XpressAPI Developer's Guide

XpressAPI Swagger Interface

Overview

XpressAPI, our modern API delivery solution, delivers data in a **Swagger interface** built around the OpenAPI Specification that can help you consume RESTful APIs. XpressAPI is comprehensible to both developers and non-developers. Once you input your credentials into Swagger, you can immediately test requests using interactive menus, dropdowns, and free form fields without writing any code.*

Swagger also provides machine-readable JSON docs that adhere to OpenAPI Specification. This file describes the structure and provides a detailed description of the data. The JSON doc can be imported into standard API development tools to streamline your data deployment.

Communication with the API client is facilitated via the REST architecture style and design pattern. REST is a standardized design technique used to describe how a web service can be accessed. Using REST, you can make https requests to get data. **OpenAPI** is the specific protocol in XpressAPI that exposes data through REST. XpressAPI is delivered via **OpenAPI Version 3.0.1** and delivers responses in JSON format.

* Please note that Swagger is limited by your browser's memory and requires time to render requests in an interface. Standard RESTful clients may process your requests more quickly.

Credentials

Your credentials will be sent in a welcome email.

These credentials serve as both your Authentication as well as your Authorization credentials.

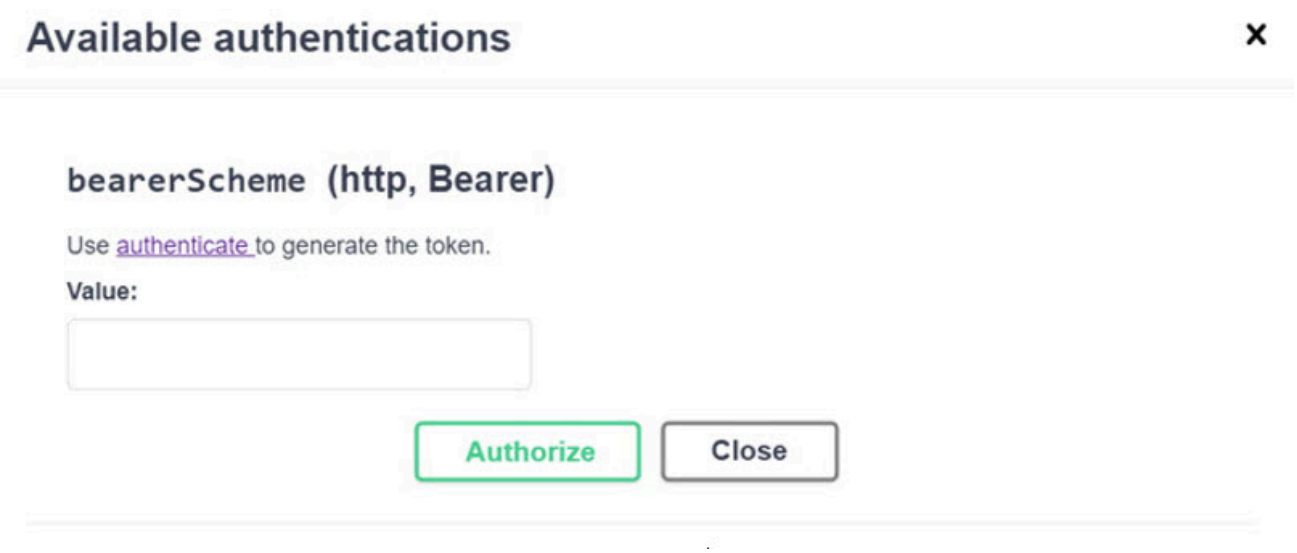
Your username will be provided along with a link to set up your password. Your username and password are confidential login credentials, please treat them as such. Please ascertain to apply your login credentials and track any login credential resets in all environments. In case you retain outdated or incorrect access credentials, you will receive a 423 - Locked Account error message after 5 unsuccessful login attempts.

To request your account to be unlocked or reset your password, reach out to our Enterprise API Support team at **support.api.mi@spglobal.com** and you will receive a password reset link via email.

To begin the authentication process, click the Authorize button 

on any dataset page.

The bearerScheme (i.e., Token-Based) authentication pop-up window appears.



