid": <ruleUid>,
  "revision": <revisionUid>,
  "data": {
    "analysisId": <analysisId>
  }
}
```

- "uid" (mandatory): The Rule (identified by its uid) to request the results of rule performance for.
- "revision" (mandatory): The Revision (identified by its uid) where the analysis has been executed.
- "data:analysisId" (mandatory): The Analysis (identified by its id) that delivers the results for rule performance.

Returns

A HTTP response with the following content:

```
{
  "metaAmountDimension": "<dimension>",
  "metaAmountDecimals": <decimal>,
  "conditionsShort": "<conditions>",
  "conclusionsShort": "<conclusions>",
  "graphDataDaily": {
    "totalFraud": [<value>],
    "numberFraud": [<value>],
    "amountFraud": [<value>],
    "numberGenuine": [<value>],
    "amountGenuine": [<value>],
    "timestamp": [<timestamp>]
  },
  "graphDataWeekly": {
    "totalFraud": [<value>],
    "numberFraud": [<value>],
    "amountFraud": [<value>],
    "numberGenuine": [<value>],
    "amountGenuine": [<value>],
    "timestamp": [timestamp]
  }
}
```

```
    },
    "data": {
      "simulationValid": true|false,
      "simulationEnabled": true|false,
      "isDataInIris": true|false,
      "simulationStatus": "<status>",
      "simulationProgress": <progress>,
      "progress": <progress>,
      "recordsProcessed": <numberOfRecordsProcessed>,
      "optimizationProgress": <progress>,
      "manRecordsProcessed": <manRecordsProcessed>,
      "finished": true|false,
      "optimizing": true|false,
      "uridFrom": <fromUrid>,
      "uridTo": <toUrid>,
      "refresh": <autoRefreshSetting>,
      "analysisResultsValid": true|false,
      "analysisStatus": "<status>"
    },
    "name": "<ruleName>",
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
  }
}
```

268 GetRulePerformanceRulesets

Detailed Description

GetRulePerformanceRulesets retrieves all rules that are enabled for the specified analysis.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRulePerformanceRulesets",
  "revision": <revisionUid>,
  "data": {
    "analysisId": <analysisId>,
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the analysis has been executed.
- "data:analysisId" (mandatory): The Analysis (identified by its uid) for which the enabled rulesets are requested.

Returns

A HTTP response with the following content:

```
{
  "rulePerformanceRulesets": [{
    "name": "<rulesetName>",
    "uid": <rulesetUid>,
    "rules": [{
      "uid": <ruleUid>,
      "name": "<ruleName>"
    },
    ... ..
  ]},
  ... ..
],
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

269 GetRuleReport

Detailed Description

GetRuleReport executes a rule report and retrieves the results.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRuleReport",
  "uid": <ruleUid>,
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "withoutContext": true|false
  }
}
```

- "uid" (mandatory): The Rule (identified by its uid) for which the rule report result is requested.
- "mandator" (mandatory): The Mandator (identified by his uid) where the rule report is executed.
- "revision" (mandatory): The Revision (identified by its uid) where the rule report is executed.
- "data:withOutContext" (mandatory): A flag whether or not the rule report uses the current simulation or starts a new simulation for the relevant elements.

Returns

A HTTP response with the following content:

```
{
  "ruleName": "<ruleName>",
  "ruleComment": "<ruleComment>",
  "rulePriority": <rulePriority>,
  "ruleEnabled": <enabled>1|0,
  "rulesetName": "<rulesetName>",
  "rulesetComment": "<rulesetComment>",
  "rulesetPriority": <rulesetPriority>,
  "rulesetEnabled": <enabled>1|0,
  "rulesetConditions": [
    <rulesetConditions>
  ],
  "mandator": "<mandatorName>",
  "revisionName": "[<revisionId> \'<revisionName>\'",
  "ruleConditions": [
    <ruleConditions>
  ]
}
```

```
],
"ruleActions": [
    <ruleActions>
],
"userName": "<userName>",
"generationTimestamp": "<timestamp>",
"amountDimension": "<amountDimension>",
"amountDecimals": <decimals>,
"interceptAttributeName": "<interceptAttributeName>",
"simulationMethod": "<simulationMethod>",
"simulationPeriodType": "<simulationPeriodType>",
"simulationNumberOfTrx": <simulationNumberOfTransactions>,
"simulationTotalAmount": <simulationTotalAmount>,
"simulationDatarangeFrom": <simulationDataRangeFrom>,
"simulationDatarangeTo": <simulationDataRangeTo>,
"simulationConditions ": [
    <simulationConditions>
],
"summary":{
    <summary>
},
"responseStatus": ["<status>", "<feedbacktext>"],
"reloadUserProfile": true|false
}
```

270 GetRulesStatistics

Detailed Description

GetRulesStatistics retrieves the results of rule analysis. The results are retrieved based on a valid Analysis.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRulesStatistics",
  "revision": <revisionUid>,
  "data": {
    "rulesets": [<rulesetUid>],
    "analysisId": <analysisId>
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the analysis has been executed.
- "data:rulesets" (mandatory): The Rulesets (identified by their uids) for which the result of rule analysis is requested.
- "data:analysisId" (mandatory): The Analysis (identified by its id) for which the results of rule analysis are requested.

Returns

A HTTP response with the following content:

```
{
  "ruleStatistics": [[
    <ruleUid>,
    <rulesetUid>,
    <revisionUid>,
    <hitRate>,
    <falseAlarmRatio>,
    <savedAmountPerFalseAlarm>,
    <fraudAmountIntercepted>,
    <fraudTransactionsIntercepted>,
    <genuineAmountIntercepted>,
    <genuineTransactionsIntercepted>,
    "<rulesetName>",
    "<ruleName>"
  ]],
  ... ..
],
  "refresh": <autoRefreshSetting>,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


271 GetRulesTable

Detailed Description

GetRulesTable retrieves the rules table of a ruleset. It includes all rules of the specified ruleset. Ruleset may be inherited.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRulesTable",
    "revision": <revisionUid>,
    "ruleset": <rulesetUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the rules table is requested.
- "ruleset" (mandatory): The Ruleset (identified by its uid) whose rules are requested.

Returns

A HTTP response with the following content:

```
{
    "ruleSetName": "<rulesetName>",
    "ruleSetConditions": [
        { <condition> },
        ... ..
    ],
    "enabled": 0|1,
    "tableContent": [
        [
            <ruleUid>,
            <revisionUid>,
            <priority>,
            "<name>",
            "<comment>",
            "Yes" | "No",
            "<conditionsShortString>",
            "<conclusionsShortString>"
        ],
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

272 GetRuleStatistics

Detailed Description

GetRuleStatistics retrieves the result of a single rule analysis. The results are retrieved based on a valid single rule Analysis.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getRuleStatistics",
  "revision": <revisionUid>,
  "data": {
    "rule": "ruleUid",
    "analysisId": <analysisId>
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the single rule analysis has been executed.
- "data:rule" (mandatory): The Rule (identified by its uid) for which the result of single rule analysis is requested.
- "data:analysisId": The Analysis (identified by its id) for which the result of single rule analysis is requested.

Returns

A HTTP response with the following content:

```
{
  "ruleStatistics": {
    "dimension": "<dimension>",
    "fraudHit": <fraudHitRate>,
    "falseAlarms": <falseAlarms>,
    "SAPFA": <savedAmountPerFalseAlarm>,
    "fraudAmountIntercepted": <fraudAmountIntercepted>,
    "fraudTrxIntercepted": <fraudTransactionsIntercepted>,
    "genuineAmountIntercepted": <genuineAmountIntercepted>,
    "genuineTrxIntercepted": <genuineTransactionsIntercepted>
  },
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

273 GetRunningSimulations

Detailed Description

GetRunningSimulations retrieves all registered simulations of a user. It does not return the simulation of the specified Revision itself but all simulations of the same Mandator in other Revisions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getRunningSimulations",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) for which the request is performed.

Returns

A HTTP response with the following content:

```
{
    "runningSimulations": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

274 GetSandboxTable

Detailed Description

GetSandboxTable retrieves the Sandbox table of a Revision.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getSandboxTable",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the sandbox table is requested.

Returns

A HTTP response with the following content:

```
{
    "header": [{
        "key": "attributeUid"
    }, {
        "key": "Attributes"
    }],
    ... ..
],
    "tableContent": [[
        ... ..
    ]],
    ... ..
],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

275 GetSearchComplianceListEntriesTable

Detailed Description

GetSearchComplianceListEntriesTable retrieves a table of search results for compliance list entries. The requesting User must have the privilege to view compliance lists.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getSearchComplianceListEntriesTable",
  "data": {
    "selectedTypes": <selectedTypes>,
    "entityType": "Individual"|"Legal",
    "listId": <listId>,
    "name": "<name>",
    "street": "<street>",
    "city": "<city>",
    "country": "<country>",
    "passport": "<passport>",
    "birthdate": "<birthdate>"
  }
}
```

- "data" (mandatory): Contains a compliance list type filter and the values to search for.
- "data:selectedTypes" (mandatory): Filters the type of compliance lists to search through. One of ["OfacSdn", "UnitedNationsList", "PoliticalExposedPersonList", "EuropeanTerrorList", "RussianTerrorList"]
- "data:entityType" (mandatory): Determines if individual or legal entries should be searched.
- "data:listId": The list id to search for.
- "data:name": The name to search for.
- "data:street": The street to search for.
- "data:city": The city to search for.
- "data:passport": The passport number to search for.
- "data:birthdate": The date of birth to search for.

Returns

A HTTP response with the following content:

```
{
  "tableContent": [
    ... ..
  ],
  "resultId": <resultId>,
  "numberOfEntriesShown": <numberOfEntriesInTable>,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

276 GetSessionCountdownData

Detailed Description

GetSessionCountdownData retrieves session countdown settings. The session countdown settings define how long a user session takes, before the user is automatically logged out.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{  
    "request": "getSessionCountdownData",  
}
```

Returns

A HTTP response with the following content:

```
{  
    "sessionCountdownThreshold": <sessionCountdownThreshold>,  
    "sessionCountdownRefresh": <sessionCountdownRefresh>,  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

277 GetSettings

Detailed Description

GetSettings retrieves all settings of system configuration.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getSettings"
}
```

Returns

A HTTP response with the following content:

```
{
    "data": {
        "streamType": "settings",
        "settings": {
            "application": {
                "name": "<applicationName>",
                "serverTimeZone": <serverTimezoneSetting>
            },
            "memory": {
                "limit": <mainMemoryUsageLimit>
            },
            ... ..
        },
        ... ..
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


278 GetSimulationDataSelection

Detailed Description

GetSimulationDataSelection retrieves the Simulation DataSelection of the specified Revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getSimulationDataSelection",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) of the simulation.

Returns

A HTTP response with the following content:

```
{
    "simulationDataSelection": {
        "mandators": [<mandatorUid>, ...],
        "periodType": "RecordsAbsolute" | "RecordsRelative" | "TimeAbsolute" |
"TimeRelative",
        "actual": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
            "toTimestamp": <toTimestamp>,
            "toDays": <toDays>
        },
        "desired": {
            "fromUrid": <fromUrid>,
            "fromRecord": <fromRecord>,
            "fromTimestamp": <fromTimestamp>,
            "fromDays": <fromDays>,
            "toUrid": <toUrid>,
            "toRecord": <toRecord>,
            "toTimestamp": <toTimestamp>,
            "toDays": <toDays>
        },
        "incDdc": true|false,
        "conditions": [ ... ]
    },
    "simulationMethod": "PerRecord" | "PerChunk",
    "metaSystemTime": true|false,
    "responseStatus": [<status>, "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

279 GetSimulationProgress

Detailed Description

GetSimulationProgress returns the current status of a running simulation for a given challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getSimulationProgress",
  "revision": <revisionUid>
}
```

- "revision": The revision for which simulation status should be returned.

Returns

A HTTP response with the following content:

```
{
  "data": {
    "simulationValid": true|false,
    "simulationEnabled": true|false,
    "isDataInIris": true|false,
    "simulationStatus": "SimDisabled" | "SimInitializing" | "SimComputing" |
    "SimReady" | "SimStopping",
    "simulationProgress": <simulationProgress>,
    "refresh": <autoRefreshSetting>,
    "singleRuleSimulation": true|false,
    "simulationInitiatedAt": <simulationInitiatedAtTimestamp>,
    "simulationProcessingStartedAt": <simulationProcessingStartedAtTimestamp>,
    "simulationProcessingFinishedAt": <simulationProcessingFinishedAtTimestamp>,
    "simulationProcessedRecords": <numberOfRecordsProcessedBySimulation>,
    "transactionsPerSecondAverage": <averageTransactionsPerSecond>,
    "averageProcessingTime": <averageProcessingTime>,
    "estimatedFinishingTimestamp": <estimateFinishingTimestamp>
  },
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

280 GetSimulationTable

Detailed Description

GetSimulationTable retrieves the table of all simulation queries of a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getSimulationTable",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) for which the simulation query table is requested.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <simulationQueryUid>,
        <mandatorUid>,
        "<simulationQueryType>",
        <indexUid>|-1,
        "<indexName" | "",
        "<mandatorName>",
        "<simulationQueryName>",
        "<comment>",
        "Yes" | "No",
        <numberOfColumns>,
        "<lastChangedByUserName>",
        <lastChangedOnTimestamp>
    ],
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

281 GetStatistics

Detailed Description

GetStatistics retrieves the statistics of the specified Analysis.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getStatistics",
  "uid": <revisionUid>,
  "data": {
    "analysisId": <analysisId>
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the Analysis was executed.
- "data:analysisId" (mandatory): The Analysis (identified by its id) for which the statistics are requested.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

282 GetStatusAlarmIndicatorsDisplay

Detailed Description

GetStatusAlarmIndicatorsDisplay returns data, to display status alarm indicators on the dashboard. Only the status alarm indicators that the user has view privilege are returned.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getStatusAlarmIndicatorsDisplay"
}
```

Returns

A HTTP response with the following content:

```
{
    "statusAlarmIndicators": [{
        "streamType": "statusAlarmIndicator",
        "uid": <statusAlarmIndicatorUid>,
        "showOnDashboard": true|false,
        "position": <position>,
        "instance": <instanceId>,
        "type": <alarmType>,
        "status": true|false,
        "replacements": {
            "{alarmTimestamp}": "",
            "{comment}": "",
            "{externalMessage}": "",
            "{instanceId}": "<instanceId>",
            "{instanceName}": "<instanceName>",
            "{integerValue}": "<integerValue>",
            "{lastUpdateTimestamp}": "<lastUpdateTimestamp>",
            "{mandator}": "<mandatorName>",
            "{name}": "<statusAlarmIndicatorName>",
            "{thresholdAbove}": "<thresholdAboveValue>",
            "{thresholdBelow}": "<thresholdBelow>",
            "{value}": "<value>"
        },
        "displayText": "<displayText>",
        "displayTooltip": "<displayTooltip>"
    },
    ... ..
  ],
  "saiConfig": {
    "explanation": "<explanation>",
    "tooltip": "<toolTip>",
  }
}
```

```
    "onlineHelpType": "<onlineHelpType>",
    "onlineHelp": "<onlineHelp>"
  },
  "refresh": <autoRefreshSettingInSecondsForDashboard>",
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

283 GetStatusAlarmIndicatorsTable

Detailed Description

GetStatusAlarmIndicatorsTable returns a table of all SAIs of all mandators the user has StatusAlarmIndicator view privilege for.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getStatusAlarmIndicatorsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <statusAlarmIndicatorUid>,
        <enabled>"Yes" | "No",
        <mandatorUid>,
        "<statusAlarmIndicatorName>",
        "<statusAlarmIndicatorComment>",
        "<alarmType>",
        "<alarmStatus>",
        <position>,
        "<lastChangedByUserName>",
        <lastchangedOnTimestamp>
    ],
    ... ..
    ],
    "nextPosition": <nextPosition>,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

284 GetSystemInformation

Detailed Description

GetSystemInformation Retrieves all data displayed on the "System internals" user interface page. The requesting User must have the privilege to view system internals.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getSystemInformation"
}
```

Returns

A HTTP response with the following content:

```
{
    "systemInternals": [
        ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


285 GetSystemLogTable

Detailed Description

GetSystemLogTable retrieves the system log table. The requesting User must have the privilege to view the system log.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getSystemLogTable",
  "data": {
    "from": <fromTimestamp>,
    "to": <toTimestamp>,
    "instances": [<instanceUid>],
    "users": [<userId>]
  }
}
```

- "data" (optional): Contains values to filter the retrieved log messages.
- "data:from" (optional): If set, only log messages that occurred after this timestamp are retrieved.
- "data:to" (optional): If set, only log messages that occurred before this timestamp are retrieved.
- "data:instances" (optional): A list of IrisInstance objects (identified by their uids) to retrieve log messages for.
- "data:users" (optional): A list of User objects (identified by their uids) to retrieve log messages for.

Returns

A HTTP response with the following content:

```
{
  "recordLimit": <maximumNumberOfRecords>,
  "tableContent": [
    ... ..
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

286 GetTargetRevisions

Detailed Description

GetTargetRevisions retrieves possible revisions to copy model elements to. Possible revisions are challenger revisions of a mandator the user has model edit privilege for and that the user is currently editing. Also, on the revision's mandator, the user needs edit privilege for the element he wants to copy.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getTargetRevisions",
    "uid": "<revisionElementUid>",
    "type": <revisionElementType>["modelling" | "input" | "output" | "list" | "index"
| "merging" | "masterdata" | "deviceIdentification" | "precedent" | "profile" | "pattern"
| "event" | "counter" | "collusion" | "formula" | "ruleset" | "finalRuleset"],
    "revision": <sourceRevisionUid>
}
```

- "uid" (mandatory): The RevisionElement (identified by its uid) that is copied.
- "type" (mandatory): The type of the RevisionElement.
- "revision" (mandatory): The Revision (identified by its uid) where the element is copied from.

Returns

A HTTP response with the following content:

```
{
    "revisions": [[
        "<mandatorName>",
        <mandatorUid>,
        "<[revisionId] \<revisionName>\'",
        <revisionUid>
    ],
    ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

287 GetTextModules

Detailed Description

GetTextModules retrieves all enabled TextModules of the specified mandator and its head mandators.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getTextModules",
    "mandator": <mandatorUid>
}
```

- "mandator" (optional): The Mandator (identified by his uid) TextModules are requested for.

