Contents

Release notes - API Selector

Data Connect provides a beta API Selector enabled on Ethos fittings on Integration Designer.

Ethos Proxy Get fitting

Use this fitting to perform an HTTP operation to get an Ethos resource through Ethos Integration.

Expected input

A message payload that is a JavaScript object.

Reference configuration

Configuration	Description
Resource	Required. Ethos resource, for example `persons`. Enter the resource name here or click **Select API** to use the API Selector to find the resource. See Find an API with the API Selector on page 15. When selecting the GET method in the API Selector, if there are multiple GET methods displayed, select the method with the path that does not include `{id}`.
Accept Versions	Required. List of supported resource version numbers. Each version will be tried in order until a response code of 406 is not returned. Up to 10 versions can be specified; additional versions will be ignored. If you use the API Selector, this field automatically populates. You can enter in multiple accept versions in the fittings. However, you can not select multiple versions from the API Selector. You can still manually type in additional API versions.
Alternative Representation	Optional. Alternative representation
Id from Payload	Optional. JSONPath to id within payload itself. Mutually exclusive from `Id from Context`. Ex. "$.person.id".

Configuration	Description
Id from Context	Optional. JSONPath to id within context. Mutually exclusive from `Id from Payload`, for example "$.guid".
Target	Optional. Dot-notation target within payload to put Ethos response. When not set it replaces the whole message payload.
Result Headers	Optional. Dot-notation target within message to put response headers. When not set headers will be dropped silently. Valid examples: `theHeaders`, `payload.theHeaders`, `header.theHeaders`, `message.payload.theHeaders`, `message.header.theHeaders`. Note that if the response is an array, the headers will be saved into `message.header.propName` even if payload was specified as the place to save them. Repeated headers will be stored in an array value.
Cache	Optional. Flag to enabled or disable response caching, defaults to false. When successful, the response will be cached using the resource or resource + id used. Cached information will expire after one hour. Please note that the cache can be populated or retrieved by other pipelines using the matching resource, resource id and method (GET).
Ignore Errors	Optional. Flag that when set to true, it disables error reporting to Ethos when an error is caught. Use this in conjunction with `errorPath` to implement custom error handling further in the pipeline.
Error Path	Optional. Dot-notation error path. Setting this will push original message forward with error details. Error path can specify a location in header or payload. When no location is specified it defaults to payload, in other words: "error" is the same as "payload.error".
Api Key	Optional. Name of a pipeline parameter which will contain the the API key to be used by this fitting. When not specified `ethosApiKey` will be used.

Example

The following is an example of how to use Ethos Proxy Get:

1. In order for this fitting to work, an Ethos API Key should be created as a parameter. Click over the dropdown menu aside the pipeline name, select **Edit Pipeline Settings**, and add a

new pipeline parameter named *ethosApiKey*. The name is reserved so it will always pick that parameter unless a different option is selected as the API key in the fitting configuration.

2. Enter the following fitting configuration:
    - Fitting Name: Ethos Proxy Get
    - Resource: persons (in lowercase)
    - Accept Versions: 12
    - Get Id From: Context
    - Id From Context: $.personGuid
3. For this example, an optional string pipeline parameter named `personGuid` will be defined. This parameter is going to be used by the fitting as the ID of the person.
4. Run the test. Use a valid Ethos Api Key and the persons GUID or ID. This will retrieve the information of the person with the specified ID.

Output

A new message with the same header as the input and with a payload that is the response. If the response is JSON, it will be parsed into a JavaScript object.

If neither the `idFromPayload` nor the `idFromContext` properties are specified the resource is queried without ID as a get all. The same happens if the `personGuid` value is not provided.

When specifying the `idFromPayload` or `idFromContext` configuration properties but provided with a JSONPath syntax that is not valid, the Ethos resource isn't reached, the original input message passes along to the next fitting and an error gets reported.

Ethos Proxy Get Array fitting

Use this fitting to perform an HTTP operation to get an Ethos array of resources through Ethos Integration.