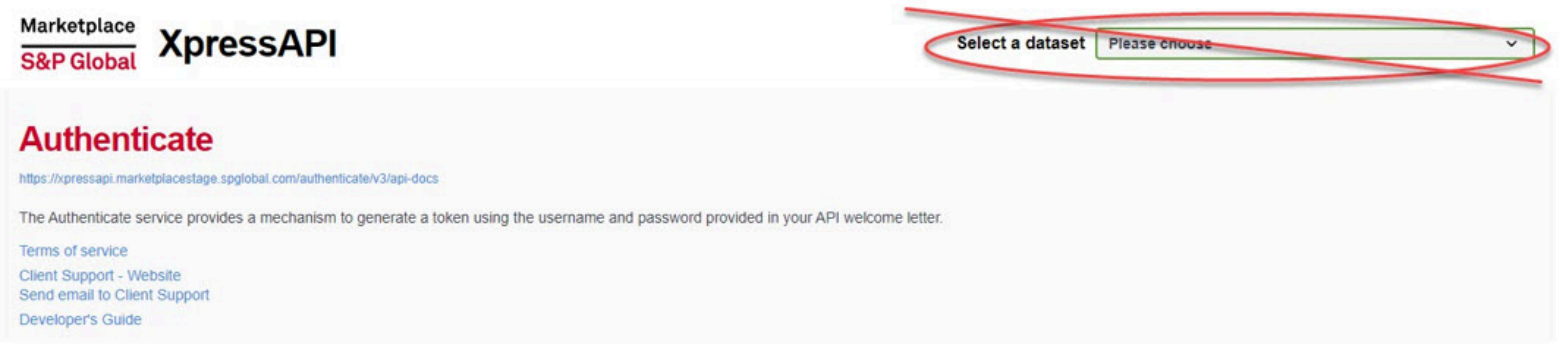
Token-Based Authentication in Swagger

Access Token

Token-based authentication is a protocol that generates encrypted security tokens. It enables you to verify your identity via a unique encrypted authentication token. A maximum of 4 concurrent threads can be submitted for requesting a token per second from an API account. Otherwise, you will receive a 429 – Too many requests error.

Click the authenticate link in the bearerScheme.

After you click the authenticate hyperlink, the Authenticate page appears. (This page does not appear in the Swagger dropdown menu.)

