API for Data Import Automation v1 (Administration)

Overview

An API to post data to channels for [Data Import Automation](#). This API is an optional feature activated with the [license](#).

API General Info

Response Content Type

It is possible to switch between `json` and `xml` response types by setting an `Accept` header to either `application/json` or `application/xml` respectively.

Successful Responses

Response Codes

- **200** - The request was executed successfully and the server returned some data. The return content type is specified inside a response header called `content-type`. Example content types are `application/xml`, `application/json`, `application/zip` and `text/plain`.
- **204** - The request was executed successfully and the server returned no data. If the `Location` header was present then the server responded with a redirect.

Error Responses

If the API call resulted in an error then the error code will be returned along with the error object in the response body. The error object contains the attributes listed below:

- **type** - type of the error
- **cause** - the root cause of the error
- **message** - message containing details of the related error
- **uuid** - a unique identifier of the error which also may appear in the exception logs
- **timestamp** - the time of the error

Note: For chunked responses, the socket will be closed prematurely if an error occurs after the initial chunk containing the status code has been sent.

Response Codes

- **400** - The request contains invalid parameters or there are invalid settings. Consult the error message and/or check Administration for invalid settings.
- **401** - The user is not authenticated.
- **403** - The user is not authorized access to the specified endpoint.
- **404** - The specified endpoint does not exist or a request parameter is invalid. Consult the error message.
- **405** - The specified endpoint does not support the requested operation. Consult the error message.
- **409** - The specified endpoint is in a state incompatible with the requested operation. Consult the error message.
- **500** - Unknown error occurred while processing the request. Consult the CPQ logs for more information.
- **504** - The specified concurrent calls limit was exceeded and the server was not able to serve the response in a specified timeout limit. That error is always served with the xml body containing the description.

Endpoints by Category

Channel data verification

GET /api/external-channels/v1/[channel]/incomplete-module-variants?from=[from]&to=[to]

Gets incomplete variants sent within specified time interval.

Parameter	Type	Description
channel	String	Channel name
from	String (Optional)	From date
to	String (Optional)	To Date
Result	IncompleteModuleVariants	A list of incomplete variants.

Channel job results

GET /api/external-channels/v1/[channel]/job-status/[jobid]

Gets the status of the job in a channel by its id.

Parameter	Type	Description
channel	String	
jobid	String	

Parameter	Type	Description
Result	PackageJobStatus	

Sending packages

POST /api/external-channels/v1/[channel]/module-variants?body=[body]

Queues Module Variant data to a channel.

Parameter	Type	Description
channel	String	Channel name
body	Package	Package object
Result	PackageJob	PackageJob containing the Id of the job.

POST /api/external-channels/v1/[channel]/direct-costs?body=[body]

Queues direct costs of module variants to a channel.

Parameter	Type	Description
channel	String	Channel name
body	Package	Package object
Result	PackageJob	PackageJob containing the Id of the job.

POST /api/external-channels/v1/[channel]/master-data-objects?body=[body]

Queues Master Data Object instances to a channel.

Parameter	Type	Description
channel	String	Channel name
body	Package	Package object
Result	PackageJob	PackageJob containing the Id of the job.

POST /api/external-channels/v1/[channel]/assembly-variants?body=[body]

Queues Assembly Variant data to a channel.

Parameter	Type	Description
channel	String	Channel name
body	[Package](#)	Package object
Result	[PackageJob](#)	PackageJob containing the Id of the job.

POST /api/external-channels/v1/[channel]/cost-lists/element?body=[body]

Queues Element Cost List items to a channel.

Parameter	Type	Description
channel	String	Channel name
body	[Package](#)	Package object
Result	[PackageJob](#)	PackageJob containing the Id of the job.

POST /api/external-channels/v1/[channel]/cost-lists/sub-element?body=[body]

Queues Sub-element Cost List items to a channel.

Parameter	Type	Description
channel	String	Channel name
body	[Package](#)	Package object
Result	[PackageJob](#)	PackageJob containing the Id of the job.

POST /api/external-channels/v1/[channel]/cost-lists/variant?body=[body]

Queues Variant Cost List items to a channel.

Parameter	Type	Description
channel	String	Channel name
body	[Package](#)	Package object
Result	[PackageJob](#)	PackageJob containing the Id of the job.

Data Types by Category

Channel data verification

IncompleteModuleVariants

JSON   XML

```json
{
  "variants" : [
    {
      "id" : "",
      "module" : "",
      "modifiedTime" : "",
      "hasInvalidCostData" : false,
      "missingCostData" : [
        {
          "key" : {
                  }
        },
        ...
      ],
      "missingFeatures" : [
        "",
        ...
      ]
    },
    ...
  ]
}
```

Channel job results

PackageJobStatus

The status of the job that puts queued data to a channel.

Field "status" can be one of the following: "Received", "Running", "Failed", "Finished".

JSON   XML

```json
{
  "output" : "",
  "jobId" : "",
  "status" : ""
}
```

Sending packages data types

Package

Package data. items: objects that are sent to a DIA channel. props: additional properties that can be sent together with the data and used in scripts.

```
                                                           JSON    XML
{
  "props" : {
    "key" : ""
  },
  "items" : [
    {
      "properties" : [
        {
          "name" : "",
          "values" : [
            {
              "text" : ""
            },
            ...
          ]
        },
        ...
      ]
    },
    ...
  ]
}
```

PackageJob

Result of sending a package

```
                                                           JSON    XML
{
  "jobId" : ""
}
```