Returns

A HTTP response with the following content:

```
{
    "textModules": [{
        "uid": <textModuleUid>,
        "name": "<textModuleName>"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

288 GetTextModulesTable

Detailed Description

GetTextModulesTable retrieves all TextModules of all mandators the user has TextModule view privilege for.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getTextModulesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <textModuleUid>,
        <mandatorUid>,
        "<mandatorName>",
        <enabled>"Yes" | "No",
        "<textModuleName>",
        "<comment>",
        "<textTemplate>",
        "<lastChangedByUserName>",
        <lastChangedTimestamp>
    ],
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

289 GetThreadPriorities

Detailed Description

GetThreadPriorities returns available thread priority options.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getThreadPriorities"
}
```

Returns

A HTTP response with the following content:

```
{
    "data": [
        "<priority_1>",
        "<priority_2>",
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

290 GetTickerEntries

Detailed Description

GetTickerEntries lists all ticker entries. Entries are filtered by user privileges for different mandators.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getTickerEntries"
}
```

Returns

A HTTP response with the following content:

```
{
    "streamType": "tickerData",
    "maxEntries": <maxTickerEntriesSetting>,
    "tickerHighlightInterval": <tickerHighLightInterval>,
    "tickerRefreshInterval": <tickerRefreshInterval>,
    "lastLogOut": <lastLogOutTimestamp>,
    "lastLogIn": <lastLogInTimestamp>,
    "tickerEntriesList": [{
        "streamType": "tickerEntry",
        "uid": <entryId>,
        "name": "<userName>",
        "date": <timestamp>,
        "title": "<tickerTitle>",
        "note": "<tickerNote>",
        "mandatorUid": <mandatorUid>,
        "mandatorUids": [<mandatorUid>],
        "fromCppType": true|false,
        "cppId": <cppId>,
        "cppCaseGroupUid": <cppCaseGroupUid>,
        "cppMandatorUid": <cppMandatorUid>
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

291 GetTimestampAttributes

Detailed Description

GetTimestampAttributes retrieves a list of Attribute objects with data type Timestamp from a given Revision. Depending on the type used, the requesting User must have the privilege to either view group by queries, merchant monitoring rules or models.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getTimestampAttributes",
    "type": <type>,
    "mandator": <mandatorUid>,
    "revision": <revisionUid>,
    "data": {
        "includeDdc": true|false
    }
}
```

- "type" (optional): Describes the context in which the request was sent. One of ["groupByQuery", "deviceIdentification", "event", "merchantMonitoringTemplate", "profile", "sequence"].
- "mandator" (optional): The Mandator (identified by its uid) to retrieve attributes from. Only used for types "groupByQuery" and "merchantMonitoringTemplate".
- "revision" (optional): The Revision (identified by its uid) to retrieve attributes from. Only used when type is not "groupByQuery" or "merchantMonitoringTemplate".
- "data" (optional): Only used when type is "groupByQuery".
- "data:includeDdc" (optional): Normally only attributes stored in MDC are retrieved. This option allows to also retrieve attributes stored in DDC.

Returns

A HTTP response with the following content:

```
{
    "timestampAttributes": [
        ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

292 GetToAddressMeta

Detailed Description

GetToAddressMeta retrieves the value of meta Attribute "EMail" from a given investigation case.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "getToAddressMeta",
  "mandator": <mandatorUid>,
  "data": {
    "caseUid": <caseUid>
  }
}
```

- "mandator" (mandatory): The Mandator (identified by its uid) to retrieve the meta Attribute from.
- "data" (mandatory): See below.
- "data:caseUid" (mandatory): The Case identified by its uid to retrieve the meta Attribute value from.

Returns

A HTTP response with the following content:

```
{
  "toAddress": <toAddress>,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


293 GetTransactionMessageReportTable

Detailed Description

GetTransactionMessageReportTable is used to retrieve an overview on how many messages of different MTIDs were processed by IBM Safer Payments per time period. The table contains a column for each MTID that was processed, and a row for each time period. There are totals for each column and each row, including a total sum.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getTransactionMessageReportTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "header": [
        <MessageTypeId>,
        ... ..
    ],
    "tableContent": [
        [
            <reportingPeriod1>,
            <year>,
            <totalMessagesForMessageTypeId>,
            ... ..
            <totalMessagesInReportingPeriod1>
        ], [
            <reportingPeriod2>,
            <year>,
            <totalMessagesForMessageTypeId>,
            ... ..
            <totalMessagesInReportingPeriod2>
        ],
        ... ..
        [
            -1,
            -1,
            <totalMessagesForMessageTypeId>,
            ... ..
            <totalMessages>
        ]
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

294 GetUserGroups

Detailed Description

GetUserGroups lists all (case) user groups for a mandator. The contents of the response depend on the mandator that is passed with the request as well as the the privileges of the requesting user.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getUserGroups",
    "mandator": <mandatorUid>
}
```

- "mandator" (mandatory): The mandator for which user groups shall be queried.

Returns

A HTTP response with the following content:

```
{
    "userGroups": [{
        "uid": <userGroupUid>,
        "name": "<userGroupName>",
        "comment": "<comment>",
        "users": [<userId>, ...],
        "lastChangedBy": "<lastChangedByUserName>",
        "lastChangedTimestamp": <lastChangedTimestamp>
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

295 GetUserGroupsTable

Detailed Description

GetUserGroupsTable is used to retrieve a table of all defined user groups the requesting user can view.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getUserGroupsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <userGroupUid>,
        <mandatorUid>,
        "<mandatorName>",
        "<userGroupName>",
        "<comment>",
        <numberOfUsers>,
        "<lastChangedByUserName>",
        <lastChangedOnTimestamp>
    ],
    ... ..
  ],
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

296 GetUsers

Detailed Description

GetUsers retrieves a list of users who can access the specified type.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getUsers",
    "type": "<type>",
    "mandator": <mandatorUid>
}
```

- "type" (mandatory): The type of request. The available values are:

```
"missedCasesReport" (requires "mandator")
"memoryManagement" (requires "mandator")
"audit_log"
"system_log"
"caseFilter"
"risklistFilter"
"definedRiskList" (requires "mandator")
"userReport" (requires "mandator")
```

- "mandator": The mandator of an element. See remarks for the requests that require mandator parameter to be set.

Returns

A HTTP response with the following content:

```
{
    "users": [{
        "streamType": "user",
        "uid": <userId>,
        "enabled": true|false,
        "login": "<loginName>",
        "name": "<userName>",
    },
    ...
    ],
    "responseStatus": [<status>, "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

297 GetUsersAndMandatorsOfRoles

Detailed Description

GetUsersAndMandatorsOfRoles lists all users and mandators associate to the specified role(s). Each table entry in the response describes a role which is taken, the user which takes the role and the mandator for which it is assigned.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getUsersAndMandatorsOfRoles",
    "data": {
        "roles": [<roleUid>]
    }
}
```

- "data:roles" (optional): A list of role uids which shall be included.

Returns

A HTTP response with the following content:

```
{
    "roles": [{
        "role": "<roleName>",
        "user": "<userName>",
        "mandator": "<mandatorName>"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

298 GetUsersOfRole

Detailed Description

GetUsersOfRole retrieves all users that take a queried role.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getUsersOfRole",
    "uid": <roleUid>
}
```

- "uid" (mandatory): Uid of the role which is queried

Returns

A HTTP response with the following content:

```
{
    "users": [{
        "streamType": "user",
        "uid": <userId>,
        "enabled": true|false,
        "login": "<loginName>",
        "name": "<userName>"
    },
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

299 GetUsersReportsTable

Detailed Description

GetUsersReportsTable returns a list of all defined user reports (also known as investigator reports) the requesting user is privileged to see.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getUsersReportsTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <reportUid>,
        "<mandatorName>",
        "<reportName>",
        "<comment>",
        "<lastChangedByUserName>",
        <lastChangedTimestamp>
    ],
    ... ..
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

300 GetUsersTable

Detailed Description

GetUsersTable returns a list of all/active users the requesting user is allowed to see. It also returns whether user account deletion is enabled in system configuration.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getUsersTable",
    "data": {
        "showDisabled": true|false
    }
}
```

- "data:showDisabled" (optional): Boolean value to include or exclude disabled users in the response.

Returns

A HTTP response with the following content:

```
{
    "tableContent": [[
        <userId>,
        "<loginName>",
        "<userName>",
        "<comment>",
        "<enabled>true|false",
        "<associatedMandatorName>",
        <numberOfRoles>,
        <lastPasswordChangeTimestamp>,
        "<lastchangedByUsername>"
        <numberOfFailedLogin>,
        <lastChangedOn>,
        <lastLoginTimestamp>,
        true|false
    ]],
    ...
],
    "mayDeleteUser": 0|1,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


301 GetWorkingQueueCasesTable

Detailed Description

GetWorkingQueueCasesTable retrieves a table with all cases either of all public/private working queues, or one selected working queue, or all working queues the requesting user is allowed to view.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getWorkingQueueCasesTable",
    "uid": <workingQueueUid>,
    "type": "public"|"private"
}
```

- "uid" (optional): The uid of a working queue. This is used when querying for one certain working queue.
- "type" (optional): "public" or "private". This is used when querying for all public/private working queues.

Neither "uid" nor "type" have to be set. If both are omitted, all accessible working queues are queried.

Returns

A HTTP response with the following content:

```
{
    "header": [{
        "uid": "<uid>",
        "name": "<name>",
        "align": "<align>",
        "type": "<type>",
        "format": "<format>"
    },
    ...
    ],
    "tableContent": [{
        "uid": <caseId>,
        "Queue": "<caseQueueName>",
        "Generated": <generatedTimestamp>,
        "Score": <scoreValue>,
        "Hits": <numberOfHits>,
        "Status": "<status>",
        "FraudStatus": "<fraudStatus>",
        "CloseCodeName": "<caseCloseCodeName>",
        "User": <userId>,
        "UserName": "<userName>",
    }
    ]
}
```

```
        "LastUserAction": <lastUserActionTimestamp>,
        "MandatorName": "<mandatorName>",
        "Mandator": <mandatorUid>,
        "CppName": "<cppName>",
        "Cpp": "<cppCode>",
        ... ..
    },
    ... ..
],
"responseStatus": ["<status>", "feedbacktext"],
"reloadProfile": true|false
}
```

302 GetWorkingQueues

Detailed Description

GetWorkingQueues lists all working queues. It determines which working queues are suitable for selection in certain situations. The content of the response depends on the type that is passed with the request as well as the privileges of the requesting user. So far, "caseFilter" is the only available type.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getWorkingQueues",
    "type": "caseFilter"
}
```

- "revision" (mandatory): The Revision (identified by its uid) running a simulation in which to execute the SimulationQuery.
- "data" (mandatory): Either "ruleUid" or "parameter" have to be set.
- "data:parameter" (optional): The name of the SimulationQuery.
- "data:ruleUid" (optional): The uid of the SimulationQuery.

Returns

A HTTP response with the following content:

```
{
    "workingQueues": [{
        "uid": <workingQueueUid>,
        "name": "<workingQueueName>"
    },
    ...
    ],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

303 GetWorkingQueuesTable

Detailed Description

GetWorkingQueuesTable returns a table of all defined working queues the requesting user can access

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getWorkingQueuesTable"
}
```

Returns

A HTTP response with the following content:

```
{
    "tableContent": [
        <workingQueueUid>,
        <mandatorUid>,
        "<mandatorName>",
        <enabled>"Yes"|"No",
        "<workingQueueName>",
        "<comment>",
        <priority>,
        <numberOfUsers>,
        <numberOfQueueManagers>,
        "<lastChangedByUserName>",
        <lastChangedOnTimestamp>,
        "<conditions>"
    ],
    ... ..
],
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

304 GetWorkingQueueStatistics

Detailed Description

GetWorkingQueueStatistics Shows the statistics of all cases in the user's working queue(s). The statistics include: Number of cases, oldest open case, average age, included case classes and included case states.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "getWorkingQueueStatistics",
    "type": "private"|"public"
}
```

- "type" (optional): Compute values for all "private" or "public" working queues. If "type" is omitted, values for all working queues are computed.

Returns

A HTTP response with the following content:

```
{
    "caseQueues": [{
        "name": "<caseClassName>",
        "value": "<numberOfCases>"
    },
    ...
    ],
    "caseStates": [{
        "name": "<caseStateName>",
        "value": "<numberOfCases>"
    },
    ...
    ],
    "general": {
        "numberOfCases": "<totalNumberOfCases>",
        "lead": "<lead>",
        "oldestOpen": "<ageOfOldestOpenCaseInDays>",
        "averageAge": "<averageAgeInDays>"
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

305 GoliveCheckAndReport

Detailed Description

GoliveCheckAndReport checks if a revision can golive and create a detailed report. Before going live with a challenger revision, goliveCheckAndReport needs to be executed. This request tests whether there are any disallowed configurations that might prohibit golive. Furthermore, it creates a detailed golive report which informs about things such as memory/disc size consumption of the new champion, new elements to be created, eventual misconfigurations and others. This request leaves the revision in "Golive waiting for confirmation" status. It must be responded to either by "cancelGolive" or "confirmGolive" request.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "goliveCheckAndReport",
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "mandator" (mandatory): The mandator to which the challenger revision belongs.
- "revision" (mandatory): The challenger revision to be checked.

Returns

A HTTP response with the following content:

```
{
    "goliveReport": [
        ... ..
    ],
    "goliveEnabled": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

306 HasEntry

Detailed Description

HasEntry checks if a given entry exists in a defined risk list. For a given defined risk list and entry value, the hasEntry request inquires whether the defined risk list already stores an entry with the given value. The user needs the privilege for "Defined risk list entries...".

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "hasEntry",
    "uid": <definedRiskListUid>,
    "data": {
        "value": "<entryInputAttributeValue>"
    }
}
```

- "uid" (mandatory): The uid of the defined risk list which is checked for the entry.
- "data:value" (mandatory): The value which is searched in the given defined risk list.

Returns

A HTTP response with the following content:

```
{
    "valueExists": true|false
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

307 InterruptCase

Detailed Description

InterruptCase interrupts an ongoing case investigation and sets the case free from its reservation.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "interruptCase",
    "uid": <caseUid>
}
```

- "uid" (mandatory): The uid of the case which is to be interrupted.

Returns

A HTTP response with the following content:

```
{
    "data": {
        "streamType": "case",
        "case": {
            "uid": <caseUid>,
            "createdOnInstance": <instanceId>,
            "localCaseUid": <caseUid>,
            "mandator": {
                "uid": <mandatorUid>,
                "name": "<mandatorName>"
            },
            "caseQueue": {
                ... ..
            },
            ... ..
        },
        ... ..
    },
    "editable": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


308 InvestigateCase

Detailed Description

InvestigateCase retrieves the data of the selected case and at the same time reserves it for the requesting user, so that no other user may work on it at the same time.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "investigateCase",
    "uid": <caseUid>
}
```

- "uid" (mandatory): The uid of the case which is to be investigated

Returns

A HTTP response with the following content:

```
{
    "data": {
        "streamType": "case",
        ... ..
    },
    "editable": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

309 IsCasesTableEncrypted

Detailed Description

IsCasesTableEncrypted asserts whether the table of cases that meet the given filter criteria contains encrypted values.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "isCasesTableEncrypted",
  "data": {
    "mandators": [
      <mandatorUid>,
      ... ...
    ],
    "caseQueues": [
      <caseQueueUid>,
      ... ...
    ],
    "dateRestriction": {
      "all": true|false,
      "from": <fromTimestamp>,
      "to": <toTimestamp>
    },
    "status": {
      "New": true|false,
      "Investigated": true|false,
      "Followup": true|false,
      "Due": true|false,
      "Closed": true|false,
      "Reopened": true|false
    },
    "users": [<userId>],
    "scoreFrom": <caseScoreFromValue>,
    "scoreTo": <caseScoreToValue>,
    "cpps": ["<cppUid_caseGroupUid>"],
    "conditions": [...]
  }
}
```

- "data" (mandatory): Contains the filter criteria for the cases table.
- "data:mandators" (optional): Include only cases from these mandators
- "data:caseQueues" (optional): Include only cases from these case queues
- "data:dateRestriction:all" (optional): Boolean value to indicate whether cases of any age should be included.
- "data:dateRestriction:from" (optional): If not dateRestriction:all, include only cases younger than this value.

- "data:dateRestriction:to" (optional): If not dateRestriction:all, include only cases older than this value.
- "data:status:New" (optional): Boolean value to indicate whether cases of status New shall be included.
- "data:status:Investigated" (optional): Boolean value to indicate whether cases of status Investigated shall be included.
- "data:status:Followup" (optional): Boolean value to indicate whether cases of status Followup shall be included.
- "data:status:Closed" (optional): Boolean value to indicate whether cases of status Closed shall be included.
- "data:status:Reopened" (optional): Boolean value to indicate whether cases of status Reopened shall be included.
- "data:users" (optional): Include only cases which are investigated/on followup by these users.
- "data:scoreFrom" (optional): Include only cases that have a score higher than this value.
- "data:scoreTo" (optional): Include only cases that have a score lower than this value.
- "data:cpps" (optional): Include only cases which are assigned to these cpps.
- "data:conditions" (optional): Include only cases which satisfy these conditions.

If any of the lists of mandators, caseQueues and working queues is empty or if all status flags are false, no cases can be found.

Returns

A HTTP response with the following content:

```
{
  "isEncrypted": true|false,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

310 IsCppTableEncrypted

Detailed Description

IsCppTableEncrypted asserts whether a table of cpps which meets given filter criteria contains any encrypted values.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "isCppTableEncrypted",
  "data": {
    "showAllEntries": true|false,
    "caseGroups": [<caseGroupUid>, ...],
    "dateRestriction": {
      "all": true|false,
      "from": <fromTimestamp>,
      "to": <toTimestamp>
    },
    "status": {
      "New": true|false,
      "Investigated": true|false,
      "Followup": true|false,
      "Due": true|false,
      "Closed": true|false,
      "Reopened": true|false
    },
    "active": {
      "Active": true|false,
      "Inactive": true|false
    }
  }
}
```

- "data" (mandatory): Contains the filter criteria
- "showAllEntries" (mandatory): Boolean flag to decide whether to include all accessible cpps.
- "caseGroups" (optional): Include only cpps which belong to these case groups.
- "dateRestriction:all" (optional): If set, include cpps of all age.
- "dateRestriction:from" (optional): If not dateRestriction:all, include only cpps younger than this value.
- "dateRestriction:to" (optional): If not dateRestriction:to, include only CPPs older than this value.
- "status:New" (optional): Include only cpps with status New.
- "status:Investigated" (optional): Include only cpps with status Investigated.
- "status:Followup" (optional): Include only cpps with status Followup.

- "status:Due" (optional): Include only cpps with status Due.
- "status:Closed" (optional): Include only cpps with status Closed.
- "status:Reopened" (optional): Include only cpps with status Reopened.
- "active:Active" (optional): Include active cpps.
- "active:Inactive" (optional): Include inactive cpps.

