

**IBM Guardium**  
Discover and Classify

# DSAR REST API

---

IBM GUARDIUM DISCOVER AND CLASSIFY

VERSION 4.1.1

## 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":"eyJhbGciOiJSUzI1Ni..RNUHDDWOCsA5fFkA","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 eyJhbGciOiJSUzI1NiI..UISxYUiW4Z4T7Cc96xmkaFhT4g5eBg" \
-H "Content-Type: application/json" \
--data "{}"
```

## HOW TO CONFIGURE A CLIENT IN KEYCLOAK

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

The screenshot shows the Keycloak Admin Console interface for the 'cm-realm'. The left sidebar contains navigation options: Manage, Clients, Client scopes, Realm roles, Users, Groups, Sessions, Events, Configure, Realm settings, Authentication, Identity providers, and User federation. The main content area is titled 'Clients' and includes a description: 'Clients are applications and services that can request authentication of a user. Learn more'. Below the description are tabs for 'Clients list', 'Initial access tokens', and 'Client registration'. A search bar is present with the text 'Search for client'. A blue 'Create client' button is highlighted with a green arrow. To the right of the button are 'Import client' and 'Refresh' options. Below the button is a table listing existing clients.

Client ID	Name	Type	Description	Home URL
...	\${client_account}	OpenID Connect	-	https://maria.qubic.ttouch corp/auth
...	\${client_account-console}	OpenID Connect	-	https://maria.qubic.ttouch corp/auth
...	\${client_admin-cli}	OpenID Connect	-	-
...	-	OpenID Connect	-	-
...	\${client_broker}	OpenID Connect	-	-
...	-	OpenID Connect	-	-
...	-	OpenID Connect	-	-
...	\${client_realm-management}	OpenID Connect	-	-
...	\${client_security-admin-console}	OpenID Connect	-	https://maria.qubic.ttouch corp/auth

### 2. 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.

cm-realm

Clients > Create client

### Create client

Clients are applications and services that can request authentication of a user.

1 General settings

2 Capability config

3 Login settings

Client type

Client ID

Name

Description

Always display in UI  Off

3. 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).

cm-realm

Clients > Create client

### Create client

Clients are applications and services that can request authentication of a user.

1 General settings

2 Capability config

3 Login settings

Client authentication  On

Authorization  On

Authentication flow

Standard flow

Implicit flow

OAuth 2.0 Device Authorization Grant

OIDC CIBA Grant

Direct access grants

Service accounts roles

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

cm-realm

Clients > Create client

### Create client

Clients are applications and services that can request authentication of a user.

1 General settings

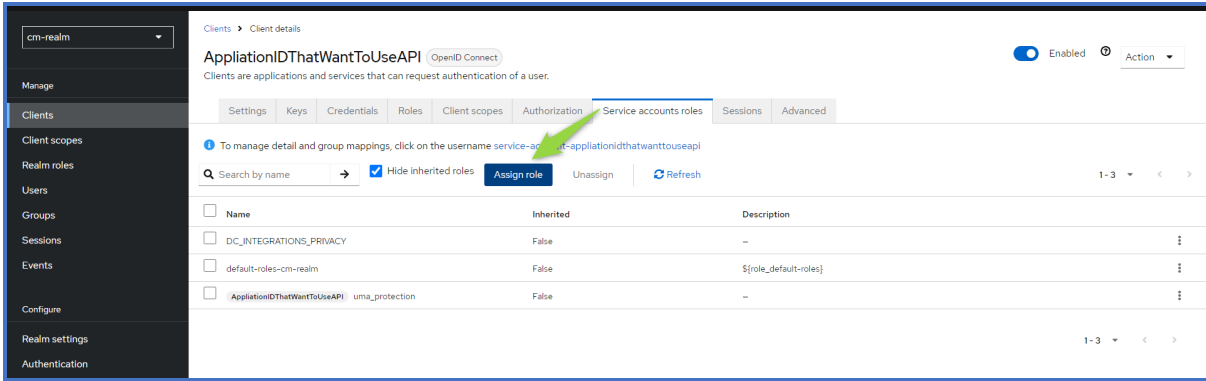
2 Capability config

3 Login settings

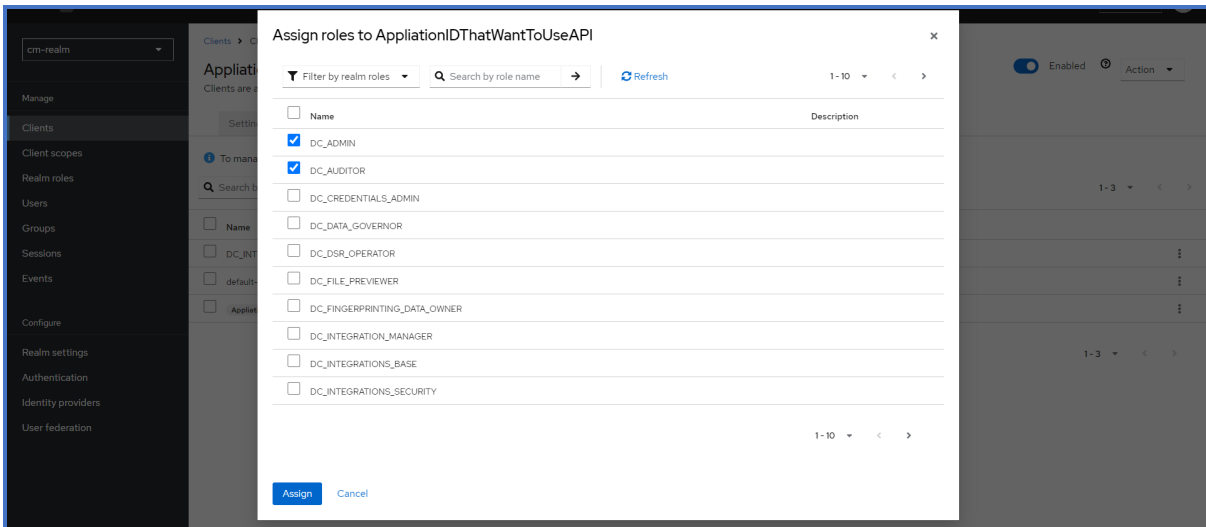
Root URL

Home URL

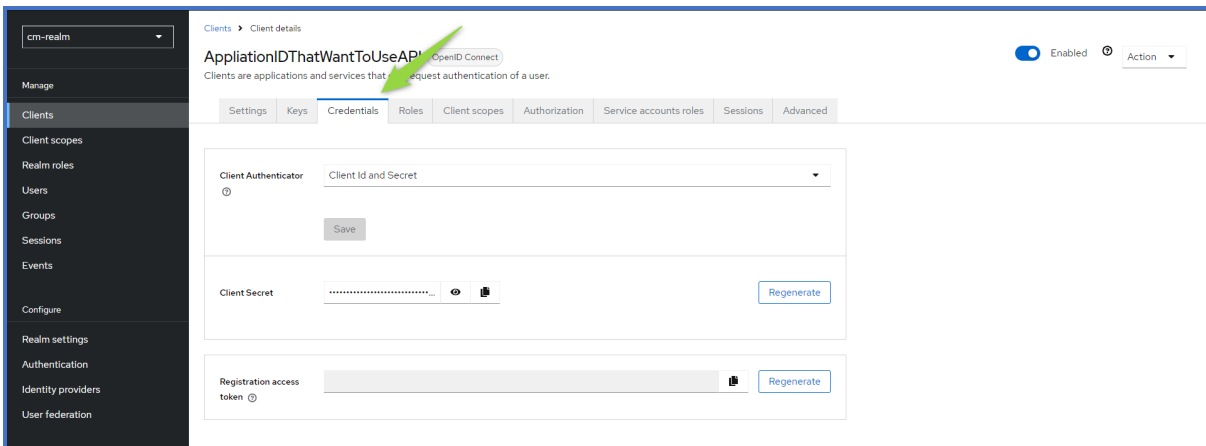
5. Go to the **Service account roles** tab in the new client profile and click **Assign role**.



