

IBM **Guardium**
Discover and Classify

DSAR REST API

IBM GUARDIUM DISCOVER AND CLASSIFY

VERSION 4.2.2

REST API AUTHENTICATION

IGDC supports OAuth M2M authentication.



To access REST API using a jwt token, you must we configure a separate client in Keycloak and use the client's credentials. See instructions on how to configure a client and get credentials [below](#).

1. Exchange credentials for the jwt token.

```
curl -X POST "https://[host]/auth/realms/cm-realm/protocol/openid-connect/token" \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "grant_type=client_credentials" \
-d "client_id=m2mcl" \
-d "client_secret=vgcguH8OyjSHacDLQkZ7YJ00gT0MOGph" \
-d "scope=offline_access"
```

Output sample:

```
{"access_token":"eyJhbGciOiJSUzI1NiIuLlRNUHDDWOCsA5fFkA","expires_in":300,"refresh_expires_in":0,"token_type":"Bearer","not-before-policy":0,"scope":"profile email offline_access"}
```

2. Use the access token from the output above for API access. For example:

```
curl -X POST "https://[Host]/pii/api/advanced-search/repositories" \
-H "Authorization: Bearer eyJhbGciOiJSUzI1NiIuLlRNUHDDWOCsA5fFkA" \
-H "Content-Type: application/json" \
--data "{}"
```

HOW TO CONFIGURE A CLIENT IN KEYCLOAK

1. Log in to Keycloak. In the dropdown menu on the left, change the current realm to **cm-realm**.

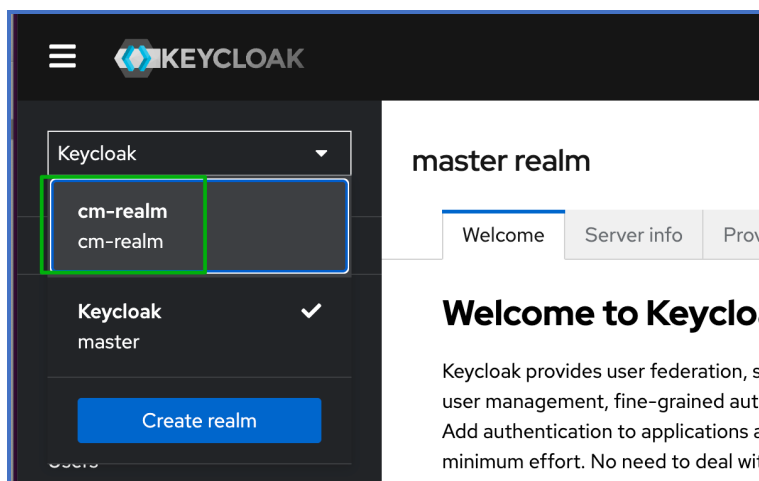
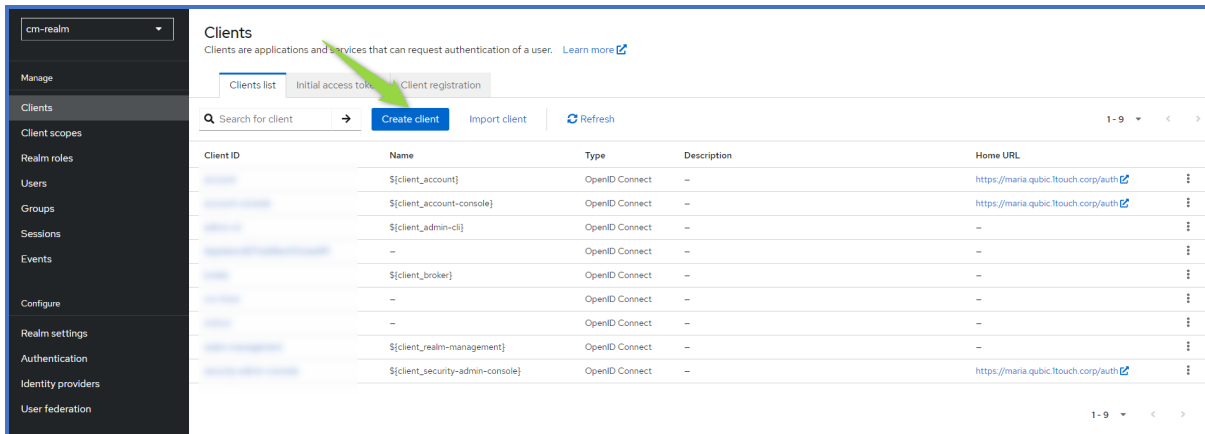
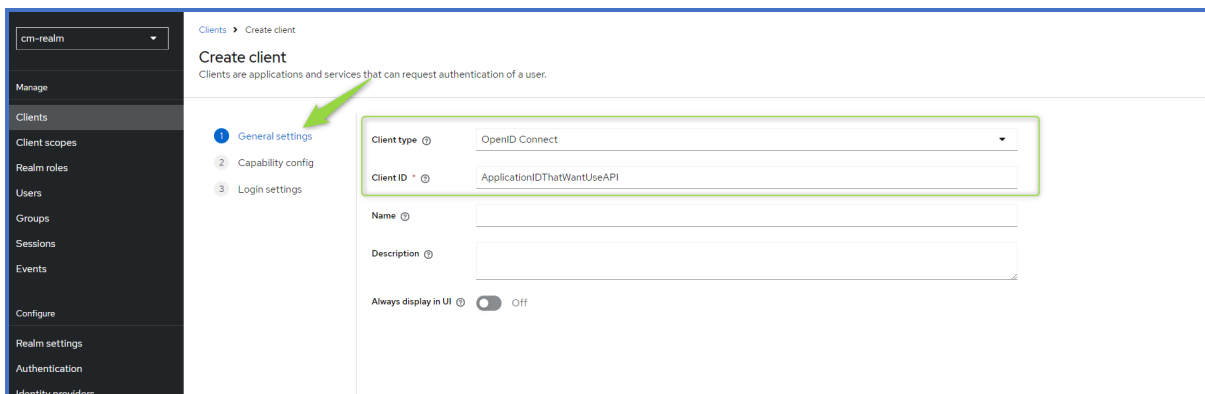