If "showAllEntries" is true, all other values can be omitted.

Returns

A HTTP response with the following content:

```
{
  "isEncrypted": true|false,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

311 IsDdcEnabled

Detailed Description

IsDdcEnabled returns the settings about DDC enablement. It informs whether including ddc is allowed for mergings, merchantMonitoringRules, preprocessings, simulation or queries.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isDdcEnabled"
}
```

Returns

A HTTP response with the following content:

```
{
    "ddcEnabled": {
        "mergings": true|false,
        "merchantMonitoringRules": true|false,
        "preprocessings": true|false,
        "simulation": true|false,
        "query": true|false
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

312 IsDdcEnabledForManualCaseCreation

Detailed Description

IsDdcEnabledForManualCaseCreation returns whether include DDC is enabled for manual case creation. It informs whether manually created cases from a query result may include data which is retrieved from disk storage.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isDdcEnabledForManualCaseCreation"
}
```

Returns

A HTTP response with the following content:

```
{
    "caseCreationDdc": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

313 IsDefinedRiskListEncrypted

Detailed Description

IsDefinedRiskListEncrypted checks whether a defined risk list's input attribute, output attribute or entries are encrypted.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isDefinedRiskListEncrypted",
    "uid": <definedRiskListUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): The uid of the checked defined risk list
- "mandator" (mandatory): The mandator to which the defined risk list belongs

Returns

A HTTP response with the following content:

```
{
    "isEncrypted": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


314 IsGoliveRunning

Detailed Description

IsGoliveRunning check if there is any golive currently running. It returns true if any golive is currently running for any mandator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isGoliveRunning"
}
```

Returns

A HTTP response with the following content:

```
{
    "goliveRunning": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

315 IsGroupByQueryResultEncrypted

Detailed Description

IsGroupByQueryResultEncrypted checks whether a group by query result contains encrypted values.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "isGroupByQueryResultEncrypted",
  "uid": <groupByQueryUid>,
  "data": {
    "downloadToken": "<downloadToken>",
    "resultId": <resultId>
  }
}
```

- "uid" (mandatory): The uid of the asserted group by query.
- "data:downloadToken" (mandatory): The downloadToken for the asserted result.
- "data:resultId" (optional): The id of the asserted result.

Returns

A HTTP response with the following content:

```
{
  "isEncrypted": true|false,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

316 IsIndexSequenceMdcOutsideSimulation

Detailed Description

IsIndexSequenceMdcOutsideSimulation check if an index sequence is fully available in MDC for a simulation in a given revision with a running simulation.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "isIndexSequenceMdcOutsideSimulation",
  "uid": <indexUid>,
  "revision": <revisionUid>,
  "data": {
    "sequenceType": "Source"|"Target"|"Both"
  }
}
```

- "uid" (mandatory): The uid of the index which is asserted
- "data:sequenceType" (optional): For peer indexes, determines whether "source" or "target" or "both" sequences shall be asserted.

Returns

A HTTP response with the following content:

```
{
  "data": {
    "outsideSimulation": true|false
  },
  "responseStatus": [<status>, <feedbacktext>],
  "reloadUserProfile": true|false
}
```

317 IsMetaAttributeInChampions

Detailed Description

IsMetaAttributeInChampions checks whether a specified meta attribute is defined in any champion revision. It takes any meta attribute type and returns a boolean value to indicate whether this meta attribute type is defined in any champion revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "isMetaAttributeInChampions",
  "data": {
    "metaAttributeType": "<metaAttributeType>"
  }
}
```

- "data:metaAttributeType" (mandatory): The checked meta attribute type. The available values for the "metaAttributeType" are:

```
"Amount"
"Timestamp"
"Account"
"Fraud"
"Intercept"
"Email"
"CaseScore"
"CaseQueue"
"MessageTypeId"
"Sequence"
"SystemTime"
"Notification"
"PrimaryInstanceId"
"PrimaryUrid"
"URID"
"UridComputationComplete"
"Reminder"
"Score"
```

Returns

A HTTP response with the following content:

```
{
  "metaAttributeExists": true|false,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

318 IsOffline

Detailed Description

IsOffline checks if an IBM Safer Payments instance is offline.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isOffline",
    "uid": <instanceUid>
}
```

- "uid" (mandatory): The uid of an instance to be checked.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": "OK", "INSTANCE_IS_ONLINE" | "INSTANCE_IS_FINISHING_DEFERRED_WRITING"
| "INSTANCE_IS_OFFLINE",
    "reloadUserProfile": true|false
}
```

319 IsPasswordValid

Detailed Description

IsPasswordValid checks whether a password fulfills the password policies that are configured in system configuration.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isPasswordValid",
    "data": {
        "password": "<newPassword>"
    }
}
```

- "password" (mandatory): The password which is to be checked against policies.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

320 IsQueryResultEncrypted

Detailed Description

IsQueryResultEncrypted Checks whether a given query result contains encrypted values.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "isQueryResultEncrypted",
  "uid": <queryUid>,
  "revision": <revisionUid>,
  "data": {
    "downloadToken": "<downloadToken>",
    "resultId": <resultId>
  }
}
```

- "uid" (mandatory): The uid of the asserted query.
- "revision" (optional): A revision is only required when asserting a simulation query.
- "data:downloadToken" (mandatory): The downloadToken for the asserted result.
- "data:resultId" (mandatory): The id of the asserted result.

Returns

A HTTP response with the following content:

```
{
  "isEncrypted": true|false,
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

321 IsRevisionDeletionEnabled

Detailed Description

IsRevisionDeletionEnabled checks whether revision deletion is enabled. Revision deletion is only possible when configured in system configuration. It is configured separately for different types of revisions: challengers, invalidated revisions and retired revisions. This request returns the deletion setting for all of these types.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isRevisionDeletionEnable"
}
```

Returns

A HTTP response with the following content:

```
{
    "deletionEnabled": {
        "Challenger": true|false,
        "Invalidated": true|false,
        "Retired": true|false
    },
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


322 IsSamplingEnabled

Detailed Description

IsSamplingEnabled checks whether sampling is enabled for a given request type from system configuration.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{  
    "request": "isSamplingEnabled"  
}
```

Returns

A HTTP response with the following content:

```
{  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

323 IsSimulationValid

Detailed Description

IsSimulationValid checks whether the simulation in a given challenger revision is valid. It returns true if the simulation of the given challenger revision is valid, i.e. running. Otherwise it returns false.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isSimulationValid",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision identifies the simulation which is checked.

Returns

A HTTP response with the following content:

```
{
    "simulationValid": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

324 IsValidInstanceId

Detailed Description

IsValidInstanceId verifies if a new instance id is valid.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isValidInstanceId",
    "uid": <instanceId>
}
```

- "uid" (mandatory): The uid of an instance to be checked.

Returns

A HTTP response with the following content:

```
{
    "isValid": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

325 IsWorkingQueueCasesTableEncrypted

Detailed Description

IsWorkingQueueCasesTableEncrypted asserts whether a working queue cases table contains any encrypted values. This request can be used to assert all public/private working queues, or one selected working queue, or all working queues the requesting user is allowed to view.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "isWorkingQueueCasesTableEncrypted",
    "uid": <workingQueueUid>,
    "type": "public"|"private"
}
```

- "uid" (optional): The uid of a working queue. This is used when querying for one certain working queue.
- "type" (optional): "public" or "private". This is used when querying for all public/private working queues.

Neither "uid" nor "type" have to be set. If both are omitted, all accessible working queues are queried.

Returns

A HTTP response with the following content:

```
{
    "isEncrypted": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

326 LoadCaseAuditTrailsFromDisk

Detailed Description

LoadCaseAuditTrailsFromDisk reloads the audit trails of a case from disk.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "loadCaseAuditTrailsFromDisk",
    "uid": <caseUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

327 Logout

Detailed Description

Logout is used to log out of the current session.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "logout"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

328 MarkFraud

Detailed Description

MarkFraud manually marks transactions as fraud after they are processed. The user can choose a custom fraud value and decide to provide a timestamp attribute which saves the time of fraud marking.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "markFraud",
  "data": {
    "urid": [<urid>],
    "fraudValue": <manualFraudValue>,
    "fraudTimestampAttribute": <timestampAttributeUid>|-1
  }
}
```

- "data:urid" (mandatory): A list of URIDs which shall be marked as fraud.
- "data:fraudValue" (mandatory): The fraud value which shall be assigned to the selected URIDs.
- "fraudTimestampAttribute" (optional): Provide a fraud timestamp attribute to save the time of fraud marking.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

329 MoveRules

Detailed Description

MoveRules moves the specified rules to a different ruleset.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "moveRules",
    "revision": <revisionUid>,
    "ruleset": <targetRulesetUid>,
    "data": {
        "rules": [<ruleUid>]
    }
}
```

- "revision" (mandatory): The revision from which the rules are copied.
- "ruleset" (mandatory): The target ruleset to which the rules shall be copied.
- "data:rules" (mandatory): A list of rule uids that shall be copied.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


330 MultiDelete

Detailed Description

MultiDelete goes through a provided list of elements and deletes all of them.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "multiDelete",
  "revision": <revisionUid>,
  "data": {
    "items": [<elementUid>, ...]
  }
}
```

- "revision" (optional): Provide a revision when deleting revision elements
- "data:items" (mandatory): A list with uids of the items to delete

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

331 ParentPathExists

Detailed Description

ParentPathExists checks whether the parent path a specified path is available on the system.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "parentPathExists",
    "data": {
        "path": "<path>"
    }
}
```

- "data:path": The path which is to be checked.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

332 PathExists

Detailed Description

PathExists checks if the specified path is available on the system.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "pathExists",
    "data": {
        "path": "<path>"
    }
}
```

- "data:path": The path which is to be checked.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

333 ReAttach

Detailed Description

ReAttach reattaches an instance to the Safer Payments cluster. The requesting User must have the privilege to change the Safer Payments cluster.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "reAttach"
    "uid": <instanceUid>
}
```

- "uid" (mandatory): The uid of the instance to reattach.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

334 RebuildAllIndexes

Detailed Description

RebuildAllIndexes rebuilds all indexes of all champions of all mandators. The requesting User must have the privilege to execute Index related maintenance functions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "rebuildAllIndexes"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

335 RebuildIndex

Detailed Description

RebuildIndex rebuilds a single Index using stored transaction data. The requesting User must have the privilege to execute Index related maintenance functions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "rebuildIndex",
    "uid": <indexUid>,
    "mandator": <mandatorUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The Index to rebuild.
- "mandator" (mandatory): The Mandator (identified by its uid) to search the Index in.
- "revision" (optional): The Revision (identified by its uid) to search the Index in. If not provided the Mandator's next champion is used.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

336 RebuildIndexSequence

Detailed Description

RebuildIndexSequence rebuilds a single Index sequence using stored transaction data. The requesting User must have the privilege to execute Index related maintenance functions.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "rebuildIndexSequence",
    "uid": <indexUid>,
    "mandator": <mandatorUid>,
    "data": {
        "threadNumberSequenceRebuild": <noOfThreads>
    }
}
```

- "uid" (mandatory): The Index to rebuild.
- "mandator" (mandatory): The Mandator (identified by its uid) to search the Index in.
- "data:threadNumberSequenceRebuild" (optional): Number of threads to be used to rebuild index sequence. Default is 1.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

337 RecomputeAttributes

Detailed Description

RecomputeAttributes recomputes attributes in rule generation.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "recomputeAttributes",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): Revision uid to identify the rule generation of which you want to recompute attributes.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


338 ReGoliveRevision

Detailed Description

ReGoliveRevision re-golives a retired model revision and put it back to champion.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "reGoliveRevision",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The uid of the retired revision which shall re-golive.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

339 ReloadComplianceLists

Detailed Description

ReloadComplianceLists reloads all compliance lists from disk. This is used e.g. when a new version of a compliance list is deployed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "reloadComplianceLists"
}
```

- "parameterName" (mandatory/optional): Description

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

340 ReloadPrivateKeys

Detailed Description

ReloadPrivateKeys reloads private keys from keys folder on disk.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{  
    "request": "reloadPrivateKeys",  
}
```

Returns

A HTTP response with the following content:

```
{  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

341 ReloadPrivateKeysPasswordSafe

Detailed Description

ReloadPrivateKeysPasswordSafe reloads private keys in password safe path.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "reloadPrivateKeysPasswordSafe"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

342 Reserve

Detailed Description

Reserve reserves an element for the current user to make potential changes. Before editing an element, it is necessary to reserve it. This prevents multiple users from editing an element at the same time.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "reserve",
    "uid": <elementUid>,
    "mandator": <mandatorUid>
}
```

- "uid" (mandatory): The uid of the element which is to be reserved.
- "mandator" (mandatory): The mandator to which the element belongs.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

343 ResetAllUserPreferences

Detailed Description

ResetAllUserPreferences resets all preferences of all users to the default preferences. After this is performed, all users logged in should log out immediately and re-login. Otherwise the resetted preferences may be overwritten unintentionally. This request requires the global privilege for the maintenance function of "Reset user preferences".

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "resetAllUserPreferences"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

344 ResetIndex

Detailed Description

ResetIndex resets the internal storage of an index and all of its dependent elements such as masterdata, calendar profiles and events. MCI, FLI, and BDI must be disabled in order for this to work.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "resetIndex",
    "uid": <indexUid>,
    "mandator": <mandatorUid>
}
```

- "uid": The uid of the index to be reset
- "mandator": The uid of the mandator the index belongs to

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

345 ResetOutgoingFli

Detailed Description

ResetOutgoingFli resets all outgoing FLI connections and rewinds FLI buffer read position to first unacknowledged FLI message. This can be useful, if the outgoing FLI is not connecting to other instances anymore.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "resetOutgoingFli"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


346 ResetPreference

Detailed Description

ResetPreference resets a single preferences of the requesting User. Preferences include the order and width of table columns, the number of displayed rows, the size of help dialogs, etc.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "resetPreference",
    "type": <type>
}
```

- "type" (mandatory): Controls which preferences should be reset.

Returns

A HTTP response with the following content:

```
{
    "preference": {
        ... ..
    },
    "responseStatus":["<status>", "<feedbackText>"],
    "reloadUserProfile":true|false
}
```

347 ResetSandboxRecords

Detailed Description

ResetSandboxRecords clears any previously computed values of counters, events, ... in the sandbox table of a revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "resetSandboxRecords",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision of which sandbox records table shall be cleared.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

348 ResetUserPreferences

Detailed Description

ResetUserPreferences resets all preferences of the requesting user to the default user preferences.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "resetUserPreferences"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

349 RestartInstance

Detailed Description

RestartInstance performs a hot restart for an instance in the cluster.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "restartInstance",
    "uid": <instanceUid>
}
```

- "uid": The uid of the instance to be restarted

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

350 Restore

Detailed Description

Restore restores an instance from a donor instance. This request can be used for example when an instance is invalid.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "restore",
    "uid": <instanceUid>,
    "data": {
        "donor": <donorUid>
    }
}
```

- "uid": The uid of the instance to be restored
- "data:donor": The uid of the instance to be the donor

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

351 RetireChampion

Detailed Description

RetireChampion retires an active champion revision of a mandator. If the champion revision is retired, all stored data of elements defined in the model revision will be lost. It is not possible to make a re-golive after retiring a champion through this request, since an active champion is a prerequisite for a re-golive. This request should be considered only when the mandator that the champion belongs to is no longer needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "retireChampion",
    "mandator": <mandatorUid>
}
```

- "mandator" (mandatory): The champion revision to be retired is identified by the mandator

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

352 RevertToChallenger

Detailed Description

RevertToChallenger cancels the golive report. When performing goliveCheckAndReport, a revision is put into status "Golive waiting for confirmation". The revertToChallenger request can be used to abort confirmation and to revert the revision back to a normal challenger revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "revertToChallenger",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): Uid of the revision which is to be reverted to challenger

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

353 RevisionElementExists

Detailed Description

RevisionElementExists checks whether a given revision element exists in the given revision.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "revisionElementExists",
    "uid": <elementUid>,
    "type": "<elementType>",
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The uid of the checked revision element.
- "type" (mandatory): The type of element that is expected. Available values for "elementType" are:

```
"Attribute"
"Counter"
"Collusion"
"Event"
"DeviceIdentification"
"Formula"
"Index"
"Masterdata"
"Merging"
"Pattern"
"Profile"
"Rule"
"Ruleset"
"FinalRuleset"
```

- "revision" (mandatory): The revision in which the element should be found.

Returns

A HTTP response with the following content:

```
{
    "status": 1,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

The "status" only shows in the response if the element exists.