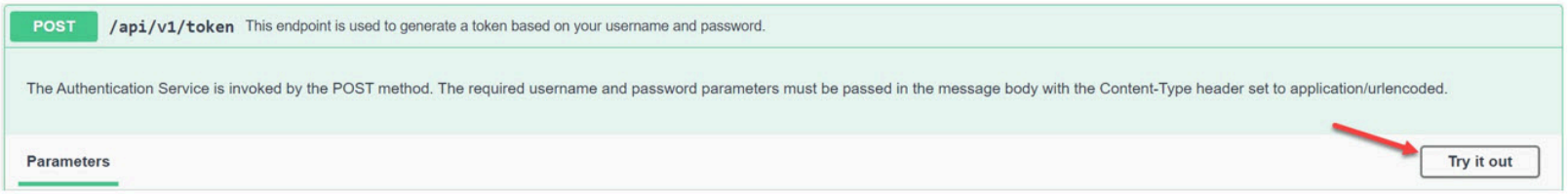


Token-Based Authentication in Swagger - POST Method

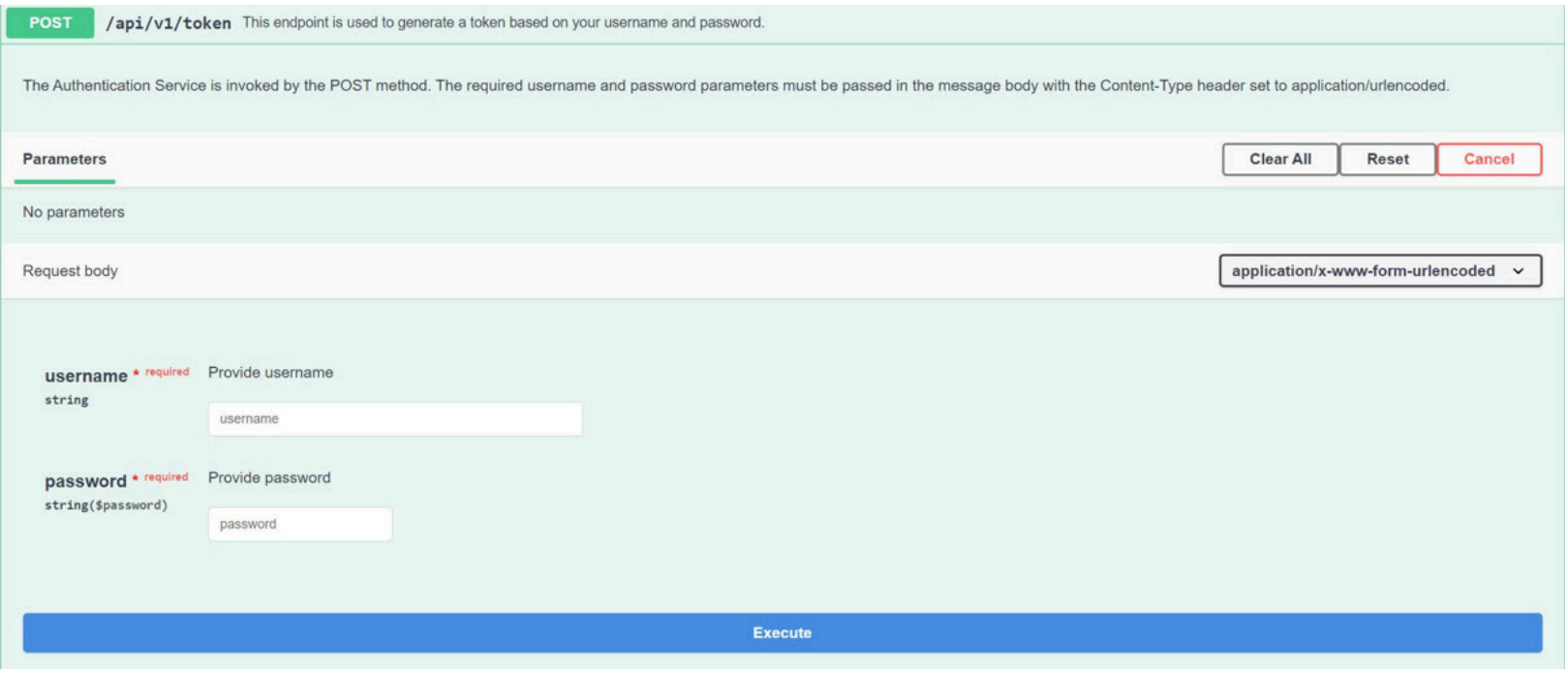
Access Token

You will need to generate an access token. The access token is valid for 3600 seconds (60 minutes).

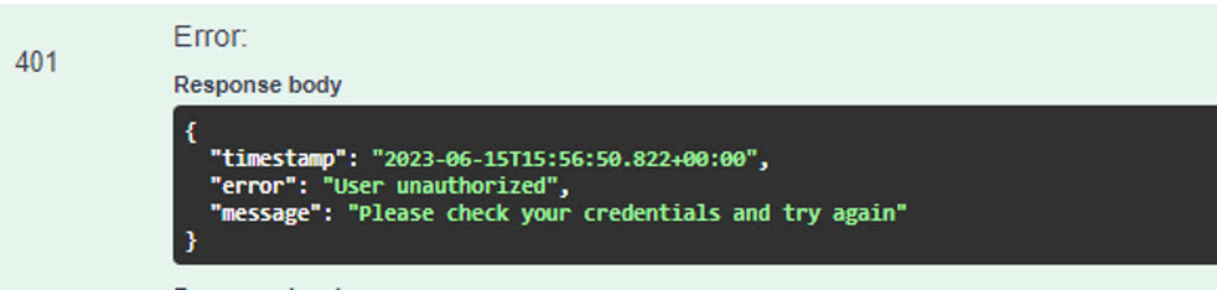
Using the POST method, first click the 'Try it out' button.