Figure 1: Keycloak realm change

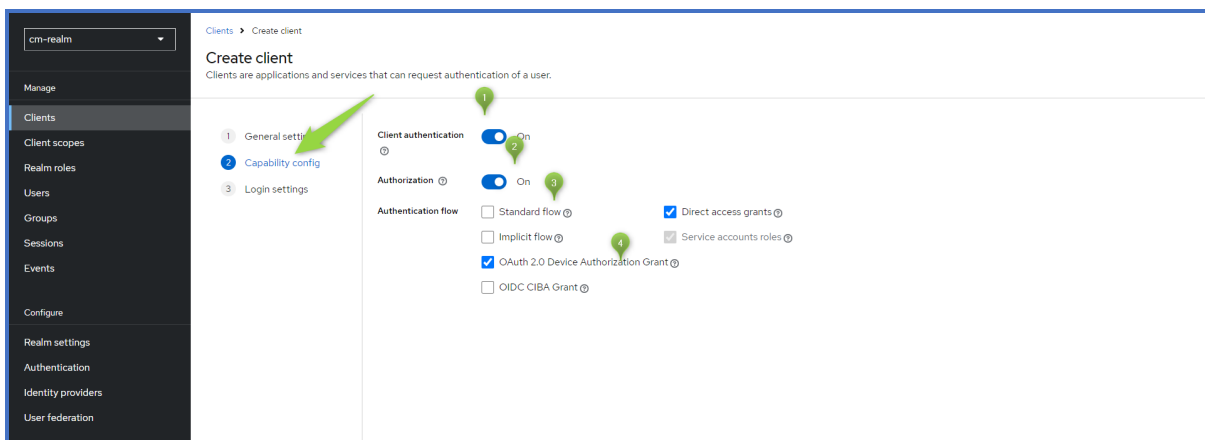
2. In Keycloak **cm-realm**, go to **Manage > Clients** and click **Create client**.



3. In the **General settings** form, select **OpenID Connect** in the **Client type** field. Then enter the id of the application in the **Client ID** field.