354 RevokeKey

Detailed Description

RevokeKey revokes an encryption Key. The revoked keys are permanently removed from IBM Safer Payments and the file system. The requesting User must have the privilege to revoke encryption keys.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "revokeKey",
    "uid": <keyUid>
}
```

- "uid" (mandatory): The Key (identified by its uid) to revoke.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

355 RewindFliBuffer

Detailed Description

RewindFliBuffer rewinds FLI buffer read position to first unacknowledged FLI message. This can be useful, if the FLI is still sending but the unsent buffer size is not decreasing.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "rewindFliBuffer"
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

356 RuleReportQueryExists

Detailed Description

RuleReportQueryExists checks whether there is any rule report query defined in the given revision, which is a precondition to execute a rule report.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "ruleReportQueryExists",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision in which a rule report query is searched.

Returns

A HTTP response with the following content:

```
{
    "ruleReportQueryExists": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

357 Save

Detailed Description

Save is used to creates or updates an element. This request is universally used for any type of element which is identified by an uid. The type of element is identified by the "streamType" parameter.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "save",
  "uid": <elementUid>|-1,
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "streamType": "<streamType>",
    "uid": <elementUid>|-1,
    "name" : "<name>",
    "comment" : "<comment>",
    ... ..
  }
}
```

- "uid" (optional): If no uid is provided, a new element is created; if an uid is provided, an existing element is changed.
- "mandator" (mandatory/optional): The mandator to which the element belongs/shall belong.
- "revision" (mandatory/optional): The revision to which the element belongs/shall belong.
- "data" (mandatory): Contains the settings for the new element. This dictionary is very different for every element.
- "data:streamType" (mandatory): Determines which type of element is to be created. Available values for "streamType" are:

```
"caseAction"
"caseCloseCode"
"caseGroup"
"caseQueue"
"chart"
"eventLogMessages"
"externalQuery"
"loadFileJob"
"reportGenerationJob"
"keyPerformanceIndicator"
"mandator"
"message"
"notification"
"passwordSafe"
"realtimeInterceptionCode"
```

```
"reminders"  
"role"  
"statusAlarmIndicator"  
"textModule"  
"user"  
"input"  
"output"  
"collusion"  
"counter"  
"deviceIdentification"  
"event"  
"formula"  
"index"  
"list"  
"mapping"  
"masterdata"  
"merging"  
"pattern"  
"precedent"  
"profile"  
"rule"  
"ruleset"  
"finalRuleset"  
"sandboxEntry"  
"complianceList"  
"definedRiskList"  
"merchantMonitoringRule"  
"caseQueueReport"  
"investigationReport"  
"missedCasesReport"  
"userReport"
```

- "data:name" (mandatory): Sets the name of the new element.
- "data:comment" (optional): Sets a descriptive comment of the new element.
- "data:..." (mandatory): Various additional parameters for all different element types.

Returns

A HTTP response with the following content:

```
{  
    "responseStatus": ["<status>", "<feedbacktext>"],  
    "reloadUserProfile": true|false  
}
```

358 SaveCluster

Detailed Description

SaveCluster saves settings of a single instance. Depending on the changed settings, the requesting User must have the privilege to change the IBM Safer Payments cluster or to add/delete instances.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "saveCluster",
  "uid": <instanceUid>,
  "data": {
    "streamType": "irisInstance",
    "id": <instanceId>,
    "uid": <instanceUid>,
    "enabled": true|false,
    "name": "<instanceName>",
    "comment": "<comment>",
    "directories": {
      "arcPath": "<archivePath>",
      "cfgPath": "<configurationPath>",
      "ddcPath": "<ddcPath>",
      "emlPath": "<emailPath>",
      "fliPath": "<fliBufferPath>",
      "incPath": "<apiIncludesPath>",
      "invPath": "<investigationPath>",
      "keyPath": "<keyPath>",
      "kpiPath": "<kpiPath>",
      "logPath": "<logPath>",
      "rdiPath": "<rdiPath>",
      "repPath": "<messageReportPath>",
      "saiPath": "<saiPath>",
      "usrPath": "<userPath>",
      "pwsPath": "<passwordSafePath>"
    },
    "interfaces": [{
      "type": "MessageCommandInterface",
      "address": "<ipAddress>",
      "port": "<port>",
      "enabled": true|false,
      "acceptAll": true|false,
      "acceptFrom": ["<ipAddress>", ...],
      "enabledSsl": true|false,
      "certificateFile": "<certificateFilePath>",
      "privateKeyFile": "<privateKeyFilePath>",
      "diffieHellmanFile": "<diffieHellamFilePath>",
      "privateKeyType": "ConsolePassword" | "PasswordFile" | "Unencrypted",
      "privateKeyPasswordFile": "<privateKeyPasswordFilePath>",
      "rejectTls10": true|false,
      "rejectTls11": true|false,
    }
  ]
}
```

```

"clientCertificateFile": "<clientCertificateFilePath>",
"clientAuthorizationCertificateFile": "<clientAuthorizationCertificateFilePath>",
"requireValidClientCertificate": true|false,
"requireFittingCommonName": true|false,
"requireValidServerCertificate": true|false,
"revokedClientsCrlFile": "<revokedClientsCrlFilePath>"
}, {
"type": "ApplicationProgrammingInterface",
"address": "<ipAddress>",
"port": "<port>",
"enabled": true|false,
"acceptAll": true|false,
"acceptFrom": ["<ipAddress>"],
"enabledSsl": true|false,
"certificateFile": "<certificateFilePath>",
"privateKeyFile": "<privateKeyFilePath>",
"diffieHellmanFile": "<diffieHellamFilePath>",
"privateKeyType": "ConsolePassword" | "PasswordFile" | "Unencrypted",
"privateKeyPasswordFile": "<privateKeyPasswordFilePath>",
"rejectTls10": true|false,
"rejectTls11": true|false,
"clientCertificateFile": "<clientCertificateFilePath>",
"clientAuthorizationCertificateFile": "<clientAuthorizationCertificateFilePath>",
"requireValidClientCertificate": true|false,
"requireFittingCommonName": true|false,
"requireValidServerCertificate": true|false,
"revokedClientsCrlFile": "<revokedClientsCrlFilePath>"
}, {
"type": "BatchDataInterface",
"enabled": true|false
}, {
"type": "FastLinkInterface",
"address": "<ipAddress>",
"port": "<port>",
"enabled": true|false
}, {
"type": "StatusControlInterface",
"address": "<ipAddress>",
"port": "<port>"
}, {
"type": "EncryptedCommunicationInterface",
"address": "<ipAddress>",
"port": "<port>",
"enabled": true|false,
"enabledSsl": true|false,
"certificateFile": "<certificateFilePath>",
"privateKeyFile": "<privateKeyFilePath>",
"diffieHellmanFile": "<diffieHellamFilePath>",
"privateKeyType": "ConsolePassword" | "PasswordFile" | "Unencrypted",
"privateKeyPasswordFile": "<privateKeyPasswordFilePath>",
"rejectTls10": true|false,
"rejectTls11": true|false,
"clientCertificateFile": "<clientCertificateFilePath>",
"clientAuthorizationCertificateFile": "<clientAuthorizationCertificateFilePath>",
"privateClientKeyFile": "<privateClientKeyFilePath>",
"privateClientKeyType": "ConsolePassword" | "PasswordFile" | "Unencrypted",
"privateClientKeyPasswordFile": "<privateClientKeyPasswordFilePath>",
"serverAuthorizationCertificateFile": "<serverAuthrizationCertificationFilePath>",

```

```

"requireValidClientCertificate": true|false,
"requireFittingCommonName": true|false,
"requireValidServerCertificate": true|false,
"revokedServersCrlFile": "<revokedServersCrlFilePath>",
"revokedClientsCrlFile": "<revokedClientsCrlFilePath>"
}, {
"type": "AlertMessageInterface",
"enabled": true|false
}, {
"type": "WebSphereMQInterface",
"enabled": true|false,
"queueManagers": [{
  "uid": <uid>|-1,
  "queueManager": "<queueManagerName>",
  "enabled": true|false,
  "address": "<ipAddress>",
  "port": <port>,
  "useFallback": true|false,
  "fallbackAddress": "<ipAddress>",
  "fallbackPort": <port>,
  "channel": "<channel>",
  "protocol": "TCP",
  "reconnectInterval": <reconnectInterval>,
  "useSsl": true|false,
  "keyRepoStem": "<keyRepositoryPath>",
  "cipherSpec": "<sslCipherSpecification>",
  "responderUrl": "<responderUrl>",
  "queues": [{
    "uid": <uid>|-1,
    "queue": "<queueName>",
    "enabled": true|false,
    "message": <messageTypeUid>,
    "queueParameter": "<parameter>",
    "useRespondQueue": true|false,
    "respondQueue": "<responseQueueName>",
    "mqGetTimeout": "mqgetTimeoutSetting"
  ],
  ...
  ]},
  ...
  ]}
],
"monitoringDirectories": {
  "euslPath": "<europeanSanctionListFilePath>",
  "ofacPath": "<ofacSdnListFilePath>",
  "unPath": "<unitedNationsSanctionListPath>",
  "pepPath": "<politicalExposedPersonsListFilePath>",
  "ruslPath": "<russianSanctionListFilePath>"
}
}
}

```

- "uid" (mandatory): The IrisInstance to change, add or remove.
- "data" (mandatory): The IrisInstance's various settings.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

359 SaveCpp

Detailed Description

SaveCpp adds or updates an existing cpp (common point of purchase) for a case group to disk.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "saveCpp",
  "uid": <caseGroupUid>,
  "mandator": <mandatorUid>,
  "data": {
    "streamType": "cpp",
    "caseGroupUid": <caseGroupUid>,
    "mandatorUid": <mandatorUid>,
    "id": <cppId> | -1,
    "name": "<name>",
    "status": "New" | "Investigated" | "Followup" | "Due" | "Closed" | "Reopened"
    "comment": "<comment>",
    "reportingAttributes": [
      {
        "name": "<reportingAttributeName>",
        "type": "Numeric" | "Text" | "Timestamp" | "Boolean" | "Hex" |
"IPv4" | "TimeInterval",
        "format": <reportingAttributeFormat>,
        "attributeId": <reportingAttributeUid>,
        "encrypted": true | false,
        "value": <reportingAttributeValue>,
        "decimals": <reportingAttributeDecimals>,
        "length": <reportingAttributeLength>,
        "category": "",
      },
      ...]
    "active": true|false
  }
}
```

- "uid": The uid of the case group the CPP belongs to
- "mandator": The uid of mandator, for which the case group is defined
- "data:streamType": The stream type
- "data:caseGroupUid": The uid of the case group the CPP belongs to
- "data:mandatorUid": The uid of mandator, for which the case group is defined
- "data:id": The id of an existing CPP or -1 to create a new CPP
- "data:name": The name of the CPP

- "data:status": The status of the CPP
- "data:comment" (optional): A comment for the CPP
- "data:reportingAttributes": A list of reporting attributes with values for the CPP
- "data:active": Defines if the CPP should be active or inactive

Returns

A HTTP response with the following content:

```
{
  "comparisonResult": "Unchanged" | "ChangedNoImpact" | "ChangedComputationImpact",
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

360 SaveSandboxEntry

Detailed Description

SaveSandboxEntry sets an attribute value in a given sandbox record.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "saveSandboxEntry",
  "uid": <entryUid>,
  "revision": <revisionUid>,
  "data": {
    "attributeUid": <attributeUid>,
    "value": <value>
  }
}
```

- "uid" (mandatory): The uid of the sandbox record.
- "revision" (mandatory): The revision in which the sandbox record is found.
- "data:attributeUid" (mandatory): The attribute for which the value is to be set.
- "data:value" (mandatory): The value which is to be set for the given attribute.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

361 SaveSettings

Detailed Description

SaveSettings saves any setting of system configuration.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "saveSettings",
  "data": {
    "settings": {
      "application": {
        "name": "<applicationName>",
        "serverTimeZone": <serverTimezone>
      },
      "memory": {
        "limit": <mainMemoryUsageLimitInGB>
      },
      "userAccounts": {
        "maxFailedLogin": <maximumFailedLogins>,
        "numberOldPasswordRejected": <numberOfOldPasswordsRejected>,
        "passwordValiditySeconds": <passwordValidityInSeconds>,
        "minPasswordLength": <passwordMinimumLength>,
        "passwordMustContainLowerCase": true|false,
        "passwordMustContainUpperCase": true|false,
        "passwordMustContainDigit": true|false,
        "passwordMustContainSpecialCharacter": true|false,
        "userAccountDeletion": true|false,
        "enableExtendedAuth": true|false,
        "enableSystemUser": true|false,
        "maximumUserInactivityPeriodDays": <maximumUserInactivityPeriodDays>,
        "restrictSystemUserIP": ["<ipAddressThatAllowSystemUsersToAccessAPIFrom>"]
      },
      "ldap": {
        "login": "local" | "ldap" | "both",
        "useActiveDirectory": true|false,
        "host": "<activeDirectoryServerIpAddress>",
        "port": <activeDirectoryServerPort>,
        "baseDn": "<baseDN>",
        "domain": "<domain>",
        "ssl": true|false,
        "timeout": <timeoutSetting>,
        "sso": true|false,
        "ssoAutoReLogin": true|false
      },
      "priorisation": {
        "enabled": true|false,
        "irisProcess": "Idle" | "BelowNormal" | "Normal" | "AboveNormal" | "High" |
"Realtime",
        "threads": {
```

```

    "interfaces": {
      "messageCommand": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
      "statusControl": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
      "applicationProgramming": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
      "batchData": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
      "fastLink": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
      "alertMessage": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical"
    },
    "caseManagement": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
    "deferredWriter": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
    "analysis": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
    "elementGeneration": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
    "statusAlarmIndicators": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
    "keyPerformanceIndicators": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
    "endOfDay": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
    "golive": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
    "simulation": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
    "query": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
    "reminder": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" | "Highest" |
"TimeCritical",
    "instanceStatusFileService": "Idle" | "Lowest" | "BelowNormal" | "Normal" |
"AboveNormal" | "Highest" | "TimeCritical",
    "definedRiskListEntry": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical"
  }
},
"deferredWriting": {
  "enabled": true|false,
  "resetChunkingForShutdownEnabled": true|false,
  "timeGap": <timeGap>,
  "minimumMdcSize": <minimumMdcSize>,
  "safetyMargin": <safetyMargin>,
  "pauseAfterElement": <pauseAfterElement>,
  "diskChunkDelay": <diskChuckDelay>,
  "diskChunkSize": <diskChuckSize>,
  "shadowCommitChunkSize": <shadowCommitChunkSize>,
  "shadowCommitChunkDelay": <shadowCommitChunkDelay>
},
"numberOfThreads": {
  "messageCommandInterface": <numberOfThreadsForMessageCommandInterface>,
  "batchDataInterface": <numberOfThreadsForBatchDataInterface>,
  "fastLinkInterface": <numberOfThreadsForFastLinkInterface>,

```

```

"analysis": <numberOfThreadsForAnalyses>,
"elementGeneration": <numberOfThreadsForModelElementGeneration>,
"simulation": <numberOfThreadsForSimulation>,
"query": <numberOfThreadsForQuery>,
"file": <numberOfThreadsForFileNotifications>,
"message": <numberOfThreadsForMessageNotifications>,
"http": <numberOfThreadsForHTTPNotifications>,
"odbcSql": <numberOfThreadsForSqlNotifications>,
"smtp": <numberOfThreadsForSmtptNotifications>,
"templateFile": <numberOfThreadsForDocxNotifications>
},
"pageAutoRefresh": {
  "dashboard": <pageAutoRefreshInSecondsForDashboard>,
  "investigation": <pageAutoRefreshInSecondsForInvestigation>,
  "analysis": <pageAutoRefreshInSecondsForAnalyses>,
  "jobSchedule": <pageAutoRefreshInSecondsForJobSchedule>,
  "elementGeneration": <pageAutoRefreshInSecondsForModelElementGeneration>,
  "golive": <pageAutoRefreshInSecondsForModelGolive>,
  "cluster": <pageAutoRefreshInSecondsForClusterAdministration>,
  "keys": <pageAutoRefreshInSecondsForEncryptionKeys>,
  "query": <pageAutoRefreshInSecondsForQueryResult>,
  "memoryManagement": <pageAutoRefreshInSecondsForMemoryManagement>,
  "sessionCountdown": <pageAutoRefreshInSecondsForSessionCountdown>
},
"investigation": {
  "caseGenerationEnabled": true|false,
  "manualFraudValue": <manualFraudValue>,
  "updateProfilesAndEventsByManualFraudMark": true|false,
  "maxCasesSelectionTable": <maximumCasesShownOnSelectionTable>,
  "maxRecCaseHistory": <maximumCasesShownInHistory>,
  "caseConsolidationRepeat": <caseConsolidationStartsEveryNumberOfSeconds>,
  "archiveCasesAfterSeconds": <archiveCasesAfterNumberOfSeconds>,
  "enableCaseAttachments": true|false,
  "maxCaseAttachmentSizeMB": <maximumCaseAttachmentSizeInMB>,
  "caseCreationDdc": true|false
},
"monitoring": {
  "monitoringResultLifetime": <monitoringResultLifetimeInSeconds>,
  "monitoringMaxTableEntries": <maximumNumberOfDisplayedEntries>,
  "monitoringComplianceEnabled": true|false,
  "monitoringComplianceEuslEnabled": true|false,
  "monitoringComplianceOfacEnabled": true|false,
  "monitoringComplianceUNEnabled": true|false,
  "monitoringComplianceGlobalWatchListEnabled": true|false,
  "monitoringCompliancePepEnabled": true|false,
  "monitoringComplianceRuslEnabled": true|false,
  "monitoringComplianceWaitFactor": <reloadWaitFactor>,
  "definedRiskListDeleteEnabled": true|false,
  "definedRiskListMassDeleteEnabled": true|false,
  "definedRiskListMassDisableEnabled": true|false,
  "definedRiskListEntryConsolidationRepeat": <consolidationStartsEveryNumberOfSeconds>
},
"model": {
  "enforcePeerConfirmation": true|false,
  "rebuildIndexOnSizeChange": true|false,
  "enableDdcInModel": true|false,
  "enableDdcInMerchantMonitoringRules": true|false,
  "enableDdcInMergings": true|false,

```

```

"numberOfCategoriesInAuditTrail": <maximumNumberOfCategoryChanges>,
"numberOfStoredChampionAuditTrailEntries": <storedChampionAuditTrailEntries>,
"numberOfDisplayedChampionAuditTrailEntries": <displayedChampionAuditTrailEntries>,
"numberOfDisplayedCollusionSimulationResults": <displayedCollusionSimulationResults>,
"goliveRelativeWaitFactor": <remoteGoliveWaitFactor>,
"treatGoliveWarningsAsErrors": true|false,
"defaultModelling": {
  "simulationMayUseDdc": true|false,
  "categoryLimitModelling": <maximumComputedIndicators>
},
"defaultCapacities": {
  "mdc": <mdcCapacity>,
  "ddc": <ddcCapacity>,
  "ddcStored": true|false,
  "mdcStored": true|false
},
"deletion": {
  "challenger": true|false,
  "invalidated": true|false,
  "retired": true|false
}
},
"interfaces": {
  "serializeComputation": {
    "enabled": true|false,
    "mergingProtectionEnabled": true|false,
    "maxMergingProtectionTries": <maxMergingProtectionTries>,
    "mergingProtectionWaitMsec": <mergingProtectionWaitMsec>,
    "mergingProtectionFailOnTimeout": true|false
  },
  "statusControl": {
    "receiveTimeoutSeconds": <receiveTimeoutInSeconds>
  },
  "messageCommand": {
    "mciFormat": "ISO20022" | "ISO8583",
    "mciLatencyViolationThreshold": <latencyViolationThresholdInMilliseconds>,
    "mciLatencyViolationLoggingEnabled": true|false,
    "useOnlyPrintableAscii": true|false,
    "closeDuringGolive": true|false,
    "responseMetaInformation": {
      "respondInstanceName": true|false,
      "respondMtid": true|false,
      "respondSystemTime": true|false,
      "respondUrid": true|false,
      "respondMessageId": true|false,
      "respondModels": true|false,
      "respondMergingSource": true|false,
      "respondStatus": true|false,
      "respondLatency": true|false,
      "respondErrorText": true|false,
      "maxDocumentedRulesInResponse": <respondMaxFiredRules>
    },
    "messageTracing": {
      "showIpData": true|false,
      "dumpMalformattedMessages": <dumpMalformattedMessages>
    },
    "threadBufferSize": {
      "incomingBuffer": <incomingBufferInBytes>,