6. In the **Assign roles** popup, select the desired roles and click **Assign**.



7. 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

400

Bad 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

400

Bad 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 body

body [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

400

Bad 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](#)

### **ActionChapterUp**

### **ActionDiffUp**

### **AddPiiValueUp**

### **AddPiiValueItemUp**

### **ApiErrorDtoUp**

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

### **ChangePiiValueUp**

### **ChangePiiValueItemUp**

### **ChapterUp**

name (optional)  
type (optional)

### **ChapterDiffUp**

name (optional)  
diff (optional)  
changeType (optional)

### **DeletePiiValueUp**

### **DeletePiiValueItemUp**

### **DiffUp**

type (optional)

**EmailServerConfigDtoUp**

hostname (optional)  
username (optional)  
authType (optional)  
port (optional)  
format: int32

**EmailServerConfigRequestUp**

hostname  
username  
password (optional)  
authType  
port (optional)  
format: int32

**EmailTemplateDtoUp**

id (optional)  
name (optional)  
description (optional)  
subject (optional)  
text (optional)  
default (optional)

**EmailTemplateRequestUp**

name  
description  
subject  
text

**FormChapterUp****FormDiffUp****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)

**ReportContentDtoUp**

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)

**ReportDtoUp**

title (optional)

introduction (optional)

chapters (optional)

logoId (optional)

socialMedia (optional)

[SocialMedia](#)

reasonOfProcessing (optional)

personalInfo (optional)

**ReportHistoryDtoUp**

id (optional)

piiId (optional)

piiName (optional)

templateId (optional)

templateName (optional)

action (optional)

generatedBy (optional)

createdAt (optional)

format: date-time

sentTo (optional)

diff (optional)

[ReportDiff](#)**ReportHistoryFilterUp**

piiNames (optional)

sentToEmails (optional)

actions (optional)

entityTypes (optional)

timestampFrom (optional)

format: date-time

timestampTo (optional)

format: date-time

**ReportHistoryFilterDtoUp**

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

**ReportTemplateDtoUp**

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

**ReportTemplateRequestUp**

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

**SendReportRequestUp**

toEmail  
subject  
emailBody  
reportTemplateId  
report  
[ReportDto](#)

**SocialMediaUp**

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

**SocialMediaDiffUp**

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

**SocialMediaItemDiffUp**

facebookLink (optional)

[StrDiff](#)

instagramLink (optional)

[StrDiff](#)

linkedinLink (optional)

[StrDiff](#)

twitterLink (optional)

[StrDiff](#)

**StrDiffUp**

from (optional)

to (optional)

changeType (optional)

**TextChapterUp****TextDiffUp****logo\_bodyUp**

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](http://ibm.com/trademark).