This fitting only supports an array for a single resource, such as an array of addresses. The intended use is not to get an array of different Ethos resources.

Expected input

A message payload that is a JavaScript object containing an array of an Ethos resource.

Reference configuration

Configuration	Description
Resource	Required. Ethos resource, for example, `addresses`. Enter the resource name here or click **Select API** to use the API Selector to find the resource. See Find an API with the API Selector on page 15. When selecting the GET method in the API Selector, if there are multiple GET methods displayed, select the method with the path that does not include `{id}`.
Accept Versions	Required. List of supported resource version numbers. If you use the API Selector, this field automatically populates. You can enter in multiple accept versions in the fittings. However, you can not select multiple versions from the API Selector. You can still manually type in additional API versions.
Alternative Representation	Optional. Alternative representation.
Id from Payload	Optional. JSONPath to id list within payload itself, for example, `$.person.addresses[*].address.id`.
Json Path Target	Required. JSONPath target within payload to put Ethos response, for example, `$.person.addresses[*].address`.
Response Id Field	Optional. Property name from response to compare for target replacement, defaults to "id".
Cache	Optional. Flag to enabled or disable response caching, defaults to false.
Ignore Errors	Optional. flag that when set to true, it disables error reporting to Ethos when an error is caught. Use this in conjunction with `errorPath` to implement custom error handling further in the pipeline.
Error Path	Optional. Dot-notation error path. Setting this will push original message forward with error details. Error path can specify a location in header or payload. When no location is specified it defaults to payload, in other words: "error" is the same as "payload.error".

Configuration	Description
Api Key	Optional. Name of a pipeline parameter which will contain the the API key to be used by this fitting. When not specified `ethosApiKey` will be used.

Example

The following is an example of how to use Ethos Proxy Get Array

1. In order for this fitting to work, an Ethos Api Key should be created as a parameter. Click over the dropdown menu aside the pipeline name, select **Edit Pipeline Settings**, and add a new pipeline parameter named *ethosApiKey*. The name is reserved so it will always pick that parameter unless a different option is selected as the Api Key in the configuration.
2. Enter the following configuration:

- Resource: `persons`
- Enter Accept Versions: `12` (Click the **+** button after entering this parameter)
- Id From Payload: `$.persons[*].person.id`
- Json Path Target: `$.persons[*].person`

1. The input should contain IDs for persons. The input data should have a JSON that looks like this:

```
{
  "persons": [
    {
      "person": {
        "id": "invalid-id"
      }
    },
    {
      "person": {
        "id": "268a03c6-56c7-462f-8479-d4ebb07984ad"
      }
    }
    ,
    {
      "person": {
        "id": "989b7725-a875-44b8-9195-e67b09bd1b3f"
      }
    }
  ]
}
```

Response is returned for that resource matching (by ID) the first item in the persons array in the payload, and inserting the response replacing that ID with the response payload. The unmatched ID remains unchanged.

Ethos Proxy Get With Filter fitting

Use this fitting to perform an HTTP operation to get an Ethos array of resources through Ethos Integration with optional filter criteria. The filter criteria is specified in the URL of the request and is usually some ID (GUID) value.

Expected input

A message payload that is a JavaScript object containing some data in the message payload used by the criteria filter.

Reference configuration

Configuration	Description
Resource	Required. Ethos resource, for example `persons`. Enter the resource name here or click **Select API** to use the API Selector to find the resource. See Find an API with the API Selector on page 15. When selecting the GET method in the API Selector, if there are multiple GET methods displayed, select the method with the path that includes `{id}`. If you are using a QAPI, you must manually enter the resource name and enable Query By Post. You cannot use the API Selector to find a QAPI. To do a POST for a QAPI, you must choose the GET option and enable the QAPI toggle. This will turn the request into a POST.
Filter	Optional. Querystring criteria to be used along specified `resource`. It can contain mustache type reference tags, for example, ?criteria={"id":"{{person.id}}"}
Page Size	Optional. Limits the result set size, for example 10
Payload Target Path	Optional. Dot-notation target within payload to put Ethos response. When not set it replaces the whole message payload, for example person.results.