```



```

    "responseBuffer": <responseBufferInBytes>
  }
},
"applicationProgramming": {
  "lockTimeoutSeconds": <sessionTimeoutInSeconds>,
  "lockTimeoutCountdownThresholdSeconds": <sessionTimeoutCountdownThresholdInSeconds>,
  "enableCsrf": true|false,
  "useDefaultHttpHeaders": true|false,
  "useHstsHttpHeaders": true|false,
  "customHttpHeaders": "<customHttpHeaders>",
  "useGzip": true|false,
  "useKeepAlive": true|false,
  "downloadRequests": true|false,
  "maxPostRequestSizeMB": <maximumPostRequestSizeInMB>,
  "defaultLanguage": "en",
  "defaultDateTimeFormat": "US" | "EUR" | "ISO",
  "defaultDecimal": "." | ",",
  "defaultDigitGroup": "," | "'" | ".",
  "defaultFieldSeparator": "tab" | "," | ";",
  "defaultExtendedSelectDialog": true|false,
  "defaultSearchInSelections": true|false,
  "defaultShowExplanation": true|false,
  "defaultShowUids": true|false,
  "colorSchemaFraudulentValues": "#<colorHex>, #<colorHex>, #<colorHex>, #<colorHex>,
#<colorHex>",
  "colorSchemaGenuineValues": "#<colorHex>, #<colorHex>, #<colorHex>, #<colorHex>,
#<colorHex>",
  "colorSchemaOtherValues": "#<colorHex>, #<colorHex>, #<colorHex>, #<colorHex>,
#<colorHex>"
},
"batchData": {
  "jobRepeat": <jobSchedulerStartsEveryNumberOfSeconds>,
  "prioMaxJobThread": "Idle" | "Lowest" | "BelowNormal" | "Normal" | "AboveNormal" |
"Highest" | "TimeCritical",
  "jobEncryptionEnabled": true|false
},
"fastLink": {
  "receiveTimeoutSeconds": <receiveTimeoutInSeconds>,
  "retryConnectionSeconds": <retryConnectionInSeconds>,
  "numMessagesToleratedInSync": <synchronisationThreshold>,
  "outgoingBufferSize": <bufferSizeInGB>,
  "jobProcessingBrakeAt": <jobProcessingFreezeAtNumberOfPercentageOfBufferSize,
valueShouldBeInDecimal>,
  "transactionMessageProcessingHaltAt": <closeMCIAtNumberOfPercentageOfBufferSize,
valueShouldBeInDecimal>
},
"alertMessage": {
  "ipAddress": "<ipAddress>",
  "ipPort": "<port>",
  "receiveTimeoutSeconds": "<receiveTimeoutInSeconds>",
  "authentication": "AuthLogin" | "NoAuth",
  "username": "<userName>",
  "password": "<password>",
  "from": "<fromEmailAddress>",
  "sendInterval": <sendIntervalInSeconds>,
  "retryInterval": <retryIntervalInSeconds>,
  "archive": true|false,
  "useSsl": true|false,

```

```

    "verificationCertificate": "<serverCACertificateFile>"
  },
  "query": {
    "maxRecordsWarning": <maximumNumberOfRecordsWarningAt>,
    "maxRecordsLimit": <maximumNumberOfRecordsLimitAt>,
    "groupingQueryAccountsLimit": <groupByQueryAccountsLimit>,
    "groupingQueryResultCategoriesLimit": <groupByQueryCategoriesLimit>,
    "queryEnableDdc": "Available" | "Never" | "Default",
    "queryResultLivetime": <queryResultLifetimeInSeconds>,
    "commonPointQueryResultLivetimeSeconds": <commonPointResultLifetimeInSeconds>,
    "maxQueryResults": <maximumQueryResults>,
    "apiQueryAccess": "Disabled" | "EnabledAllUsers" | "EnabledSystemUsers"
  },
  "encryption": {
    "enabled": true|false,
    "keyReuseEnabled": true|false,
    "keyRetryTimeoutSeconds": <retryKeyretrivalEveryNumberOfSeconds>,
    "maximumKeyLifeBeforeShutdownDays": <maximumKeyLifeInDays>,
    "maximumMasterKeyLifeBeforeShutdownDays": <maximumMasterKeyLifeInDays>,
    "maskEmailsEnabled": true|false,
    "wipeDeletedConfigurationAndUserFiles": true|false,
    "changeMasterKeyRelativeWaitFactor": <waitFactor>,
    "encryptSensitiveExports": true|false
  },
  "eventLogMessages": {
    "defaultEventLogMessageViewInterval": <defaultViewPeriodInSeconds>,
    "maxRecEventLogMessage": <maximumRecordsShown>,
    "enableAllLogMessages": true|false,
    "eventLogMessagesEnabled": true|false,
    "syslogTemplate": "{instanceId} {dateIso}.{nanoSeconds} {level} {number} {user}
{message} {ipData}"
  },
  "ticker": {
    "enabled": true|false,
    "maxTickerEntries": <maximumNumberOfSavedTickerEntries>
  },
  "other": {
    "eodDayTimeString": "<startEndOfDayJobAt [hh:mm] ",
    "shutdownGracePeriod": <shutdownGracePeriodInSeconds>,
    "updateInstanceStatusFileSec": <updateIntervalInSecondsForInstanceStatusFile>,
    "maxReportLimit": <reportLimitAt>,
    "checkIndexesOnStartup": true|false,
    "fileCreateZeros": "AlwaysFill" | "NoFill" | "FillInitialOnly",
    "wipeDdcPosition": true|false,
    "computationThreadBufferElementSize": <computationThreadBufferElementSize>,
    "multipleRelationsIndexSearch": true|false,
    "sslCipherList": "<sslCipherList>",
    "sslUseCertificateChain": true|false
  }
}
}
}
}

```

- "data:settings" (mandatory): Contains the settings dictionary
- "data:settings:applications" (optional): Contains general application settings

- "data:settings:applications:name" (optional): Sets the application name which is e.g. displayed as browser tab name
- "data:settings:applications:serverTimeZone" (optional): Sets the server time zone in UTC +/- seconds
- "data:settings:memory" (optional): Contains memory settings
- "data:settings:memory:limit" (optional): Sets a maximum memory border (in GB) a SP instance may use
- "data:settings:userAccounts" (optional): Contains settings for user accounts, including password policies
- "data:settings:userAccounts:maxFailedLogin" (optional): A user account is blocked after these many failed login attempts
- "data:settings:userAccounts:numberOldPasswordRejected" (optional): Length of password history which is rejected as new password
- "data:settings:userAccounts:passwordValiditySeconds" (optional): Time period in seconds that a password is valid until it must be changed
- "data:settings:userAccounts:minPasswordLength" (optional): Minimum password length
- "data:settings:userAccounts:passwordMustContainLowerCase" (optional): Enforce using lower case characters in passwords
- "data:settings:userAccounts:passwordMustContainUpperCase" (optional): Enforce using upper case characters in passwords
- "data:settings:userAccounts:passwordMustContainDigit" (optional): Enforce using digits in passwords
- "data:settings:userAccounts:passwordMustContainSpecialCharacter" (optional): Enforce using special characters in passwords
- "data:settings:userAccounts:userAccountDeletion" (optional): Enable functionality to permanently delete user accounts
- "data:settings:userAccounts:enableExtendedAuth" (optional): Enable two factor authentication
- "data:settings:userAccounts:enableSystemUser" (optional): Enables definition of system accounts that may be used by API scripts
- "data:settings:userAccounts:maximumUserInactivityPeriodDays" (optional): Users are disabled automatically when not logged in for this amount of dates
- "data:settings:userAccounts:restrictSystemUserIP" (optional): If system users are enabled, this is a list of IPs from which they may connect
- "data:settings:ldap" (optional): Allows to configure ldap.
- "data:settings:ldap:login" (optional): Choose whether to use normal SP login, ldap or both
- "data:settings:ldap:useActiveDirectory" (optional): If enabled, active directory authentication will be used
- "data:settings:ldap:host" (optional): IP address or domain name of LDAP server
- "data:settings:ldap:port" (optional): IP port of LDAP server
- "data:settings:ldap:baseDn" (optional): Base of distinguished name
- "data:settings:ldap:domain" (optional): Active Directory domain for login

- "data:settings:ldap:ssl" (optional): Enables encrypted communication between IBM Safer Payments and LDAP host
- "data:settings:ldap:timeout" (optional): Time to wait to connect to LDAP server before timeout
- "data:settings:ldap:sso" (optional): If enabled, users may automatically login using their windows credentials
- "data:settings:ldap:ssoAutoRelogin" (optional): Allow a user to automatically sign in after their session has expired
- "data:settings:priorisation" (optional): Contains settings for thread priorisations to set priorities for CPU resources
- "data:settings:priorisation:enabled" (optional): Enables priorisations between threads
- "data:settings:priorisation:irisProcess" (optional): Sets priority of the SP process in general
- "data:settings:priorisation:threads:interfaces" (optional): Contains priority settings for the several interfaces
- "data:settings:priorisation:threads:interfaces:messageCommand" (optional): Sets priority of MCI
- "data:settings:priorisation:threads:interfaces:statusControl" (optional): Sets priority of SCI
- "data:settings:priorisation:threads:interfaces:applicationProgramming" (optional): Sets priority of API
- "data:settings:priorisation:threads:interfaces:batchData" (optional): Sets priority of BDI
- "data:settings:priorisation:threads:interfaces:fastLink" (optional): Sets priority of FLI
- "data:settings:priorisation:threads:interfaces:alertMessage" (optional): Sets priority of AMI
- "data:settings:priorisation:threads:caseManagement" (optional): Sets priority of activities in case management
- "data:settings:priorisation:threads:deferredWriter" (optional): Sets priority of deferred writing thread
- "data:settings:priorisation:threads:analysis" (optional): Sets priority of analysis computations
- "data:settings:priorisation:threads:elementGeneration" (optional): Sets priority of element creations
- "data:settings:priorisation:threads:statusAlarmIndicators" (optional): Sets priority of status alarm indicators
- "data:settings:priorisation:threads:keyPerformanceIndicators" (optional): Sets priority of key performance indicators
- "data:settings:priorisation:threads:endOfDay" (optional): Sets priority of end of day job
- "data:settings:priorisation:threads:golive" (optional): Sets priority of golives
- "data:settings:priorisation:threads:simulation" (optional): Sets priority of simulation computation
- "data:settings:priorisation:threads:query" (optional): Sets priority of query execution
- "data:settings:priorisation:threads:reminder" (optional): Sets priority of reminder computation
- "data:settings:priorisation:threads:instanceStatusFileService" (optional): Sets priority of instance status file write thread
- "data:settings:priorisation:threads:definedRiskListEntry" (optional): Sets priority of defined risk list entry generation
- "data:settings:deferredWriting" (optional): Contains settings for deferred writing

- "data:settings:deferredWriting:enabled" (optional): Enables deferred writing
- "data:settings:deferredWriting:resetChunkingForShutdownEnabled" (optional): If enabled, resets all deferred writing chunking parameters during shutdown to achieve maximum writing speed
- "data:settings:deferredWriting:timeGap" (optional): Time in seconds that MDC will never be written to DDC
- "data:settings:deferredWriting:minimumMdcSize" (optional): Defined minimum MDC size that Safer Payments uses for sanity checks during golives
- "data:settings:deferredWriting:safetyMargin" (optional): Number of records that within the MDC are not used for caching but rather to account for the new transactions to be stored in MDC during commitment of MDC data to DDC
- "data:settings:deferredWriting:pauseAfterElement" (optional): Time in milliseconds to wait after each element is written to disk
- "data:settings:deferredWriting:diskChunkDelay" (optional): Time in milliseconds to wait before writing the next chunk
- "data:settings:deferredWriting:diskChunkSize" (optional): Bytesize of data chunk which is written to disk before Safer Payments waits the defined time
- "data:settings:deferredWriting:shadowCommitChunkSize" (optional): If greater than 0, transaction computation will get smoother as between chunks, computation continues
- "data:settings:deferredWriting:shadowCommitChunkDelay" (optional): Time in milliseconds to wait in between chunks for transaction computation
- "data:settings:numberOfThreads" (optional): Defines how many threads different SP computations may acquire
- "data:settings:numberOfThreads:messageCommandInterface" (optional): Number of threads for MCI
- "data:settings:numberOfThreads:batchDataInterface" (optional): Number of threads for BDI
- "data:settings:numberOfThreads:fastLinkInterface" (optional): Number of threads for FLI
- "data:settings:numberOfThreads:analysis" (optional): Number of threads analysis services computations spin off in each computational step
- "data:settings:numberOfThreads:elementGeneration" (optional): Number of threads model element generation spins off in each computational step
- "data:settings:numberOfThreads:simulation" (optional): Number of threads simulation spins off in each simulation run
- "data:settings:numberOfThreads:query" (optional): Number of threads queries spin off
- "data:settings:numberOfThreads:file" (optional): Number of file notification threads defines maximum number of file notifications that may be processed in parallel
- "data:settings:numberOfThreads:message" (optional): Number of message notification threads defines maximum number of message notifications that may be processed in parallel
- "data:settings:numberOfThreads:http" (optional): Number of http notification threads defines maximum number of http notifications that may be processed in parallel
- "data:settings:numberOfThreads:odbcSql" (optional): Number of odbcSql notification threads defines maximum number of odbcSql notifications that may be processed in parallel

- "data:settings:numberOfThreads:smtp" (optional): Number of smtp notification threads defines maximum number of smtp notifications that may be processed in parallel
- "data:settings:numberOfThreads:templateFile" (optional): Number of templateFile notification threads defines maximum number of docx notifications that may be processed in parallel
- "data:settings:pageAutoRefresh" (optional): Allows to define the intervals (in seconds) in which different pages are automatically refreshed
- "data:settings:pageAutoRefresh:dashboard" (optional): Automatically refresh the dashboard page after x seconds
- "data:settings:pageAutoRefresh:investigation" (optional): Automatically refresh cases table after x seconds
- "data:settings:pageAutoRefresh:workCase" (optional): Automatically refresh work case page after x seconds
- "data:settings:pageAutoRefresh:analysis" (optional): Automatically refresh analysis results during computation after x seconds
- "data:settings:pageAutoRefresh:jobSchedule" (optional): Automatically refresh job schedule page after x seconds
- "data:settings:pageAutoRefresh:elementGeneration" (optional): Automatically refresh a model element generation page after x seconds
- "data:settings:pageAutoRefresh:golive" (optional): Automatically refresh model revision selection page during a golive after x seconds
- "data:settings:pageAutoRefresh:cluster" (optional): Automatically refresh cluster page after x seconds
- "data:settings:pageAutoRefresh:keys" (optional): Automatically refresh encryption keys page after x seconds
- "data:settings:pageAutoRefresh:query" (optional): Automatically refresh query results during computation after x seconds
- "data:settings:pageAutoRefresh:memoryManagement" (optional): Automatically refresh memory management page after x seconds
- "data:settings:pageAutoRefresh:sessionCountdown" (optional): Auto-refresh period in seconds for synchronization of session countdowns in different tabs
- "data:settings:investigation" (optional): Configures handling of investigation cases
- "data:settings:investigation:caseGenerationEnabled" (optional): Enables to work with cases in Safer Payments
- "data:settings:investigation:manualFraudValue" (optional): If no categories are defined for meta attribute "fraud", this value is assigned to records that are manually marked fraudulent
- "data:settings:investigation:updateProfilesAndEventsByManualFraudMark" (optional): If enabled, a manual fraud marking will cause an update of profile and event values
- "data:settings:investigation:maxCasesSelectionTable" (optional): Case selection table will not show more entries than defined here
- "data:settings:investigation:maxRecCaseHistory" (optional): Case history table will not show more entries than defined here
- "data:settings:investigation:caseConsolidationRepeat" (optional): Time period in seconds in which case consolidation is periodically started

- "data:settings:investigation:caseEscalationRepeat" (optional): Time period in seconds in which case escalation is periodically started
- "data:settings:investigation:caseDispatchingRepeat" (optional): Time period in seconds in which case dispatching is periodically started
- "data:settings:investigation:archiveCasesAfterSeconds" (optional): Time period in days after which cases are automatically archived
- "data:settings:investigation:enableCaseAttachments" (optional): If enabled, files may be uploaded to a case
- "data:settings:investigation:maxCaseAttachmentSizeMB" (optional): Defines a maximum size allowed for case attachments
- "data:settings:investigation:caseCreationDdc" (optional): If enabled, cases manually created from a query result may include data from DDC
- "data:settings:monitoring" (optional): Configures usage of defined risk lists and compliance lists
- "data:settings:monitoring:monitoringResultLifetime" (optional): Time in seconds for how long a monitoring search result is stored
- "data:settings:monitoring:monitoringMaxTableEntries" (optional): Maximum number of defined risk list entries that are shown in a table
- "data:settings:monitoring:monitoringComplianceEnabled" (optional): Enables monitoring of compliance lists
- "data:settings:monitoring:monitoringComplianceEuslEnabled" (optional): Enables usage of European sanction list
- "data:settings:monitoring:monitoringComplianceOfacEnabled" (optional): Enables usage of OFAC list
- "data:settings:monitoring:monitoringComplianceUNEnabled" (optional): Enables usage of United Nations list
- "data:settings:monitoring:monitoringComplianceGlobalWatchListEnabled" (optional): Enables usage of Global watch list
- "data:settings:monitoring:monitoringCompliancePepEnabled" (optional): Enables usage of politically exposed persons list (PEP)
- "data:settings:monitoring:monitoringComplianceRuslEnabled" (optional): Enables usage of Russian sanction list (PEP)
- "data:settings:monitoring:monitoringComplianceWaitFactor" (optional): The relative wait factor for a compliance list reload request to be transmitted to remote instances
- "data:settings:monitoring:definedRiskListDeleteEnabled" (optional): Allows to permanently delete defined risk lists
- "data:settings:monitoring:definedRiskListMassDeleteEnabled" (optional): Allows bulk deletion of defined risk list entries
- "data:settings:monitoring:definedRiskListMassDisableEnabled" (optional): Allows bulk enabling/disabling of defined risk list entries
- "data:settings:monitoring:definedRiskListEntryConsolidationRepeat" (optional): Time period in seconds in which defined risk list entry consolidation is periodically started