Then provide your username and password and then click the 'Execute' button.



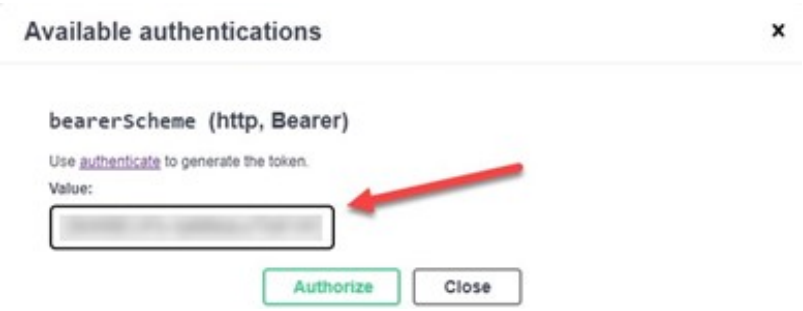
If your credentials are not authenticated, you will receive a 401 Error:



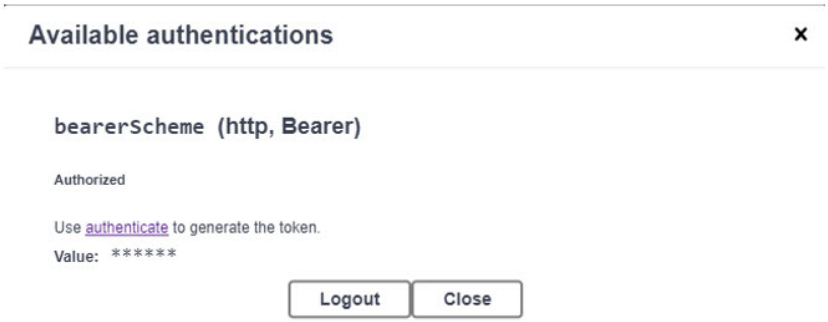
If your credentials have been authenticated, the response will contain the following:

- scope – The S&P Global platform where the token is applicable
- expires_in_seconds - Number of remaining seconds for token expiration
- token_type - Type of the token i.e., bearer
- access_token - Token used for authentication
- refresh_token - Token used for refreshing authentication

Paste the access_token in the Value field. Do not include the double quotes ("). Click the Authorize button.



A new window will appear.

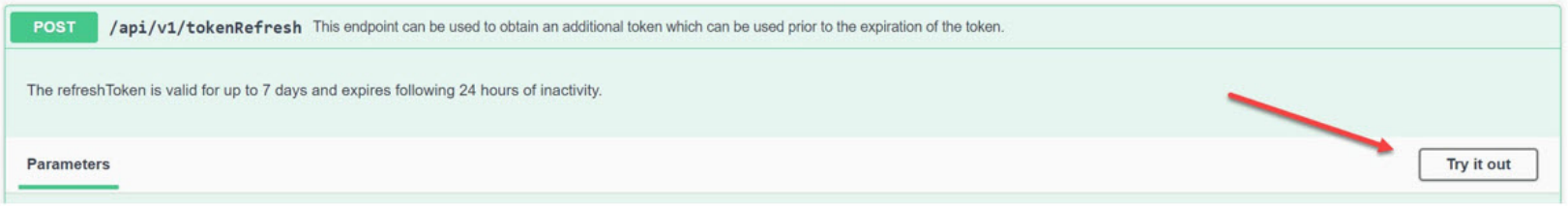


You are now ready to begin using XpressAPI. Navigate to your desired dataset page. (**IMPORTANT**: Your credentials still need to be authorized. XpressAPI will authorize your credentials after you make your initial request on your desired dataset page.)