Configuration	Description
Result Headers	Optional. Dot-notation target within message to put response headers. When not set headers will be dropped silently. Valid examples: `theHeaders`, `payload.theHeaders`, `header.theHeaders`, `message.payload.theHeaders`, `message.header.theHeaders`. Note that if the response is an array, the headers will be saved into `message.header.propName` even if payload was specified as the place to save them. Repeated headers will be stored in an array value.
Accept Versions	Required. List of supported resource version numbers. If you use the API Selector, this field automatically populates. You can enter in multiple accept versions in the fittings. However, you can not select multiple versions from the API Selector. You can still manually type in additional API versions.
Alternative Representation	Optional. Alternative representation.
Cache	Optional. Flag to enabled or disable response caching, defaults to false.
Query By Post	Optional. If set, the URL will be /qapi instead of /api, and any `filter` configuration will be ignored.
Content Version	Required. If `queryByPost` is set. Version for Ethos Resource, for example 12. Ignored otherwise. If you use the API Selector, this field automatically populates.
Body Path	Optional. Used if `queryByPost` is true. A JSON path in the payload which will be used as the body of the post. If set but the path returns undefined, no operation will be done and message will be pushed through.
Ignore Errors	Optional. Flag that when set to true, it disables error reporting to Ethos when an error is caught. Use this in conjunction with `errorPath` to implement custom error handling further in the pipeline.

Configuration	Description
ErrorPath	Optional. Dot-notation error path. Setting this will push original message forward with error details. Error path can specify a location in header or payload. When no location is specified it defaults to payload, in other words: "error" is the same as "payload.error".
Api Key	Optional. Name of a pipeline parameter which will contain the the API key to be used by this fitting. When not specified `ethosApiKey` will be used.

Example

The following is an example of how to use Ethos proxy Get with filter:

1. In order for this fitting to work, an Ethos API Key should be created as a parameter. Click over the dropdown menu aside the pipeline name, select **Edit Pipeline Settings**, and add a new pipeline parameter named *ethosApiKey*. The name is reserved so it will always pick that parameter unless a different option is selected as the API key in the fitting configuration.
2. Enter the following fitting configuration:

- Resource: `persons`
- Filter: `?criteria={"names":[{"firstName":"james"}]}`
- Accept Versions: `12` (Click the **+** button after entering this parameter)

1. Run the test. This retrieves all the persons that match with the given filter criteria.

Output

A new message with the same header as the input and with a payload containing the response. This fitting supports paging. This fitting will determine if paging is needed based on the x-total-count response header, the page size (if specified), and the response body length.

With Paging

If paging is needed, and the payload target path is specified, then the paged response(s) will be concatenated into an array and the array will be inserted into the message payload at the location specified by the `payloadTargetPath` property in the fitting configuration. The message will then be pushed through this fitting. It should be understood that the `payloadTargetPath` property is not intended to be used with a high volume of data from many paged responses to prevent running out of memory.

If the `payloadTargetPath` is not specified, then each paged response will be pushed through this fitting as a separate message.

The page size can also be specified in the fitting configuration. Each paging request will use a limit criteria equal to the page size. If the page size is not specified, the default page size for the given resource as determined by the Ethos resource API is used.

Without Paging

If paging is not needed and the `payloadTargetPath` configuration property is specified, the response will be inserted into the message payload at the location specified by the `payloadTargetPath`, and the message will be pushed through this fitting.

If paging is not needed and the `payloadTargetPath` is not specified, then the response will replace the message payload, and the message will be pushed through this fitting.

Query by Post

To use an Ethos QAPI endpoint, set `queryByPost` to `true` and also specify `contentVersion`. The payload of the message coming into this fitting becomes the body of the outgoing QAPI call. Paging is supported on the QAPI response.