- "data:settings:model" (optional): Configures settings related to decision models "data:settings:model:enforcePeerConfirmation" (optional): If enabled, the user that has last edited a model revision may not be the one who confirms a golive
- "data:settings:model:rebuildIndexOnSizeChange" (optional): If active, indexes will be recomputed from scratch during golive, if their sequence settings have changed and they now can access more records
- "data:settings:model:enableDdcInModel" (optional): If enabled, counters and index sequencings may be defined to also use data from DDC
- "data:settings:model:enableDdcInMerchantMonitoringRules" (optional): If enabled, merchant monitoring rules may be defined to also use data from DDC
- "data:settings:model:enableDdcInMergings" (optional): If enabled, mergings may be defined to also evaluate data from DDC
- "data:settings:model:numberOfCategoriesInAuditTrail" (optional): The maximum number of changed categories which will be stored in detail in the revision audit trail or shown in the revision compare function
- "data:settings:model:numberOfStoredChampionAuditTrailEntries" (optional): The maximum number of audit trail entries which will be stored in the champion audit trail
- "data:settings:model:numberOfDisplayedChampionAuditTrailEntries" (optional): The maximum number of displayed audit trail entries out of the champion audit trail
- "data:settings:model:numberOfDisplayedCollusionSimulationResults" (optional): The maximum number of collusion simulation results that will be displayed
- "data:settings:model:goliveRelativeWaitFactor" (optional): The relative wait factor for a golive to be transmitted to remote instances; "0" for immediate transmission after successful golive; "1" for a cascaded golive
- "data:settings:model:treatGoliveWarningsAsErrors" (optional): If enabled, all warnings in golive report will be treated as errors and thus prevent the golive
- "data:settings:model:defaultModelling" (optional): Allows to configure settings for modelling
- "data:settings:model:defaultModelling:simulationMayUseDdc" (optional): If enabled, simulation may access data from DDC
- "data:settings:model:defaultModelling:categoryLimitModelling" (optional): Defines how many indicators shall be considered in computation
- "data:settings:model:defaultCapacities" (optional): Defines the default mdc/ddc settings that should be pre-chosen when creating a new element
- "data:settings:model:defaultCapacities:mdc" (optional): Default MDC capacity
- "data:settings:model:defaultCapacities:ddc" (optional): Default DDC capacity
- "data:settings:model:defaultCapacities:ddcStored" (optional): If enabled, new elements will have MDC enabled as default
- "data:settings:model:defaultCapacities:mdcStored" (optional): If enabled, new elements will have DDC enabled as default
- "data:settings:model:deletion" (optional): Defines which types of revisions may be deleted
- "data:settings:model:deletion:challenger" (optional): If enabled, challenger revisions may be deleted
- "data:settings:model:deletion:invalidated" (optional): If enabled, invalidated revisions may be deleted

- "data:settings:model:deletion:retired" (optional): If enabled, retired revisions may be deleted
- "data:settings:interfaces" (optional): Contains settings for the various interfaces of Safer Payments
- "data:settings:interfaces:serializeComputation" (optional): Contains settings for serializing computations
- "data:settings:interfaces:serializeComputation:enabled" (optional): If enabled, computation of critical areas will be slower; if not enabled, doublet detection and mergings will not work correctly for parallelly incoming transactions
- "data:settings:interfaces:serializeComputation:mergingProtectionEnabled" (optional): If enabled, records that still are in computation may not be overwritten by merging sources, retrying the merging source until the computation is complete
- "data:settings:interfaces:serializeComputation:maxMergingProtectionTries" (optional): Defines the maximum amount of retries for mergings with access protection
- "data:settings:interfaces:serializeComputation:mergingProtectionWaitMsec" (optional): Defines the time in milliseconds in between access protection retries
- "data:settings:interfaces:serializeComputation:mergingProtectionFailOnTimeout" (optional): Define whether to abort a transaction or continuing when maximum retries are exceeded
- "data:settings:interfaces:statusControl" (optional): Configures settings of Status Control Interface (SCI)
- "data:settings:interfaces:statusControl:receiveTimeoutSeconds" (optional): When this timeout expires, the receiving socket is closed and the status message is dumped
- "data:settings:interfaces:messageCommand" (optional):
- "data:settings:interfaces:messageCommand:mciFormat" (optional): Select which ISO format this interface shall support
- "data:settings:interfaces:messageCommand:mciLatencyViolationThreshold" (optional): Latency threshold which is used to define what is considered as a latency violation
- "data:settings:interfaces:messageCommand:mciLatencyViolationLoggingEnabled" (optional): If enabled, detailed timing of MCI transactions are shown in system internals for latency violations
- "data:settings:interfaces:messageCommand:useOnlyPrintableAscii" (optional): If enabled, MCI response will only contain printable ASCII characters, others are escaped with _
- "data:settings:interfaces:messageCommand:closeDuringGolive" (optional): If enabled, all MCI connections will close during golive
- "data:settings:interfaces:messageCommand:responseMetaInformation" (optional): If enabled, users can define a custom response message for MCI messages. Only available in association with a custom parser.
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondInstanceName" (optional): If enabled, a MCI response will contain the instance name of the processing instance
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondMtid" (optional): If enabled, a MCI response will contain the MessageTypeID that has processed the message
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondSystemTime" (optional): If enabled, a MCI response will contain the servers system time
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondUrid" (optional): If enabled, a MCI response will contain the URID that was assigned to an incoming message

- "data:settings:interfaces:messageCommand:responseMetaInformation:respondMessageId" (optional): If enabled, a MCI response will echo the MessageId value that was sent with the incoming message
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondModels" (optional): If enabled, a MCI response will contain the model revisions that were used to compute the incoming message
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondMergingSource" (optional): If enabled, a MCI response will respond whether the incoming message was used as a merging source
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondStatus" (optional): If enabled, a MCI response will include Safer Payments server status
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondLatency" (optional): If enabled, a MCI response will include processing latency of the incoming transaction
- "data:settings:interfaces:messageCommand:responseMetaInformation:respondErrorText" (optional): If enabled, a MCI response will contain textual descriptions next to error codes
- "data:settings:interfaces:messageCommand:responseMetaInformation:maxDocumentedRulesInResponse" (optional): If set to a non-zero value, a MCI response will contain rules that had been fired during processing of the incoming transaction
- "data:settings:interfaces:messageCommand:messageTracing" (optional): These settings allow for the tracing of XML message requests within Safer Payments to test the MCI
- "data:settings:interfaces:messageCommand:messageTracing:showIpData" (optional): Causes Safer Payments to echo all data exchanged between Safer Payments and connected systems
- "data:settings:interfaces:messageCommand:messageTracing:dumpMalformattedMessages" (optional): If value is not zero, Safer Payments dumps this maximum number of malformatted messages it receives in the "log" subdirectory as individual files
- "data:settings:interfaces:messageCommand:threadBufferSize" (optional): Configures sizes of MCI buffers
- "data:settings:interfaces:messageCommand:threadBufferSize:incomingBuffer" (optional): Bytesize of MCI incoming buffer
- "data:settings:interfaces:messageCommand:threadBufferSize:responseBuffer" (optional): Bytesize of MCI outgoing buffer
- "data:settings:interfaces:applicationProgramming" (optional):
- "data:settings:interfaces:applicationProgramming:lockTimeoutSeconds" (optional): User will automatically log out when there is no action within the defined amount of seconds
- "data:settings:interfaces:applicationProgramming:lockTimeoutCountdownThresholdSeconds" (optional)↔ : Timeout countdown in UI will be highlighted starting from this threshold
- "data:settings:interfaces:applicationProgramming:enableCsrf" (optional): Use a cookie-stored session variable to avoid session hijacking via CSRF
- "data:settings:interfaces:applicationProgramming:useDefaultHttpHeaders" (optional): Use the default and secure HTTP headers for API requests
- "data:settings:interfaces:applicationProgramming:useHstsHttpHeaders" (optional): If enabled, use standard secure set of HTTP headers
- "data:settings:interfaces:applicationProgramming:customHttpHeaders" (optional): If useDefaultHttpHeaders is disabled, you can provide your custom headers here

- "data:settings:interfaces:applicationProgramming:useGzip" (optional): Use HTTP gzip compression for API requests
- "data:settings:interfaces:applicationProgramming:useKeepAlive" (optional): Use HTTP 1.1 keep-alive for API requests
- "data:settings:interfaces:applicationProgramming:downloadRequests" (optional): Allows the user to download all his or her previous requests
- "data:settings:interfaces:applicationProgramming:maxPostRequestSizeMB" (optional): Maximum size of a HTTP POST request
- "data:settings:interfaces:applicationProgramming:defaultLanguage" (optional): The default language with which new users are created
- "data:settings:interfaces:applicationProgramming:defaultDateTimeFormat" (optional): The default date time format with which new users are created
- "data:settings:interfaces:applicationProgramming:defaultDecimal" (optional): The default decimal separator with which new users are created
- "data:settings:interfaces:applicationProgramming:defaultDigitGroup" (optional): The default digit group separator with which new users are created
- "data:settings:interfaces:applicationProgramming:defaultFieldSeparator" (optional): The default field separator with which new users are created
- "data:settings:interfaces:applicationProgramming:defaultExtendedSelectDialog" (optional): The default setting whether to enable extended select dialogs for newly created users
- "data:settings:interfaces:applicationProgramming:defaultSearchInSelections" (optional): The default setting whether to enable search functionality for newly created users
- "data:settings:interfaces:applicationProgramming:defaultShowExplanation" (optional): The default setting whether to show explanations for newly created users
- "data:settings:interfaces:applicationProgramming:defaultShowUids" (optional): The default setting whether to show UIDs for newly created users
- "data:settings:interfaces:applicationProgramming:colorSchemaFraudulentValues" (optional): These colors are used to display fraudulent values in charts
- "data:settings:interfaces:applicationProgramming:colorSchemaGenuineValues" (optional): These colors are used to display genuine values in charts
- "data:settings:interfaces:applicationProgramming:colorSchemaOtherValues" (optional): These colors are used to display other fraud values in charts
- "data:settings:interfaces:batchData" (optional):
- "data:settings:interfaces:batchData:jobRepeat" (optional): Time period in seconds in which the job scheduler is periodically started
- "data:settings:interfaces:batchData:prioMaxJobThread" (optional): Maximum possible priority of a job
- "data:settings:interfaces:batchData:jobEncryptionEnabled" (optional): If enabled, encrypted job files can be imported
- "data:settings:interfaces:fastLink" (optional):

- "data:settings:interfaces:fastLink:receiveTimeoutSeconds" (optional): IBM Safer Payments closes its receiving sockets and restarts its incoming interface if the timeout expires
- "data:settings:interfaces:fastLink:retryConnectionSeconds" (optional): Time in seconds after a sending IBM Safer Payments instance retries to connect to another instance that was considered unresponsive before
- "data:settings:interfaces:fastLink:numMessagesToleratedInSync" (optional): Number of queued FastLink messages to still be considered synchronized
- "data:settings:interfaces:fastLink:outgoingBufferSize" (optional): Size in GB that IBM Safer Payments allocates for the buffering of FastLink messages
- "data:settings:interfaces:fastLink:jobProcessingBrakeAt" (optional): If the largest filling degree of all outgoing FastLink interface buffers exceeds this percentage, all data loading jobs will be freezed until it is not exceeded anymore
- "data:settings:interfaces:fastLink:transactionMessageProcessingHaltAt" (optional): If the largest filling degree of all FastLink interface buffers exceeds this percentage, all MCI connections are closed until it is not exceeded anymore
- "data:settings:interfaces:alertMessage" (optional):
- "data:settings:interfaces:alertMessage:ipAddress" (optional): Address of the SMTP server/gateway the alert message interface shall use
- "data:settings:interfaces:alertMessage:ipPort" (optional): Port of the SMTP server/gateway the alert message interface shall use
- "data:settings:interfaces:alertMessage:receiveTimeoutSeconds" (optional): IBM Safer Payments closes its receiving sockets and restarts its incoming interface if the timeout expires
- "data:settings:interfaces:alertMessage:authentication" (optional): SMTP authentication method according to RFC 4954
- "data:settings:interfaces:alertMessage:username" (optional): SMTP user name to be used for login type S↔MTP authentication
- "data:settings:interfaces:alertMessage:password" (optional): SMTP user password to be used for login type SMTP authentication
- "data:settings:interfaces:alertMessage:from" (optional): This email address will be used as sender and reply-to address with all outgoing email/SMS messages
- "data:settings:interfaces:alertMessage:sendInterval" (optional): Time period in seconds in which IBM Safer Payments sends out email/SMS
- "data:settings:interfaces:alertMessage:retryInterval" (optional): Time period in seconds IBM Safer Payments waits before it resends messages that could not be delivered in a previous attempt
- "data:settings:interfaces:alertMessage:archive" (optional): If checked, successfully sent Email/SMS messages are archived on disk
- "data:settings:interfaces:alertMessage:useSsl" (optional): Use SSL for communication to the SMTP server
- "data:settings:interfaces:alertMessage:verificationCertificate" (optional): Load the certificates for one or more trusted certification authorities from this PEM file
- "data:settings:query" (optional): Contains settings for the different type of queries in IBM Safer Payments

- "data:settings:query:maxRecordsWarning" (optional): If a user defines a query with a maximum number of records larger than this value, a warning will be generated
- "data:settings:query:maxRecordsLimit" (optional): IBM Safer Payments will prohibit defining a query with a maximum number of records larger than this value
- "data:settings:query:groupingQueryAccountsLimit" (optional): Defines the maximum number of displayed accounts in a group by query result
- "data:settings:query:groupingQueryResultCategoriesLimit" (optional): Defines the maximum number of categories that will be evaluated in a group by query
- "data:settings:query:queryEnableDdc" (optional): If checked, users may define queries that also use data from DDC
- "data:settings:query:queryResultLivetime" (optional): Defines the time a query result is stored by IBM Safer Payments
- "data:settings:query:commonPointQueryResultLivetimeSeconds" (optional): Defines the time a common point query result is stored by IBM Safer Payments
- "data:settings:query:maxQueryResults" (optional): Maximum number of results that are being stored for a group by query
- "data:settings:query:apiQueryAccess" (optional): Restricts the access to single API query requests
- "data:settings:encryption" (optional): Contains settings for encryption and encryption keys
- "data:settings:encryption:enabled" (optional): Enables functions for key management and encryption of elements
- "data:settings:encryption:keyReuseEnabled" (optional): If enabled, a previous key that is now inactive can be reactivated in the future
- "data:settings:encryption:keyRetryTimeoutSeconds" (optional): Time interval in seconds in which IBM Safer Payments attempts to retrieve keys from other cluster instances
- "data:settings:encryption:maximumKeyLifeBeforeShutdownDays" (optional): After the current key was active for the specified number of days, IBM Safer Payments will automatically shut down
- "data:settings:encryption:maximumMasterKeyLifeBeforeShutdownDays" (optional): After the current master key was active for the specified number of days, IBM Safer Payments will automatically shut down
- "data:settings:encryption:maskEmailsEnabled" (optional): Mask values of encrypted elements in outgoing messages
- "data:settings:encryption:wipeDeletedConfigurationAndUserFiles" (optional): If enabled, all configuration and user files are permanently and securely deleted when the respective element stored in it is deleted; if disabled, the respective file is renamed to ".deleted" suffix but not physically removed
- "data:settings:encryption:changeMasterKeyRelativeWaitFactor" (optional): The relative wait factor for a master key to get transmitted to other IBM Safer Payments instances; "0" for immediate transmission after successful change, "1" for a cascaded master key change
- "data:settings:encryption:encryptSensitiveExports" (optional): If enabled, CSV exports that contain encrypted values will be wrapped in an AES encrypted zip file
- "data:settings:eventLogMessages" (optional): Contains various settings for handling of event log messages

- "data:settings:eventLogMessages:defaultEventLogMessageViewInterval" (optional): Default time period from now into the past for which the UI integrated event log viewer shows log events
- "data:settings:eventLogMessages:maxRecEventLogMessage" (optional): Maximum number of entries shown in an event log table
- "data:settings:eventLogMessages:enableAllLogMessages" (optional): If enabled, the individual configuration of event log messages is ignored and all available log messages are always printed
- "data:settings:eventLogMessages:eventLogMessagesEnabled" (optional): If enabled, messages will be sent to the operating system event log
- "data:settings:eventLogMessages:syslogTemplate" (optional): If log messages are sent to the operating system event log, a custom format template can be defined here
- "data:settings:ticker" (optional): Contains settings for the ticker that allows to share messages between users of IBM Safer Payments
- "data:settings:ticker:enabled" (optional): Enables the ticker functionality
- "data:settings:ticker:maxTickerEntries" (optional): Maximum number of saved ticker entries; if this number is exceeded, oldest entries will be deleted
- "data:settings:other" (optional): Contains miscellaneous settings which do not fit into other sections
- "data:settings:other:eodDayTimeString" (optional): The time when the daily end of day job to clean out unused data is started
- "data:settings:other:shutdownGracePeriod" (optional): Time period in seconds IBM Safer Payments waits for service threads to end after a shutdown was initiated; after this period, the threads are forced to end
- "data:settings:other:updateInstanceStatusFileSec" (optional): Time interval after which the instance status file is updated; 0 disables the instance status file
- "data:settings:other:maxReportLimit" (optional): If a user defines a report with a maximum number of records larger than this value, IBM Safer Payments will automatically limit the result
- "data:settings:other:checkIndexesOnStartup" (optional): If enabled, IBM Safer Payments will check structural consistency of indexes during startup
- "data:settings:other:fileCreateZeros" (optional): Define when IBM Safer Payments enforces to fill the end of files with zeros on resize
- "data:settings:other:wipeDdcPosition" (optional): If enabled, the DDC write position of an encrypted attribute will be wiped before writing a new value
- "data:settings:other:computationThreadBufferElementSize" (optional): Allocated size (in bytes) of a buffer element in each computation thread
- "data:settings:other:multipleRelationsIndexSearch" (optional): If enabled, it is possible to automatically search for potential indexes when querying multiple relations masterdata
- "data:settings:other:sslCipherList" (optional): Defines which ciphers are allowed for all SSL-encrypted interfaces, similarly to the Apache setting "SSLCipherSuite"
- "data:settings:other:sslUseCertificateChain" (optional): Enables whether SSL certificates are verified against certificate chains if given