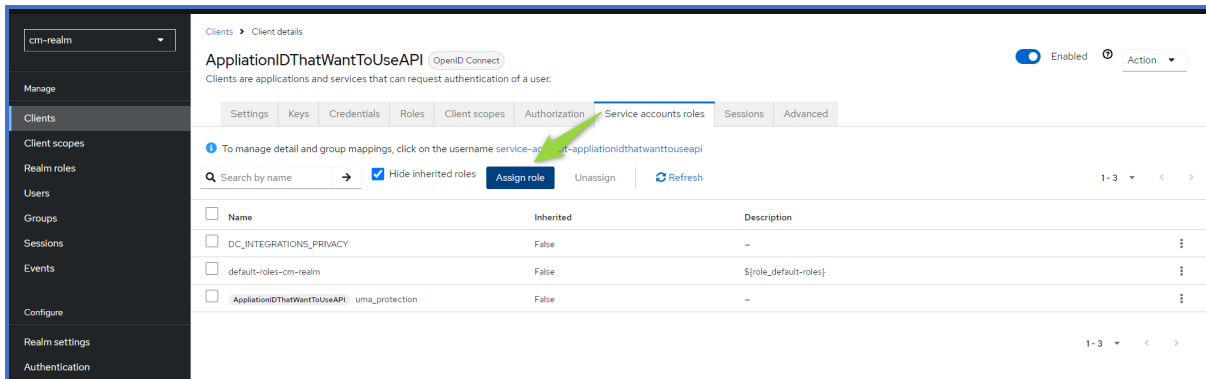
4. In the **Capability config** form, turn on the **Client authentication** (1) and **Authorization** (2) toggles. Then disable the **Standard flow** checkbox (3), and enable the **OAuth 2.0 Device Authorization Grant** checkbox (4).



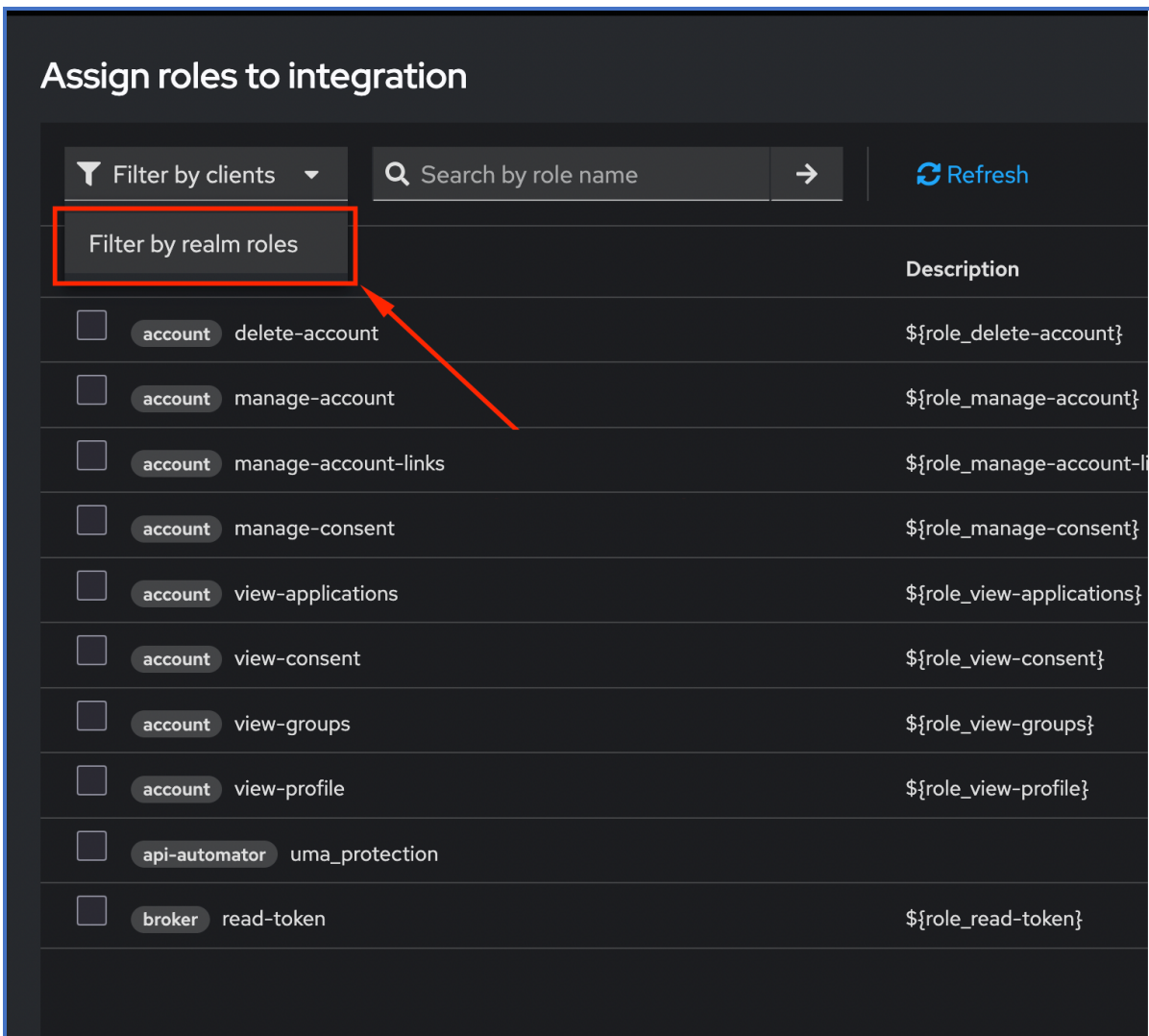
5. Leave empty all fields in the **Login settings** form.



