Integration Layer API

Integration Layer consists of two service endpoints. DSL accepts data from applications and the integration layer performs synchronization of that data between eligible applications, indexing and collaboration with external CSPs, i.e. sends the data to external CSPs based on the data specific trust circles or the predefined [trust circle OR team] ID. DCL accepts data from external CSPs, authorizes that data based on data specific trust circles and the integration layer performs synchronization of that data between eligible applications.

Version 1.0.4

Paths List all operations

`/v1/dsl/integrationData`

POST /v1/dsl/integrationData

dsl

Summary

Process new integration data from application

Description

Integration layer gets new data from an application. Synchronizes the data between eligible applications. Indexes the data into Elastic. Sends the data to external CSPs if toShare is set to true.

Parameters

Name	Located in	Required	Schema
intDataObj	body	Yes	⇄

```
▾ IntegrationData {
    dataParams:     ▸DataParams { }
    sharingParams:  ▸SharingParams { }
    dataType:       ▸IntegrationDataType string
    dataObject:     ▸ string
}
```

Responses

Code	Description
200	Successful operation
400	Malformed Integration Data Structure

Try this operation

PUT /v1/dsl/integrationData

dsl

Summary

Process integration data update from application

Description

Integration layer gets updated data from an application. Synchronizes the data between eligible applications. Updates the existing indexed document in Elastic. Sends the data to external CSPs if toShare is set to true.

Parameters

Name	Located in	Required	Schema
intDataObj	body	Yes	⇄

```
▾ IntegrationData {
    dataParams:     ▸DataParams { }
    sharingParams:  ▸SharingParams { }
    dataType:       ▸IntegrationDataType string
    dataObject:     ▸ string
}
```

Responses

Code	Description
200	Successful operation
400	Malformed Integration Data Structure

Try this operation

DELETE /v1/dsl/integrationData

dsl

Summary

Process integration data deletion in application

Description

Integration layer gets deletion request from an application on a specific integration data record. Forwards the deletion request to eligible applications. toShare is always set to false. Deletes the existing indexed document from Elastic. No deletion requests are forwarded to external CSPs. dataObject is always null.

Parameters

Name	Located in	Required	Schema
intDataObj	body	Yes	⇄

```
▾ IntegrationData {
    dataParams:     ▸DataParams { }
    sharingParams:  ▸SharingParams { }
    dataType:       ▸IntegrationDataType string
    dataObject:     ▸ string
  }
```

Responses

Code	Description
200	Successful operation
400	Malformed Integration Data Structure

Try this operation

/v1/dcl/integrationData

POST /v1/dcl/integrationData

dcl

Summary

Process new or updated integration data from external CSP

Description

Integration layer gets new or updated data from external CSP. Authorizes the data based on data specific trust circles. Sets the isExternal flag to true. Synchronizes the data to principal application based on data type. Principal applications expected to emit updated data back to integration layer after debuplication (/dsl will handle emitions).

Parameters

Name	Located in	Required	Schema
intDataObj	body	Yes	⇄ ▾ **IntegrationData** { dataParams: ▸ **DataParams** { } sharingParams: ▸ **SharingParams** { } dataType: ▸ **IntegrationDataType** string dataObject: ▸ string }

Responses

Code	Description
200	Successful operation
400	Malformed Integration Data Structure
403	CSP authorization failed. External CSP is not authorized to send specific data type
415	Unsupported data type. Local CSP does not support given data type. No application that handles given data type is installed

Try this operation

Models

IntegrationData

⇄ ▾ **IntegrationData** {
dataParams: ▸ **DataParams** { }
sharingParams: ▸ **SharingParams** { }
dataType: ▸ **IntegrationDataType** string
dataObject: ▾ string (json)
}

DataParams

DataParams {

origin[Csp/Application/Record]Id are only set by the application that initiates flow #1 (see SAD). In this case Origin[csp/application/record]Id and [csp/application/record]Id are equal. In the case where a principal application adapter gets data from external CSP and re-emitts them for indexing, [csp/application/record]Id are set to that of the principal application while Origin[csp/application/record]Id remain unchanged.

originCspId: string *

originApplicationId: string *

originRecordId: string *

cspId: string *

applicationId: string *

recordId: string *

dateTime: string * (dateTime)
timestamp of the record emission

url: string (url)
reference to the emmitted record based on [csp/application/record]Id. (see SAD)

}

IntegrationDataType

IntegrationDataType string

Enum:

Array[9]

0: "event"
1: "threat"
2: "incident"
3: "vulnerability"
4: "artefact"
5: "chat"
6: "file"
7: "contact"
8: "trustCircle"

SharingParams

SharingParams {

only one of the trustCircleId or teamId should exist. If none exists, sharing is done based on data specific trust circles

toShare: boolean *

isExternal: boolean *

trustCircleId: string

teamId: string

}