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

362 SendCaseActionFromPreview

Detailed Description

SendCaseActionFromPreview sends a case action from the preview screen. Starting from the original message template which is saved in the case action, this screen allows the user to create a customized adaptation. It is supported for SMTP and ODBC SQL case action types. The content which is sent may include a handpicked transaction data table.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "sendCaseActionFromPreview",
  "uid": <caseUid>,
  "mandator": <mandatorUid>,
  "data": {
    "toAddress": "<targetEmailAddress>",
    "caseActionUid": <caseActionUid>,
    "subjectTemplate": "<subjectTemplate>",
    "textModuleUid": <textModuleUid>,
    "bodyTemplate": "<bodyTemplate>",
    "bodyTemplateHtml": "<bodyTemplateHtml>",
    "comment": "<comment>",
    "table": <transactionTableData>
  }
}
```

- "uid" (mandatory): The uid of the case to fill case action data from
- "mandator" (mandatory): The mandator to which the case belongs
- "data:toAddress": The address to which the SMTP message shall be delivered (SMTP only)
- "data:caseActionUid": The uid of the case action which is to be sent (SMTP or SQL)
- "data:subjectTemplate": The text which is used for the subject of an SMTP message (SMTP only)
- "data:textModuleUid" (optional): Use a text module
- "data:bodyTemplate" (mandatory): The content of the email or SQL statement
- "data:bodyTemplateHtml" (optional): Content which supports HTML formatted text (SMTP only)
- "comment" (optional): A comment which is added to the case audit trail
- "table" (optional): Transaction data may be added in form of a transaction table
- "table:header" (optional): Transaction data table header
- "table:data" (optional): Transaction data table content

All templates may include placeholders for {reporting attributes}, {{transaction data attribute values}}, [[masterdata values]], [CaseVariables], [TextModules]

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

363 SendCaseActions

Detailed Description

SendCaseActions sends multiple case actions for a given case.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "sendCaseActions",
  "uid": <caseUid>,
  "mandator": <mandatorUid>,
  "data": {
    "caseActions": [<caseActionUid>],
    "comment": "<comment>"
  }
}
```

- "uid" (mandatory): The uid of the case which is supposed to be sent via case action
- "mandator" (mandatory): The uid of the mandator to which the case belongs
- "data:caseActions" (mandatory): A list of case action uids which shall be sent
- "data:comment" (optional): A comment which is added to the case audit trail.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

364 Serialize

Detailed Description

Serialize writes an element to disk. The requesting User must have the privilege to rewrite elements to disk via the corresponding maintenance function.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "serialize",
    "uid": <elementUid>
}
```

- "uid" (mandatory): The uid of the element to write to disk

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

365 SetAllMyPreferencesAsDefault

Detailed Description

SetAllMyPreferencesAsDefault takes all the requesting user's preferences and stores them as the default preferences. After this is done, resetting the preferences of any user will set them to preferences stored by the call of setAllMyPreferencesDefault.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "setAllMyPreferencesAsDefault"
}
```

- "parameterName" (mandatory/optional): Description

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

366 SetAnalysis

Detailed Description

SetAnalysis adds or updates an analysis definition. Settings include name, data selection and optimization.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setAnalysis",
  "uid": -1|<analysisId>,
  "revision": <revisionUid>,
  "data": {
    "enabled": true|false,
    "name": "<analysisName>",
    "comment": "<comment>",
    "optimization": true|false,
    "adoptSimulationDataSelection": true|false,
    "dataSelection": {
      "mandators": [<mandatorUid>],
      "periodType": "TimeAbsolute" | "RecordsAbsolute",
      "desired": {
        "fromUrid": <fromUrid>,
        "fromTimestamp": <fromTimestamp>,
        "toUrid": <toUrid>,
        "toTimestamp": <toTimestamp>,
      },
      "conditions": [...]
    },
    "execute": true|false
  }
}
```

- "uid" (mandatory): Use -1 for new analysis or the ID of an existing analysis. The "analysisId" can be obtained from the request of "getAnalysisTable".
- "revision" (mandatory): Revision uid to identify the revision to which the analysis belongs.
- "data:enabled" (mandatory): Enable or disable the analysis.
- "data:name" (mandatory): Name for the analysis.
- "data:comment" (optional): Provide a description or additional info for this analysis.
- "data:optimization" (optional): Enable or disable analysis optimization.
- "data:adoptSimulationDataSelection" (optional): Take data selection from simulation, otherwise use custom values as follows.
- "data:dataSelection" (optional): Provide a custom data selection if "adoptSimulationDataSelection" is not enabled.

- "data:dataSelection:mandators" (optional): Analyze transactions from these mandators.
- "data:dataSelection:periodType" (optional): Provide data selection in form of time or records.
- "data:dataSelection:desired" (optional): Contains the from-to values for data selection. If periodType is TimeAbsolute, the timestamp keys must be filled. If it is RecordsAbsolute, the urid keys must be filled.
- "data:dataSelection:desired:fromUrid" (optional): Analyze transactions starting from this URID.
- "data:dataSelection:desired:fromTimestamp" (optional): Analyze transactions that are younger than this time.
- "data:dataSelection:desired:toUrid" (optional): Analyze transactions up to this URID.
- "data:dataSelection:desired:toTimestamp" (optional): Analyze transactions that are older than this time.
- "data:dataSelection:conditions" (optional): Analyze only transactions which meet these conditions.
- "data:execute" (optional): Execute the analysis immediately after saving.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

367 SetAttachmentData

Detailed Description

SetAttachmentData sets data of a case attachment with a comment of additional information. The comment will be displayed in the attachments table of case investigation.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "setAttachmentData",
    "uid": <caseUid>,
    "data": {
        "id": <attachmentId>,
        "comment": "<comment>"
    }
}
```

- "uid" (mandatory): Uid to identify the case to which the attachment belongs
- "data:id" (mandatory): Attachment id to identify the right attachment
- "data:comment" (optional): Information to be saved to the attachment

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

368 SetFinalRulesetsEnabled

Detailed Description

SetFinalRulesetsEnabled enables/disables final rulesets.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setFinalRulesetsEnabled",
  "revision": <revisionUid>,
  "data": {
    "enabled": true|false,
    "rulesets": [<rulesetUid>]
  }
}
```

- "revision": The revision to which final rulesets belong.
- "data:enabled": Flag indicating enabled/disabled operation.
- "data:rulesets": The final rulesets that should be enabled/disabled.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


369 SetKey

Detailed Description

SetKey sets the left or right key of a public encryption Key. The requesting User must have the privilege to enter left or right keys.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setKey",
  "uid": <keyUid>,
  "data": {
    "streamType": "key",
    "uid": <keyUid>,
    "public": {
      "right|"left": {
        "passphrase": "<passphrase>",
        "enteredComment": "<comment>"
      }
    }
  }
}
```

- "uid" (mandatory): The public encryption Key for which to set the left or right key.
- "data" (mandatory): Contains the entered passphrase.
- "data:uid" (mandatory): The public encryption Key again.
- "data:public" (mandatory): Contains the left or right passphrase.
- "data:right|left" (mandatory): Usage depends on which key was entered.
- "data:right|left:passphrase": The entered passphrase.
- "data:right|left:enteredComment": The entered comment.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

370 SetMasterdata

Detailed Description

SetMasterdata manually sets a masterdata value of a chosen index entry.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setMasterdata",
  "uid": <masterdataUid>,
  "mandator": <mandatorUid>,
  "data": {
    "parameter": "<parameter>",
    "value": "<value>"
  }
}
```

- "uid" (mandatory): The uid of the masterdata which is to be assigned the new value
- "mandator" (mandatory): The uid of the mandator in which the masterdata is defined
- "data:parameter" (mandatory): The index entry which is to be assigned with the new masterdata value
- "data:value" (mandatory): The value which is to be assigned to the given index entry

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

371 SetModelComponentsEnabled

Detailed Description

SetModelComponentsEnabled enables/disables model components.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "setModelComponentsEnabled",
    "revision": <revisionUid>,
    "data": {
        "enabled": true|false,
        "rulesets": [<rulesetUid>]
    }
}
```

- "revision": The revision to which model components belong.
- "data:"
- "data:enabled": Flag indicating enabled/disabled operation.
- "data:modelComponents": The model components that should be enabled/disabled.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": [<status>, "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

372 SetModelling

Detailed Description

SetModelling enable/disable an attribute for the different modelling sections: sandbox, simulation, analysis, rule generation.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setModelling",
  "revision": <revisionUid>,
  "data": {
    "attribute": <attributeUid>,
    "testEnabled": true|false,
    "simulationEnabled": true|false,
    "analysisEnabled": true|false,
    "ruleGenerationEnabled": true|false
  }
}
```

- "revision" (mandatory): The revision in which the modelling setting shall be changed
- "data:attribute" (mandatory): The uid to identify the changed attribute
- "data:testEnabled" (optional): Enables the attribute for sandbox computations
- "data:simulationEnabled" (optional): Enables the attribute for simulation
- "data:analysisEnabled" (optional): Enables the attribute for analyses
- "data:ruleGenerationEnabled" (optional): Enables the attribute for rule generation

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

373 SetMultiInterfaces

Detailed Description

SetMultiInterfaces enables/disables interfaces of multiple instances of IBM Safer Payments cluster. The requesting User must have the privilege to change the IBM Safer Payments cluster.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "setMultiInterfaces",
    "data": {
        "interfaceType": "<interfaceType>",
        "instances": [<instanceId>],
        "enable": true|false
    }
}
```

- "data" (mandatory): Values to identify the interface to enable/disable and the instances on which to perform the action.
- "data:interfaceType" (mandatory): Type of the interface to enable/disable. One of ["MessageCommandInterface", "ApplicationProgrammingInterface", "BatchDataInterface", "FastLinkInterface", "StatusControlInterface", "EncryptedCommunicationInterface", "AlertMessageInterface", "WebSphereMQInterface"].
- "data:instances" (mandatory): A list of IrisInstance objects (identified by their uids) to change interface settings of.
- "data:enable" (mandatory): Indicates if interface should be enabled or disabled.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": [<status>, "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

374 SetMultiModelling

Detailed Description

SetMultiModelling enables/disables a list of multiple attributes for one of sandbox/simulation/analysis/rule generation.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setMultiModelling",
  "revision": <revisionUid>,
  "data": {
    "modellingType": "Test" | "Simulation" | "Analysis" | "RuleGeneration",
    "enable": true|false,
    "attributes": [<attributeUid>]
  }
}
```

- "revision" (mandatory): The uid of the revision in which modelling settings shall be changed
- "data:modellingType" (mandatory): The modelling setting that shall be changed for all the given attributes
- "data:enable" (mandatory): Enable or disable the given modelling setting for all given attributes
- "data:attributes" (mandatory): A list of attributes for which the modelling setting shall be changed

Returns

A HTTP response with the following content:

```
{
  "responseStatus": [<status>, "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

375 SetOffline

Detailed Description

SetOffline closes API, BDI, incoming FLI and MCI of an instance in cluster. The requesting User must have the privilege to change the IBM Safer Payments cluster.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "setOffline",
    "uid": <instanceUid>
}
```

- "uid": The uid of the instance to take offline.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbackText>"],
    "reloadUserProfile": true|false
}
```

376 SetOnline

Detailed Description

SetOnline opens API, BDI, incoming FLI and MCI, if enabled, of one instance in the cluster. The requesting User must have the privilege to change the IBM Safer Payments cluster.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "setOnline",
    "uid": <instanceUid>
}
```

- "uid": The uid of the instance to go online.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbackText>"],
    "reloadUserProfile": true|false
}
```


377 SetPreference

Detailed Description

SetPreference Sets a single preference of the requesting User. Preferences include the order and width of table columns, the number of displayed rows, the size of help dialogs and more.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "setPreference",
    "type": <type>,
    "data": {
        ... ..
    }
}
```

- "type" (mandatory): Controls which preferences should be set.
- "data" (mandatory): Depending on "type" contains the new preferences.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbackText>"],
    "reloadUserProfile": true|false
}
```

378 SetQuickSearchCasesTablePreference

Detailed Description

SetQuickSearchCasesTablePreference sets the search parameters for cases. This request determines the response of another request getQuickSearchCasesTable.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setQuickSearchCasesTablePreference",
  "data": {
    "caseSearch": {
      "attribute": <attributeUid>,
      "value": "<value>"
    }
  }
}
```

- "data:caseSearch:attribute" (mandatory): Search cases by this attribute
- "data:caseSearch:value" (mandatory): Search cases with this value for the given attribute

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

379 SetRuleGenerationCategoricIndicators

Detailed Description

SetRuleGenerationCategoricIndicators sets/unsets indicators of categoric attributes in rule generation. In rule generation, indicators are attribute values or value ranges that are evaluated to indicate fraud or not. They will eventually be molded into conditions of the generated rules. Attributes have different "usage types" to reflect different groupings of values (e.g. boolean attributes are "categoric" as they allow a set of predefined values, numeric attributes with decimals are "logarithmic" and numeric attributes without decimals are "interval"). For attributes of usage type "categoric", the setRuleGenerationCategoricIndicators request may be used to manually select indicator values which may later be put into a condition.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setRuleGenerationCategoricIndicators",
  "uid": <attributeUid>,
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "value": true|false,
    "indicators": [<indicatorId>]
  }
}
```

- "uid" (mandatory): Uid of the attribute whose indicators shall be set
- "mandator" (mandatory): Uid of the mandator in which the rule generation is taking place
- "revision" (mandatory): Uid of the revision in which rule generation is taking place
- "data:value" (mandatory): Decides whether to enable or disable the indicators
- "data:indicators" (mandatory): A list of indicators that shall be enabled or disabled

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

380 SetRuleGenerationDataSelection

Detailed Description

SetRuleGenerationDataSelection sets the data selection to be evaluated for a rule generation computation. It contains two data selection streams: the general training data selection defines which transactions shall be considered when generating rules, and the verification data selection defines against which transactions the generated rules shall be verified in order to test efficiency.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setRuleGenerationDataSelection",
  "revision": <revisionUid>,
  "data": {
    "verify": true|false,
    "generationScenario": "FromScratch" | "CurrentRevisionOnly" |
"CurrentRevisionAndHistory",
    "trainingDataSelectionStream": {
      "mandators": [<mandatorUid>],
      "periodType": "RecordsAbsolute" | "TimeAbsolute",
      "desired": {
        "fromUrid": <fromUrid>,
        "fromTimestamp": <fromTimestamp>,
        "toUrid": <toUrid>,
        "toTimestamp": <toTimestamp>,
      },
      "conditions": [ <conditions> ]
    },
    "verificationDataSelectionStream": {
      "mandators": [<mandatorUid>],
      "periodType": "RecordsAbsolute" | "TimeAbsolute",
      "desired": {
        "fromUrid": <fromUrid>,
        "fromTimestamp": <fromTimestamp>,
        "toUrid": <toUrid>,
        "toTimestamp": <toTimestamp>,
      },
      "conditions": [ <conditions> ]
    }
  }
}
```

- "revision" (mandatory): Identifies the revision in which the rule generation data selection is to be set
- "data:verify" (optional): Enables/disables verification of data selection
- "data:generationScenario" (optional):
- "data:trainingDataSelectionStream" (mandatory): Defines which data is used for rule generation

- "data:trainingDataSelectionStream:mandators" (optional): Include transactions from these mandators
- "data:trainingDataSelectionStream:periodType" (optional): Choose whether to provide data selection in form of time or records
- "data:trainingDataSelectionStream:desired" (optional): Contains the from-to values for data selection. If periodType is TimeAbsolute, the timestamp keys must be filled. If it is RecordsAbsolute, the urid keys must be filled.
- "data:trainingDataSelectionStream:desired:fromUrid" (optional): Include transactions starting from this URID
- "data:trainingDataSelectionStream:desired:fromTimestamp" (optional): Include transactions that are younger than this time
- "data:trainingDataSelectionStream:desired:toUrid" (optional): Include transactions up to this URID
- "data:trainingDataSelectionStream:desired:toTimestamp" (optional): Include transactions that are older than this time
- "data:trainingDataSelectionStream:conditions" (optional): Include only transactions which meet these conditions
- "data:verificationDataSelectionStream" (optional): Defines which data is used for verification of rule generation
- "data:verificationDataSelectionStream:mandators" (optional): Verify the generated rules against transactions from these mandators
- "data:verificationDataSelectionStream:periodType" (optional): Choose whether to provide data selection in form of time or records
- "data:verificationDataSelectionStream:desired" (optional): Contains the from-to values for data selection. If periodType is TimeAbsolute, the timestamp keys must be filled. If it is RecordsAbsolute, the urid keys must be filled.
- "data:verificationDataSelectionStream:desired:fromUrid" (optional): Verify the generated rules against transactions starting from this URID
- "data:verificationDataSelectionStream:desired:fromTimestamp" (optional): Verify the generated rules against transactions that are younger than this time
- "data:verificationDataSelectionStream:desired:toUrid" (optional): Verify the generated rules against transactions up to this URID
- "data:verificationDataSelectionStream:desired:toTimestamp" (optional): Verify the generated rules against transactions that are older than this time
- "data:verificationDataSelectionStream:conditions" (optional): Verify the generated rules only against transactions which meet these conditions

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

381 SetRuleGenerationFollowingIndicators

Detailed Description

SetRuleGenerationFollowingIndicators sets/unsets a range of indicators of logarithmic/interval attributes in rule generation. In rule generation, indicators are attribute values or value ranges that are evaluated to indicate fraud or not. They will eventually be molded into conditions of the generated rules. Attributes have different "usage types" to reflect different groupings of values (e.g. boolean attributes are "categoric" as they allow a set of predefined values, numeric attributes with decimals are "logarithmic" and numeric attributes without decimals are "interval"). For attributes of usage type "logarithmic" or "interval", the setRuleGenerationCategoricIndicators request may be used to manually select a range of indicator values which may later be put into a condition.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setRuleGenerationFollowingIndicators",
  "uid": <attributeUid>,
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "value": true|false,
    "fromIndicator": <fromIndicatorId>,
    "toIndicator": <toIndicatorId>
  }
}
```

- "uid" (mandatory): Uid of the attribute whose indicators shall be set
- "mandator" (mandatory): Uid of the mandator in which the rule generation is taking place
- "revision" (mandatory): Uid of the revision in which rule generation is taking place
- "data:value" (mandatory): Decides whether to enable or disable the indicators
- "data:fromIndicator" (mandatory): Start from this indicator, value is given as row number in indicators table. Must be smaller than toIndicator
- "data:toIndicator" (mandatory) End with this indicator, value is given as row number in indicators table. Must be greater than fromIndicator

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

382 SetRuleGenerationIndicator

Detailed Description

SetRuleGenerationIndicator manually select indicator values which may later be put into a condition. In rule generation, indicators are attribute values or value ranges that are evaluated to indicate fraud or not. They will eventually be molded into conditions of the generated rules.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setRuleGenerationIndicator",
  "uid": <attributeUid>,
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "indicator": <indicatorId>,
    "value": true|false
  }
}
```

- "uid" (mandatory): Uid of the attribute whose indicator shall be set
- "mandator" (mandatory): Uid of the mandator in which the rule generation is taking place
- "revision" (mandatory): Uid of the revision in which rule generation is taking place
- "data:indicator" (mandatory): Identifies the indicator by row number in the indicator table
- "data:value" (mandatory): Decides whether to enable or disable the indicator

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

383 SetRuleGenerationPrecedingIndicators

Detailed Description

SetRuleGenerationPrecedingIndicators sets/unsets a range of indicators of logarithmic/interval attributes in rule generation. In rule generation, indicators are attribute values or value ranges that are evaluated to indicate fraud or not. They will eventually be molded into conditions of the generated rules. Attributes have different "usage types" to reflect different groupings of values (e.g. boolean attributes are "categoric" as they allow a set of predefined values, numeric attributes with decimals are "logarithmic" and numeric attributes without decimals are "interval"). For attributes of usage type "logarithmic" or "interval", the setRuleGenerationCategoricIndicators request may be used to manually select a range of indicator values which may later be put into a condition. In opposition to setRuleGenerationFollowingIndicators, this request goes backwards, i.e. the ending value for identifying indicators must be smaller than the starting value.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setRuleGenerationFollowingIndicators",
  "uid": <attributeUid>,
  "mandator": <mandatorUid>,
  "revision": <revisionUid>,
  "data": {
    "value": true|false,
    "fromIndicator": <fromIndicatorId>,
    "toIndicator": <toIndicatorId>
  }
}
```

- "uid" (mandatory): Uid of the attribute whose indicators shall be set
- "mandator" (mandatory): Uid of the mandator in which the rule generation is taking place
- "revision" (mandatory): Uid of the revision in which rule generation is taking place
- "data:value" (mandatory): Decides whether to enable or disable the indicators
- "data:fromIndicator" (mandatory): Start from this indicator, value is given as row number in indicators table. Must be greater than toIndicator
- "data:toIndicator" (mandatory): End with this indicator, value is given as row number in indicators table. Must be smaller than fromIndicator

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```


384 SetRuleGenerationSettings

Detailed Description

SetRuleGenerationSettings defines the settings according to which rules shall be generated. It allows to sketch up a template for rule names and set maximum/minimum values a rule must be fulfill in order to be suggested (e.g. maximum hit rate, minimum condition improvement).