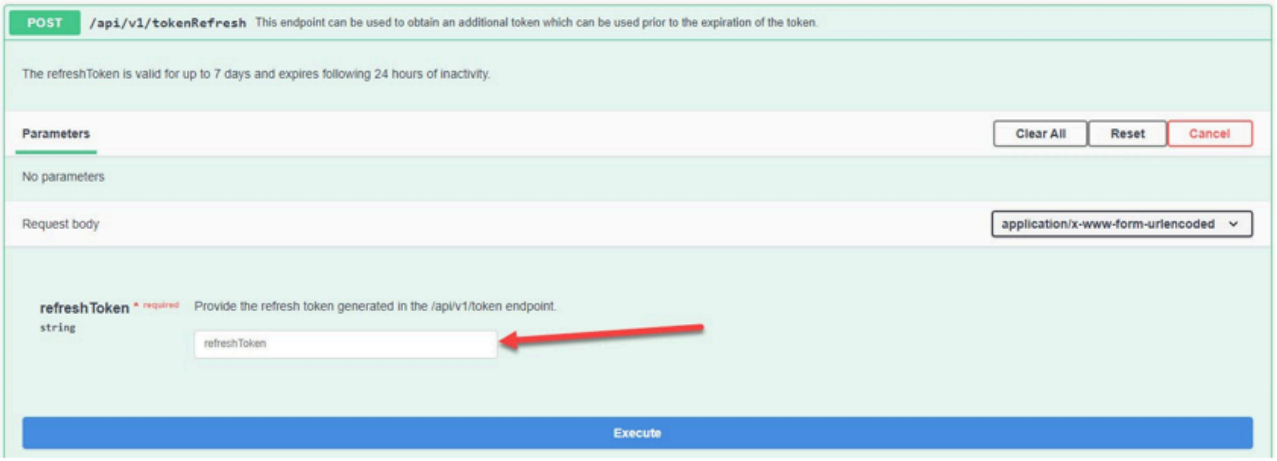
Refresh Token

A refresh token is a a token that can be used to refresh your access token prior to expiration. The XpressAPI refreshToken is valid for 7 days. If it is not used within 24 hours, it will expire. If a refresh attempt is made sooner than 3,600 seconds (60 minutes), the access token will be cached.

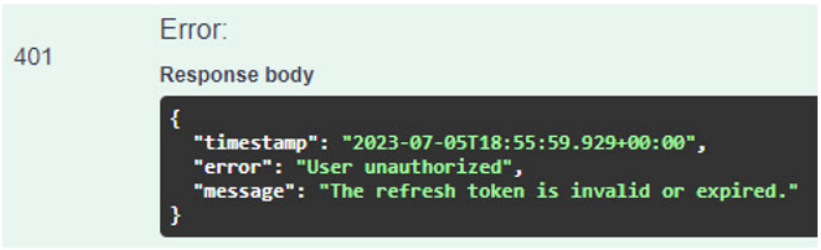
Using the POST method, click the 'Try it out' button.



Then paste the refresh token generated in the /api/v1/token endpoint. Do not include the double quotes (").



The refreshToken is valid for 7 days. If it is not used within 24 hours, it will expire.