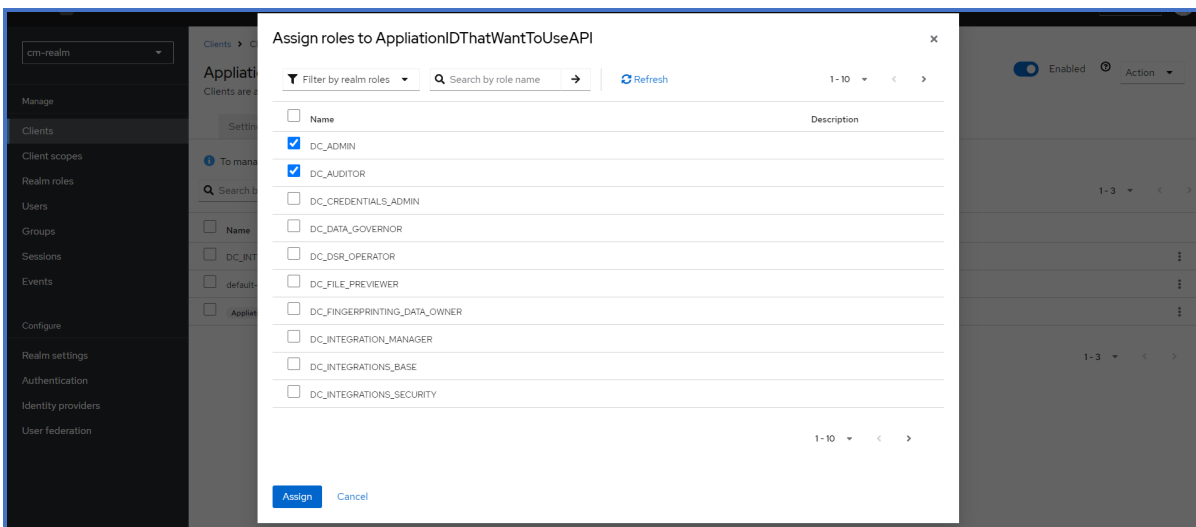
6. Go to the **Service account roles** tab in the new client profile and click **Assign role**.



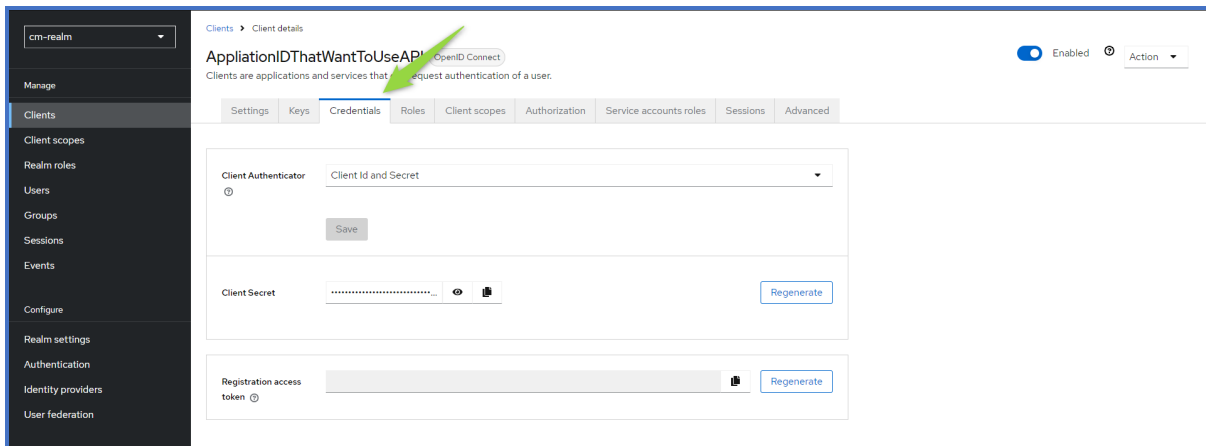
7. In the **Assign roles** popup, select the **Filter by realm roles** option.