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setRuleGenerationSettings",
  "revision": <revisionUid>,
  "data": {
    "balanceFraudFalsePositives": <weightingSetting>,
    "lowerBoundFalsePositives": <minFalsePositives>,
    "upperBoundFraudHitRate": <upperBoundHitRate>,
    "stopRuleFalsePositives": <stopFalsePositives>,
    "maxNumberConditions": <maxNumberOfConditions>,
    "minPopulationQuantise": <minPopulationQuantise>,
    "minSampleSizePercent": <minSampleSizeInPercent>,
    "minSelectHitRate": <minSelectHitRate>,
    "minConditionImprovement": <minConditionImprovement>,
    "relaxAllUpAutoThreshold": <relaxAllUpThreshold>,
    "maxNumberRules": <maxNumberOfRules>,
    "maxFraudHit": <maxHitRate>,
    "rulesName": "<ruleName>",
    "rulesComment": "<comment>",
    "aPrioriConditions": [ <predefinedConditions> ],
    "ruleConclusion": {
      "interceptValue": <interceptValue>
    }
  }
}
```

- "revision" (mandatory): The revision whose rule generation settings shall be changed
- "data:balanceFraudFalsePositives" (optional): Determines how much IBM Safer Payments favors hit rate over false positives
- "data:lowerBoundFalsePositives" (optional): Any false positive value below this value is considered to be indifferent to this value
- "data:upperBoundFraudHitRate" (optional): Hit rate in percent at which automatic rule generation stops
- "data:stopRuleFalsePositives" (optional): Stop rule generation if the last generated rule has false positives equal to or greater than this setting
- "data:maxNumberConditions" (optional): Maximum number of conditions that are generated for each rule

- "data:minPopulationQuantise" (optional): Minimum number of records that a quantile must have
- "data:minSampleSizePercent" (optional): Relative number of records covered by a new condition that must be met to be suggested as a condition
- "data:minSelectHitRate" (optional): Minimum hit rate for automated selection of an indicator
- "data:minConditionImprovement" (optional): Minimum quality improvement that adding a new condition must provide to be suggested
- "data:relaxAllUpAutoThreshold" (optional): Threshold in quality for complete relaxation of indicators up from current
- "data:maxNumberRules" (optional): Maximum number of rules that shall be generated during fully automated rule generation
- "data:maxFraudHit" (optional): Any hit rate value above this setting is considered to be indifferent to the value
- "data:rulesName" (optional): A template for the names of the generated rules. Use {n} for rule number
- "data:rulesComment" (optional): A template for the comments of the generated rules. Available placeholders: {n}, {timestamp}, {revision}, {user}
- "data:aPrioriConditions" (optional): Conditions that a record must meet to be included in rule generation
- "data:ruleConclusion:interceptValue" (optional): The intercept value which will be set by all generated rules as a rule conclusion

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

385 SetRulesEnabled

Detailed Description

SetRulesEnabled enables or disables one or multiple rules.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setRulesEnabled",
  "revision": <revisionUid>,
  "data": {
    "enabled": true|false,
    "rules": [<ruleUid>]
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) of the rules to be enabled/disabled.
- "data:enabled" (mandatory): A flag whether the rules should be enabled or disabled.
- "data:rules" (mandatory): An array of Rules (identified by their uids) to be enabled/disabled.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

386 SetSimulationDataSelection

Detailed Description

SetSimulationDataSelection Sets the data selection of a Simulation. The desired data selection will be adjusted to fit into the actually available data set of the mandator.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setSimulationDataSelection",
  "revision": <revisionUid>,
  "data": {
    "simulationDataSelection": {
      "mandators": [<mandatorUid>],
      "periodType": "RecordsAbsolute" | "TimeAbsolute",
      "desired": {
        "fromUrid": <fromUrid>,
        "fromTimestamp": <fromTimestamp>,
        "toUrid": <toUrid>,
        "toTimestamp": <toTimestamp>,
      },
      "incDdc": true|false,
      "conditions": [ <conditions> ]
    }
  }
}
"simulationMethod": "PerChunk" | "PerRecord"
```

- "revision" (mandatory): The Revision (identified by its uid) of the simulation.
- "data:simulationDataSelection:mandators" (mandatory): The mandators (identified by their uids) whose data should be included in the simulation.
- "data:simulationDataSelection:periodType" (mandatory): A string that indicates whether the data selection refers to urids or a time span.
- "data:simulationDataSelection:desired:fromUrid" (optional): If periodType "RecordsAbsolute", the desired first urid of the data selection.
- "data:simulationDataSelection:desired:fromTimestamp" (optional): If periodType "TimeAbsolute", the desired start date of the data selection.
- "data:simulationDataSelection:desired:toUrid" (optional): If periodType "RecordsAbsolute", the desired last urid of the data selection.
- "data:simulationDataSelection:desired:toTimestamp" (optional): If periodType "TimeAbsolute", the desired end date of the data selection.
- "data:simulationDataSelection:incDdc" (mandatory): A flag whether or not data from the ddc may be accessed during simulation.
- "data:simulationDataSelection:incDdc" (optional): An array of Conditions to further restrict the data selection.
- "simulationMethod" (mandatory): A string that indicates whether the simulation is performed per chunk or per record.

Returns

A HTTP response with the following content:

```
{  
  "responseStatus": ["<status>", "<feedbacktext>"],  
  "reloadUserProfile": true|false  
}
```

387 SetUserExportPassword

Detailed Description

SetUserExportPassword sets a password for the next csv export of a table. It is needed when the instance is encrypted and "encrypt sensitive exports" is enabled in system configuration. The following export action will use the password to encrypt the csv export.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setUserExportPassword",
  "uid": "<queryUid>|<groupByQueryUid>|<definedRiskListUid>",
  "revision": "<revisionUid>",
  "mandator": "<mandator>",
  "data": {
    "streamType": "casesExport" | "cppExport" | "queryResultExport" |
"definedRiskListExport",
    "exportPassword": "<password>"
  }
}
```

- "uid" (optional): The Query/GroupByQuery/DefinedRiskList (identified by its uid), if the streamType is "queryResultExport" or "definedRiskListExport".
- "revision" (optional): The Revision (identified by its uid), if the streamType is "queryResultExport" and the export is for a simulation query.
- "mandator" (optional): The Mandator (identified by his uid) of the exported element, if the streamType is "definedRiskListExport".
- "data:streamType" (mandatory): A string to identify which kind of table is exported.
- "data:exportPassword" (mandatory): The password to be set.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

388 SetXdcSizes

Detailed Description

SetXdcSizes changes all MDC and/or DDC sizes of all attributes in a given challenger revision.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "setXdcSizes",
  "type": "DDC" | "MDC" | "MdcWhereDdc",
  "revision": <revisionUid>,
  "data": {
    "newSize": <newSize>
  }
}
```

- "type" (mandatory): Decides which storage size is set to the new size
- "revision" (mandatory): Identifies the challenger revision to change sizes
- "data:newSize" (mandatory): The size which is supposed to be set

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

389 Shutdown

Detailed Description

Shutdown performs shutdown of an instance in the cluster. The requesting User must have the privilege to change the Safer Payments cluster.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "shutdown",
    "uid": <instanceUid>
}
```

- "uid" (mandatory): The IrisInstance to shutdown.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbackText>"],
    "reloadUserProfile": true|false
}
```


390 SimulateCollusion

Detailed Description

SimulateCollusion starts simulation of a Collusion. It requires a valid general Simulation DataSelection.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "simulateCollusion",
    "uid": <collusionUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The collusion (identified by its uid) to simulate.
- "revision" (mandatory): The Revision (identified by its uid) where the collusion simulation takes place.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

391 StartAnalysis

Detailed Description

StartAnalysis starts the computation of one or multiple Analysis. It requires a valid Simulation.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "startAnalysis",
  "revision": <revisionUid>,
  "data": {
    "analysisId": [<analysisId>]
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the analyses are executed.
- "data:analysisId" (mandatory): An array with analyses (identified by their ids) to execute.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

392 StartAutomaticRuleGeneration

Detailed Description

StartAutomaticRuleGeneration start a rule generation. It requires at least one attribute that is enabled for rule generation in modelling and a valid rule generation DataSelection.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "startAutomaticRuleGeneration",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the rule generation is executed.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

393 StopAnalysis

Detailed Description

StopAnalysis stops one or multiple running analyses and discards all results of the analyses.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "stopAnalysis",
  "revision": <revisionUid>,
  "data": {
    "analysisId": [<analysisId>]
  }
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the analysis is running.
- "data:analysisId" (mandatory): The Analysis (identified by its id) to stop.

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

394 StopAutomaticRuleGeneration

Detailed Description

StopAutomaticRuleGeneration stops a running rule generation and discards all results of the RuleGenerator.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "stopAutomaticRuleGeneration",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The Revision (identified by its uid) where the rule generation is running.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

395 StopCollusionSimulation

Detailed Description

StopCollusionSimulation stops a running simulation of a Collusion.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "stopCollusionSimulation",
    "uid": <collusionUid>,
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The Collusion (identified by its uid) that is currently simulated.
- "revision" (mandatory): The Revision (identified by its uid) where the collusion simulation takes place.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

396 StopJob

Detailed Description

StopJob is used to stop a job immediately. BUser privilege to execute jobs is needed.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "stopJob",
    "uid": <jobUid>,
}
```

- "uid": The uid of the job to be stopped

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

397 StopQuery

Detailed Description

StopQuery stops a query during its computation. This request can be used for investigation as well as for simulation queries.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "stopQuery",
    "uid": <queryUid>,
    "revision": <revisionUid>,
    "data": {
        "resultId": <resultId>
    }
}
```

- "uid" (mandatory): Uid of the query to be stopped
- "revision" (optional): If the query is a simulation query, it's revision has to be provided
- "data:resultId" (mandatory): The id of the result which computation is supposed to be stopped

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


398 StopRuleGeneration

Detailed Description

StopRuleGeneration discards rules and stop a running rule generation in a revision. All computed data and generated rules will be discarded.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "stopRuleGeneration",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The revision whose rule generation shall be stopped

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

399 TakeoverCase

Detailed Description

TakeoverCase takes over a case which is investigated (i.e. reserved) by another user. It will then be reserved to the requesting user.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "takeoverCase",
    "uid": <caseUid>
}
```

- "uid" (mandatory): Uid of the case which is to be taken over

Returns

A HTTP response with the following content:

```
{
    "data": {
        "streamType": "case",
        "case": {
            "uid": <caseUid>,
            ... ..
        },
        ... ..
    },
    "editable": true|false,
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

400 TakeoverRevision

Detailed Description

TakeoverRevision takes over a revision which is modelled by a different user. A revision is always reserved so that only one user may work on it. This request allows to overtake a revision from another user and reserve it for oneself.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "takeoverRevision",
    "revision": <revisionUid>
}
```

- "revision" (mandatory): The uid of the revision which is to be taken over

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

401 Test

Detailed Description

Test sends a case action/notification/external query, optionally with data from a provided case, and thus test whether IBM Safer Payments may communicate properly with the external system.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "test",
  "uid": <caseActionUid>,
  "mandator": <mandatorUid>,
  "data": {
    "caseUid": <caseUid>|-1
    "textModuleUid": <textModuleUid>|-1
  }
}
```

- "uid" (mandatory): The uid of the case action/notification/external query to be tested
- "mandator" (mandatory): The mandator to which the element belongs
- "data:caseUid" (optional): If a case uid is provided, placeholders will be filled from that case
- "data:textModuleUid" (optional): If a text module uid is provided, [TextModule] placeholders will be filled from that

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

402 TestLdapConnection

Detailed Description

TestLdapConnection checks whether it is possible to connect & login to an LDAP server. It is used when setting up LDAP login in IBM Safer Payments. It tries to connect to a given LDAP server with given credentials and returns whether it was possible to connect or not.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "testLdapConnection",
  "data": {
    "ldapLogin": "<userName>",
    "useActiveDirectory": true|false,
    "ldapPassword": "<ldapPassword>",
    "hostName": "<ldapServerHostName>",
    "baseDn": "<baseDn>",
    "domain": "<domain>",
    "ldapPort": "<ldapPort>",
    "ssl": true|false
  }
}
```

- "data:ldapLogin" (mandatory): Login name for the LDAP server
- "data:useActiveDirectory" (optional): If enabled, active directory authentication will be used
- "data:ldapPassword" (optional): Password for the user login to the LDAP server
- "data:hostName" (optional): IP address or domain name of LDAP server
- "data:baseDn" (optional): Base of distinguished name
- "data:domain" (optional): Active Directory domain for login
- "data:ldapPort" (optional): IP port of LDAP server
- "data:ssl" (optional): Enables encrypted communication between IBM Safer Payments and LDAP host

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

403 UnmarkFraud

Detailed Description

UnmarkFraud is used to manually mark transactions as genuine instead of fraud after they are processed. The user can choose a custom fraud value and decide to provide a timestamp attribute which saves the time of marking.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "unmarkFraud",
    "data": {
        "urid": [<urid>],
        "fraudValue": <genuineValue>,
        "fraudTimestampAttribute": <timestampAttributeUid>|-1
    }
}
```

- "data:urid" (mandatory): A list of URIDs which shall be marked as genuine.
- "data:fraudValue" (mandatory): The fraud value which shall be assigned to the selected URIDs.
- "fraudTimestampAttribute" (optional): Provide a fraud timestamp attribute to save the time of marking.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```

404 Unreserve

Detailed Description

Unreserve unreserves an element so that it can be edited by another User. The requesting User must have the privilege to change the element.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "unreserve",
    "uid": <elementUid>
    "revision": <revisionUid>
}
```

- "uid" (mandatory): The element to unreserve.
- "revision" (optional): The Revision to look in. Usage depends on type of element.

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbackText>"],
    "reloadUserProfile": true|false
}
```

405 UnreserveCase

Detailed Description

UnreserveCase unreserves a case.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
    "request": "unreserveCase",
    "uid": <caseUid>
}
```

Returns

A HTTP response with the following content:

```
{
    "responseStatus": ["<status>", "<feedbacktext>"],
    "reloadUserProfile": true|false
}
```


406 UnreserveDefinedRiskListEntry

Detailed Description

UnreserveDefinedRiskListEntry unreserves a defined risk list entry.

Parameters

httpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "unreserveDefinedRiskListEntry",
  "uid": <definedRiskListUid>,
  "mandator": <mandatorUid>,
  "data": {
    "attributeValue": <attributeValue>,
    "id": <definedRiskListEntryId>
  }
}
```

Returns

A HTTP response with the following content:

```
{
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true|false
}
```

407 UpdateRiskListEntry

Detailed Description

UpdateRiskListEntry updates a Risk List Entry with the values provided with the http request.

Parameters

HttpRequest: Holds the request parameters that have been passed to the server:

```
{
  "request": "updateRiskListEntry",
  "uid" : <definedRiskListUid>,
  "mandator" : <mandatorUid>,
  "data" : {
    "value": "<inputAttributeValue>",
    "outputAttributeCategoryValue": "<outputAttributeCategoryValue>" |
"undefined",
    "expiresAt": <expiresAt>,
    "startsAt": <startsAt>,
    "id": <entryId>,
    "comment": "<comment>",
    "label": "<label>",
    "active": true | false,
    "conditions": [<conditions>]
  }
}
```

- "uid": The uid of the risk list, in which the entry should be updated.
- "mandator": The uid of mandator, for which the risk list is defined.
- "data:value": The value of the entry.
- "data:outputAttributeCategoryValue" : If a category is defined for the attribute, the category value has to be send.
- "data:expiresAt": If the entry should expire, the date can be assigned here.
- "data:startsAt": If the entry should only be computed after a defined date, the date can be assigned here.
- "data:id": The id of the entry in the list.
- "data:comment" : A comment for this entry can be assigned/changed.
- "data:label" : The id of the entry in the list.
- "data:active" : Defines if the entry should be active or inactive.
- "data:conditions" : Additional conditions which have to be met for this entry to be applied in computation.

Returns

A HTTP response with the following content :

```
{
  "comparisonResult": "Unchanged" | "ChangedNoImpact" | "ChangedComputationImpact",
  "responseStatus": ["<status>", "<feedbacktext>"],
  "reloadUserProfile": true | false
}
```

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you. Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Terms and Conditions for Product Documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein. IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed. You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Users' Comments - We'd Like to Hear from You

We appreciate your comments about this publication. Please contact [IBM Technical Support](#) if you have question or wish to make comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this reference book. The comment you send should pertain to only the information in this manual or product and the way in which the information is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.