Ethos Proxy Post fitting

Use this fitting to perform an HTTP operation to post an Ethos resource through Ethos Integration.

Expected input

A message payload that is a JavaScript object.

Reference configuration

Configuration	Description
Resource	Required. Ethos resource, for example `persons`. Enter the resource name here or click **Select API** to use the API Selector to find the resource. See Find an API with the API Selector on page 15. You will see both /api and /qapi returned from the API Selector. QAPIs are not supported from Ethos Proxy Post at this time.
Content Version	Required. Version for Ethos resource, for example 12. If you use the API Selector, this field automatically populates.

Configuration	Description
Accept Version	Required. Version for Ethos resource, for example 12. If you use the API Selector, this field automatically populates. You can enter in multiple accept versions in the fittings. However, you can not select multiple versions from the API Selector. You can still manually type in additional API versions.
Alternative Content Representation	Optional. Alternative content representation.
Target	Optional. Dot-notation target within payload to put Ethos response. When not set it replaces the whole message payload.
Body Path	Optional. A JSON Path in the payload which will be used as the body of the post. If set but the path returns undefined, no operation will be done and the message will be pushed through.
Ignore Errors	Optional. Flag that when set to true, it disables error reporting to Ethos when an error is caught. Use this in conjunction with `errorPath` to implement custom error handling further in the pipeline.
Error Path	Optional. Dot-notation error path. Setting this will push original message forward with error details. Error path can specify a location in header or payload. When no location is specified it defaults to payload, in other words: "error" is the same as "payload.error".
Api Key	Optional. Name of a pipeline parameter which will contain the the API key to be used by this fitting. When not specified `ethosApiKey` will be used.

Example

The following is an example of how to use Ethos Proxy Post:

1. In order for this fitting to work, an Ethos API key should be created as a parameter. Click over the dropdown menu aside the pipeline name, select **Edit Pipeline Settings**, and add a new pipeline parameter named ethosApiKey. The name is reserved so it will always pick that parameter unless a different option is selected as the API key in the configuration.
2. Configure the fitting by adding a valid resource, content and accept versions. For example:

- Resource: `instructional-methods`
- Content Version: `6`
- Accept Version: `6`

1. It will be necessary to add some input data for this example:

```
{
  "abbreviation": "RD1",
  "title": "Recitation/Discussion",
  "id": "00000000-0000-0000-0000-000000000000"
}
```

Output

A new message with the same header as the input and with a payload that is the response. If the response is JSON, it will be parsed into a JavaScript object.

Ethos Proxy Put fitting

Use this fitting to perform an HTTP operation to put an Ethos resource through Ethos Integration.

Expected input

A message payload that is a JavaScript object.

Reference configuration

Configuration	Description
Resource	Required. Ethos resource, for example `persons`. Enter the resource name here or click **Select API** to use the API Selector to find the resource. See Find an API with the API Selector on page 15.
Content Version	Required. Version for Ethos resource, for example 12. If you use the API Selector, this field automatically populates.
Accept Version	Required. Version for Ethos resource, for example 12. If you use the API Selector, this field automatically populates. You can enter in multiple accept versions in the fittings. However, you can not select multiple versions from the API Selector. You can still manually type in additional API versions.

Configuration	Description
Alternative Content Representation	Optional. Alternative content representation.
Id From Payload	Required (if no `Id from Context` is defined). JSON Path to id within payload itself. Mutually exclusive from `Id From Context`. For example, `$.person.id`.
Id From Context	Required (if no `Id from Payload` is defined). JSON Path to id within context. Mutually exclusive from `Id From Payload`. For example, `$.guid`.
Target	Optional. Dot-notation target within payload to put Ethos response. When not set it replaces the whole message payload.
Body Path	Optional. A JSON Path in the payload which will be used as the body of the request. If set but the path returns undefined, no operation will be done and the message will be pushed through.
Ignore Errors	Optional. Flag that when set to true, it disables error reporting to Ethos when an error is caught. Use this in conjunction with `errorPath` to implement custom error handling further in the pipeline.
Error Path	Optional. Dot-notation error path. Setting this will push original message forward with error details. Error path can specify a location in header or payload. When no location is specified it defaults to payload, in other words: "error" is the same as "payload.error".
Api Key	Optional. Name of a pipeline parameter which will contain the the API key to be used by this fitting. When not specified `ethosApiKey` will be used.
Allow concurrent	Optional. If set, allows multiple threads to send PUT requests concurrently for each input message.