8. Select the desired roles and click **Assign**.



9. Go to the **Credentials** tab and copy the client credentials.



DSAR REST API -

METHODS

[[Jump to Models](#)]

TABLE OF CONTENTS

[Chapter Controller](#)

- [get /dsar/api/chapters/default](#)

[Email Server Config Controller](#)

- [post /dsar/api/emails/server/check](#)
- [delete /dsar/api/emails/server](#)
- [get /dsar/api/emails/server](#)
- [put /dsar/api/emails/server](#)

[Email Template Controller](#)

- [post /dsar/api/emails/templates](#)
- [delete /dsar/api/emails/templates/{id}](#)
- [get /dsar/api/emails/templates](#)
- [get /dsar/api/emails/templates/{id}](#)
- [patch /dsar/api/emails/templates/{id}/setDefault](#)
- [put /dsar/api/emails/templates/{id}](#)

[Logo Controller](#)

- [delete /dsar/api/logo/{logoId}](#)
- [get /dsar/api/logo/{logoId}](#)
- [post /dsar/api/logo](#)

[Report Controller](#)

- [get /dsar/api/pii/{piiId}/reports/preview](#)
- [get /dsar/api/pii/{piiId}/reports](#)
- [post /dsar/api/pii/{piiId}/reports/send](#)

[Report History Controller](#)

- [get /dsar/api/reports/history](#)
- [get /dsar/api/reports/history/filters/entity-types](#)

- [post /dsar/api/reports/history/search](#)
- [get /dsar/api/reports/history/filters](#)

Report Template Controller

- [post /dsar/api/reports/templates](#)
- [delete /dsar/api/reports/templates/{id}](#)
- [get /dsar/api/reports/templates](#)
- [get /dsar/api/reports/templates/{id}](#)
- [patch /dsar/api/reports/templates/{id}/setDefault](#)
- [put /dsar/api/reports/templates/{id}](#)

CHAPTER CONTROLLER

[Up](#)

get /dsar/api/chapters/default

Get default chapters (get2)

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

EMAIL SERVER CONFIG CONTROLLER

[Up](#)

post /dsar/api/emails/server/check

Test email server configuration (checkEmailServerConfiguration)

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [EmailServerConfigRequest](#) (required)

Body Parameter –

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

delete /dsar/api/emails/server

Delete email server configuration
(deleteEmailServerConfiguration)

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES**200**

OK

400Bad Request [ApiErrorDto](#)**404**Not Found [ApiErrorDto](#)**500**Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/emails/server

Get email server configuration (getEmailServerConfiguration)

RETURN TYPE[EmailServerConfigDto](#)**EXAMPLE DATA**

Content-Type: application/json

```
{
  "hostname" : "",
  "port" : "",
  "authType" : "",
  "username" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES**200**OK [EmailServerConfigDto](#)**400**Bad Request [ApiErrorDto](#)**404**Not Found [ApiErrorDto](#)**500**Internal Server Error [ApiErrorDto](#)

[Up](#)

put /dsar/api/emails/server

Save/update email server configuration (saveEmailServerConfiguration)

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [EmailServerConfigRequest](#) (required)

Body Parameter –

RETURN TYPE

[EmailServerConfigDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "hostname" : "",
  "port" : "",
  "authType" : "",
  "username" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [EmailServerConfigDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

EMAIL TEMPLATE CONTROLLER

[Up](#)

post /dsar/api/emails/templates

Create email template (createEmailTemplate)

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [EmailTemplateRequest](#) (required)

Body Parameter –

RETURN TYPE

[EmailTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "subject" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "text" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [EmailTemplateDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

delete /dsar/api/emails/templates/{id}

Delete email template (deleteEmailTemplate)

PATH PARAMETERS

id (required)

Path Parameter –

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/emails/templates

Get all email templates (getAllEmailTemplates)

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/emails/templates/{id}

Get email template by id (getEmailTemplate)

PATH PARAMETERS

id (required)

Path Parameter –

RETURN TYPE

[EmailTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "subject" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "text" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [EmailTemplateDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

Up

patch /dsar/api/emails/templates/{id}/setDefault

Set default email template (setDefaultEmailTemplate)

PATH PARAMETERS

id (required)

Path Parameter –

RETURN TYPE

[EmailTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "subject" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "text" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES**200**

OK [EmailTemplateDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

put /dsar/api/emails/templates/{id}

Update email template (updateEmailTemplate)

PATH PARAMETERS

id (required)

Path Parameter –

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [EmailTemplateRequest](#) (required)

Body Parameter –

RETURN TYPE

[EmailTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "subject" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "text" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [EmailTemplateDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

LOGO CONTROLLER

[Up](#)