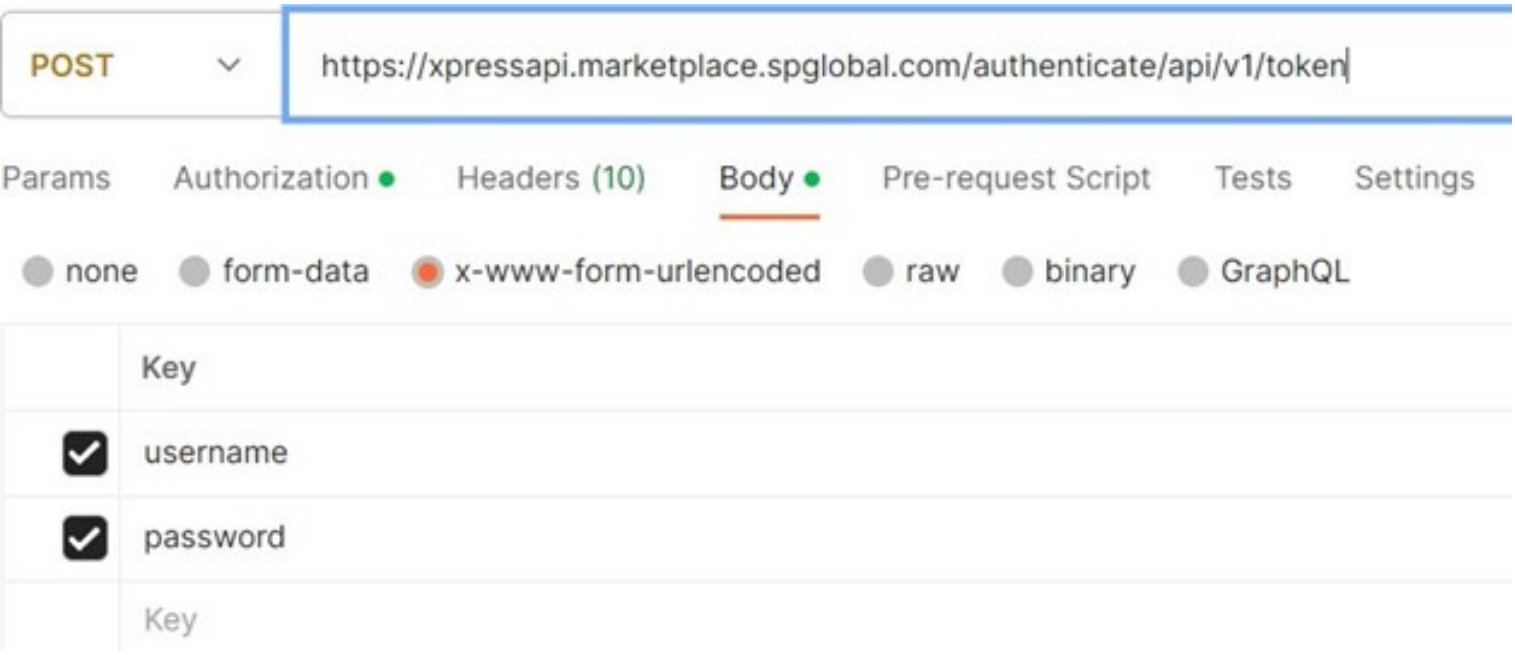
Token-Based Authentication Using a Rest Tool - POST Method

Access Token

You will need to generate an access token. The access token is valid for 3600 seconds (60 minutes).

The endpoint to generate an access token is: **https://xpressapi.marketplace.spglobal.com/authenticate/api/v1/token**

Calls to our endpoint must be authenticated using your username and password. Your username and password parameters must be passed in the message body with the Content-Type header set to application/urlencoded.



This will return an access token that needs to be appended to all subsequent requests. The access token must be added in the Request Header as an Authorization Bearer token.

The response will contain the following:

- scope – The S&P Global platform where the token is applicable
- expires_in_seconds - Number of remaining seconds for token expiration
- token_type - Type of the token i.e., bearer
- access_token - Token used for authentication
- refresh_token - Token used for refreshing authentication

If your credentials are not authenticated, you will receive a 401 Error.