Output

A new message with the same header as the input and with a payload that is the response. If the response is JSON, it will be parsed into a JavaScript object.

When specifying the `idFromPayload` or `idFromContext` configuration properties but provided with JSONPath syntax that is not valid, the Ethos resource is not reached, the original input message passes along to the next fitting and an error gets reported.

Ethos Proxy Put fitting example

For this fitting to work, start by creating an Ethos API Key as a parameter.

Procedure

1. Near the pipeline name, click the drop-down menu and select **Edit Pipeline Settings**.
2. Add a new pipeline parameter named **ethosApiKey**. The name is reserved so it will always pick that parameter unless a different option is selected as the API key in the configuration.
3. Configure the fitting. For example:

Field	Example selection or entry
Resource	persons
Content Version	12
Accept Version	12
Get Id From	Payload
Id From Payload	$.student.id

Find an API with the API Selector

The API Selector provides an intuitive means to view and select data returned by an API, so that the data can be used in other applications.

About this task

With the API Selector, you can search for an API based on your existing knowledge about Ellucian's APIs or the source systems that the APIs pull from. The API Selector can be accessed from Ellucian solutions such as Data Connect to populate those solutions with data returned through an API.

The API Selector searches the following APIs:

- Ellucian-delivered APIs including Colleague Ethos APIs, Banner Ethos APIs, Banner Business Process APIs, and other APIs exposed through Ellucian solutions. Your search is limited to APIs that are relevant to your implementation - for example, Colleague customers will not see Banner APIs in the search results.
- Data Connect Serverless APIs developed for your institution.
- APIs generated by Banner Custom Schemas for your institution.

Procedure

1. Access the API Selector from the application that will use the data. For example, that application might include a **Select API** link.

   The **Search API Catalog** page is displayed.
2. (Optional) In the drop-down list to the left of the search box, select a category to limit the search, or accept the default (All) to search all categories.

Category	Description	Example search term
Name	Name of the API	
Source	Originating system of the API	`Banner` or `Colleague`
Database Object	Data model or data source in the originating system	Banner/Colleague table or column name
Response Property	Name of a property that is returned by the API	
Request Parameter	Name of a parameter that the API accepts	

3. In the search box, enter your search term.

   As you type, a list of up to ten APIs that match your search term is displayed below the search box.

   - If you see the desired API in the drop-down list, click that API name to go to the details page for that API.
   - If you don't see the desired API in the drop-down list:

     a. Press Enter to view a search results page with summary information about all APIs that match your search.

     b. (Optional) Use the filters at the top of the search results page to refine the search results.

     c. Click the name of an API to go to the details page for that API.
4. On the API details page, view detailed information about the API.

   The APIs are documented using the industry standard OpenAPI specifications and are displayed using Swagger.
5. In the **Version** drop-down, the latest version is selected by default. You should likely be using the latest version, which has the most up-to-date information. If the latest version does not have the fields you need to interact with your source system, review the other versions.
6. If this is the API that you want to use in your application, click the **SELECT** link for the desired method.

   For example, click **SELECT GET** if your application requires the GET method.

   The API Selector sends a JSON object, with metadata related to the selected method, back to your application.

Note: You can also click **Copy URL** ⧉ to copy the URL for the method, to be used outside of the requesting application (for example, if you want to test the call using a REST API client).