```
delete /dsar/api/logo/{logoId}
```

Delete logo (delete1)

PATH PARAMETERS

logoId (required)

Path Parameter –

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

```
get /dsar/api/logo/{logoId}
```

Download logo (download)

PATH PARAMETERS

logoId (required)

Path Parameter –

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- image/png
- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

post /dsar/api/logo

Upload logo (upload)

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [logo_body](#) (optional)

Body Parameter –

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

REPORT CONTROLLER

[Up](#)

```
get /dsar/api/pii/{piiId}/reports/preview
```

Preview pdf report with data subject(`getPreviewReport`)

PATH PARAMETERS

`piiId` (required)
Path Parameter –

QUERY PARAMETERS

`reportTemplateId` (optional)
Query Parameter –

RETURN TYPE

[ReportDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "personalInfo" : "",
  "chapters" : "",
  "reasonOfProcessing" : "",
  "title" : "",
  "socialMedia" : {
    "instagramLink" : "",
    "linkedinLink" : "",
    "twitterLink" : "",
    "facebookLink" : ""
  },
  "introduction" : "",
  "logoId" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [ReportDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/pii/{piiId}/reports

Generate pdf report with data subject(getReport)

PATH PARAMETERS

piiId (required)

Path Parameter –

QUERY PARAMETERS

reportTemplateId (optional)

Query Parameter –

fileName (optional)

Query Parameter – default: PII Report

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/pdf
- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

post /dsar/api/pii/{piiId}/reports/send

Send pdf report by email (sendReportByEmail)

PATH PARAMETERS

piiId (required)

Path Parameter –

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [SendReportRequest](#) (required)

Body Parameter –

QUERY PARAMETERS

fileName (optional)

Query Parameter – default: PII Report

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

REPORT HISTORY CONTROLLER

[Up](#)

get /dsar/api/reports/history

Get report history (get1)

QUERY PARAMETERS

offset (optional)

Query Parameter – default: 0 format: int32

limit (optional)

Query Parameter – default: 100 format: int32

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/reports/history/filters/entity-types

Get data element filters for report history
(getEntityTypeFilters)

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES**200**

OK

400Bad Request [ApiErrorDto](#)**404**Not Found [ApiErrorDto](#)**500**Internal Server Error [ApiErrorDto](#)[Up](#)

post /dsar/api/reports/history/search

Get filtered report history (getFilteredReportHistory)

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODYbody [ReportHistoryFilter](#) (required)

Body Parameter –

QUERY PARAMETERS

offset (optional)

Query Parameter – default: 0 format: int32

limit (optional)

Query Parameter – default: 100 format: int32

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/reports/history/filters

Get filters for report history (getFilters)

RETURN TYPE

[ReportHistoryFilterDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "sentToEmails" : "",
  "actions" : "",
  "piiNames" : ""
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [ReportHistoryFilterDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

REPORT TEMPLATE CONTROLLER

[Up](#)

post /dsar/api/reports/templates

Create report template (create)

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [ReportTemplateRequest](#) (required)

Body Parameter –

RETURN TYPE

[ReportTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "content" : {
    "chapters" : "",
    "title" : "",
    "socialMedia" : {
      "instagramLink" : "",
      "linkedinLink" : "",
      "twitterLink" : "",
      "facebookLink" : ""
    },
    "introduction" : "",
    "logoId" : ""
  },
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES**200**OK [ReportTemplateDto](#)**400**Bad Request [ApiErrorDto](#)**404**Not Found [ApiErrorDto](#)**500**Internal Server Error [ApiErrorDto](#)

[Up](#)

delete /dsar/api/reports/templates/{id}

Delete report template (delete)

PATH PARAMETERS

id (required)

Path Parameter –

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES**200**

OK

400Bad Request [ApiErrorDto](#)**404**Not Found [ApiErrorDto](#)**500**Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/reports/templates

Get all report templates (get)

EXAMPLE DATA

Content-Type: application/json

""

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

get /dsar/api/reports/templates/{id}

Get report template by id (getById)

PATH PARAMETERS

id (required)

Path Parameter –

RETURN TYPE

[ReportTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "content" : {
    "chapters" : "",
    "title" : "",
  }
}
```

```
"socialMedia" : {
  "instagramLink" : "",
  "linkedinLink" : "",
  "twitterLink" : "",
  "facebookLink" : ""
},
"introduction" : "",
"logoId" : ""
}
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [ReportTemplateDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

patch /dsar/api/reports/templates/{id}/setDefault

Set report template as default (setDefault)

PATH PARAMETERS

id (required)

Path Parameter —

RETURN TYPE

[ReportTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "content" : {
    "chapters" : "",
    "title" : "",
    "socialMedia" : {
      "instagramLink" : "",
      "linkedinLink" : "",
      "twitterLink" : "",
      "facebookLink" : ""
    },
    "introduction" : "",
    "logoId" : ""
  }
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [ReportTemplateDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

[Up](#)

put /dsar/api/reports/templates/{id}

Update report template (update)

PATH PARAMETERS

id (required)

Path Parameter —

CONSUMES

This API call consumes the following media types via the Content-Type request header:

- application/json

REQUEST BODY

body [ReportTemplateRequest](#) (required)

Body Parameter –

RETURN TYPE

[ReportTemplateDto](#)

EXAMPLE DATA

Content-Type: application/json

```
{
  "default" : "",
  "name" : "",
  "description" : "",
  "id" : "",
  "content" : {
    "chapters" : "",
    "title" : "",
    "socialMedia" : {
      "instagramLink" : "",
      "linkedinLink" : "",
      "twitterLink" : "",
      "facebookLink" : ""
    },
    "introduction" : "",
    "logoId" : ""
  }
}
```

PRODUCES

This API call produces the following media types according to the Accept request header; the media type will be conveyed by the Content-Type response header.

- application/json

RESPONSES

200

OK [ReportTemplateDto](#)

400

Bad Request [ApiErrorDto](#)

404

Not Found [ApiErrorDto](#)

500

Internal Server Error [ApiErrorDto](#)

MODELS

[[Jump to Methods](#)]

TABLE OF CONTENTS

1. [ActionChapter](#)
2. [ActionDiff](#)
3. [AddPiiValue](#)
4. [AddPiiValueItem](#)
5. [ApiErrorDto](#)
6. [ChangePiiValue](#)
7. [ChangePiiValueItem](#)
8. [Chapter](#)
9. [ChapterDiff](#)
10. [DeletePiiValue](#)
11. [DeletePiiValueItem](#)
12. [Diff](#)
13. [EmailServerConfigDto](#)
14. [EmailServerConfigRequest](#)
15. [EmailTemplateDto](#)
16. [EmailTemplateRequest](#)
17. [FormChapter](#)
18. [FormDiff](#)
19. [Legal](#)
20. [LegalDiff](#)
21. [LegalItemDiff](#)
22. [LegalTextChapter](#)
23. [LegalTextChapterDiff](#)
24. [PiiValue](#)

25. [PiiValueDiff](#)
26. [PiiValueItem](#)
27. [ReasonOfProcessingDiff](#)
28. [ReportContentDto](#)
29. [ReportDiff](#)
30. [ReportDto](#)
31. [ReportHistoryDto](#)
32. [ReportHistoryFilter](#)
33. [ReportHistoryFilterDto](#)
34. [ReportTemplateDto](#)
35. [ReportTemplateRequest](#)
36. [SendReportRequest](#)
37. [SocialMedia](#)
38. [SocialMediaDiff](#)
39. [SocialMediaItemDiff](#)
40. [StrDiff](#)
41. [TextChapter](#)
42. [TextDiff](#)
43. [logo_body](#)

ACTIONCHAPTER[UP](#)

ACTIONDIFF[UP](#)

ADDPIIVALUE[UP](#)

ADDPIIVALUEITEM[UP](#)

APIERRORDTO[UP](#)

error (optional)
errorCode (optional)
data (optional)

CHANGEPIIVALUE[UP](#)

CHANGEPIIVALUEITEM[UP](#)

CHAPTER[UP](#)

name (optional)

type (optional)

CHAPTERDIFF[UP](#)

name (optional)

diff (optional)

changeType (optional)

DELETEPIIVALUE[UP](#)

DELETEPIIVALUEITEM[UP](#)

DIFF[UP](#)

type (optional)

EMAILSERVERCONFIGD[TO](#)[UP](#)

hostname (optional)

username (optional)

authType (optional)

port (optional)

format: int32

EMAILSERVERCONFIGR[EQUEST](#)[UP](#)

hostname

username

password (optional)

authType

port (optional)

format: int32

EMAILTEMPLATED[TO](#)[UP](#)

id (optional)

name (optional)

description (optional)

subject (optional)

text (optional)

default (optional)

EMAILTEMPLATER[EQUEST](#)[UP](#)

name

description

subject

text

FORMCHAPTER[UP](#)

FORMDIFF[UP](#)

LEGALUP

law (optional)
description (optional)
url (optional)

LEGALDIFFUP

law (optional)
changeType (optional)
diff (optional)
[LegalItemDiff](#)

LEGALITEMDIFFUP

description (optional)
[StrDiff](#)
url (optional)
[StrDiff](#)

LEGALTEXTCHAPTERUP**LEGALTEXTCHAPTERDIFFUP****PIIVALUEUP**

value
confidenceLevel
specialDataElement (optional)
protected (optional)

PIIVALUEDIFFUP

entityType (optional)
changeType (optional)

PIIVALUEITEMUP

changeType (optional)

REASONOFPROCESSINGDIFFUP

value (optional)
changeType (optional)

REPORTCONTENTDITOUP

title (optional)
introduction (optional)
chapters (optional)
logoId (optional)
socialMedia (optional)
[SocialMedia](#)

REPORTDIFFUP

title (optional)

[StrDiff](#)

introduction (optional)

[StrDiff](#)

logoId (optional)

[StrDiff](#)

chapters (optional)

reasonOfProcessing (optional)

socialMedia (optional)

[SocialMediaDiff](#)

personalInfo (optional)

REPORTD~~T~~TO~~U~~P

title (optional)

introduction (optional)

chapters (optional)

logoId (optional)

socialMedia (optional)

[SocialMedia](#)

reasonOfProcessing (optional)

personalInfo (optional)

REPORTHISTORYD~~T~~TO~~U~~P

id (optional)

piiId (optional)

piiName (optional)

templateId (optional)

templateName (optional)

action (optional)

generatedBy (optional)

createdAt (optional)

format: date-time

sentTo (optional)

diff (optional)

[ReportDiff](#)**REPORTHISTORYFILTER~~U~~P**

piiNames (optional)

sentToEmails (optional)

actions (optional)

entityTypes (optional)

timestampFrom (optional)

format: date-time

timestampTo (optional)

format: date-time

REPORTHISTORYFILTERDTO[UP](#)

piiNames (optional)
sentToEmails (optional)
actions (optional)

REPORTTEMPLATEDTO[UP](#)

id (optional)
name (optional)
description (optional)
content (optional)
[ReportContentDto](#)
default (optional)

REPORTTEMPLATEREQUEST[UP](#)

name (optional)
description (optional)
title (optional)
introduction (optional)
chapters (optional)
logoId (optional)
socialMedia (optional)
[SocialMedia](#)

SENDREPORTREQUEST[UP](#)

toEmail
subject
emailBody
reportTemplateId
report
[ReportDto](#)

SOCIALMEDIA[UP](#)

facebookLink (optional)
instagramLink (optional)
linkedinLink (optional)
twitterLink (optional)

SOCIALMEDIADIFF[UP](#)

diff (optional)
[SocialMediaItemDiff](#)
changeType (optional)

SOCIALMEDIAITEMDIFF[UP](#)

facebookLink (optional)
[StrDiff](#)
instagramLink (optional)
[StrDiff](#)

linkedinLink (optional)

[StrDiff](#)

twitterLink (optional)

[StrDiff](#)

STRDIFF[UP](#)

from (optional)

to (optional)

changeType (optional)

TEXTCHAPTER[UP](#)

TEXTDIFF[UP](#)

LOGO_BODY[UP](#)

file

format: binary

IBM, the IBM logo, and IBM Guardium Discover and Classify are trademarks or registered trademarks of International Business Machines Corporation, in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on ibm.com/trademark.