Refresh Token

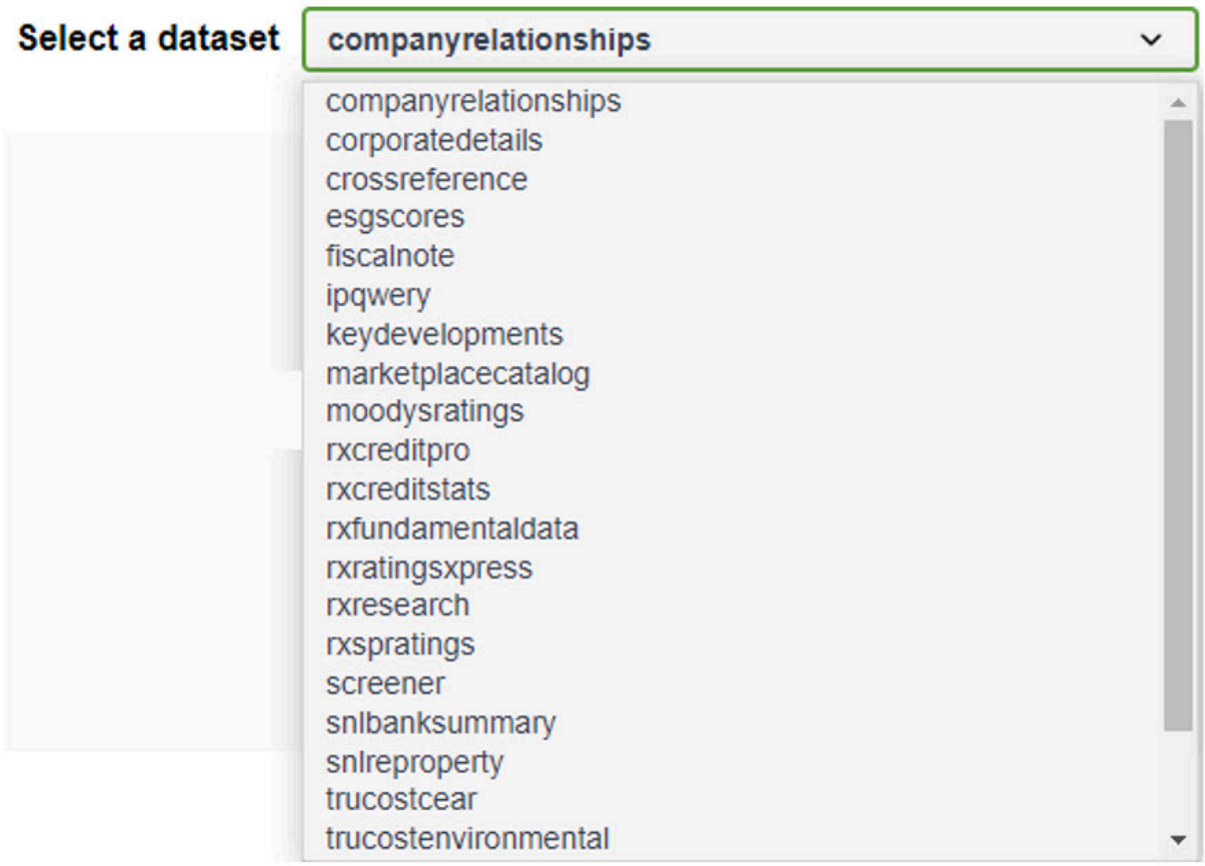
A refresh token is a special token that can be used to refresh your access token prior to expiration. The XpressAPI refreshToken is valid for 7 days. If it is not used within 24 hours, it will expire. If a refresh attempt is made sooner than 3,600 seconds (60 minutes), the access token will be cached.

The endpoint to generate a refresh token is: **https://xpressapi.marketplace.spglobal.com/authenticate/api/v1/tokenRefresh**

Provide the refresh token in the /api/v1/token endpoint.

Your Data Subscription

Your data subscription is comprised of endpoints as defined within individual **Swagger datasets**.*



Each Swagger dataset page displays:

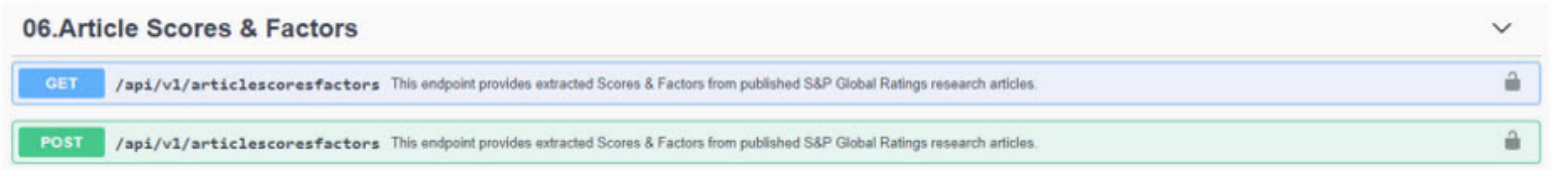
- all endpoints, request types, and parameters for that dataset
- the request URL and the corresponding cURL code associated with each request
- the response body
- potential error codes
- response payload structures

* Although you will be able to see endpoints for all datasets, you will only be authorized for datasets you are subscribed to. Some datasets require additional subscription levels (i.e., capabilities) to retrieve data.

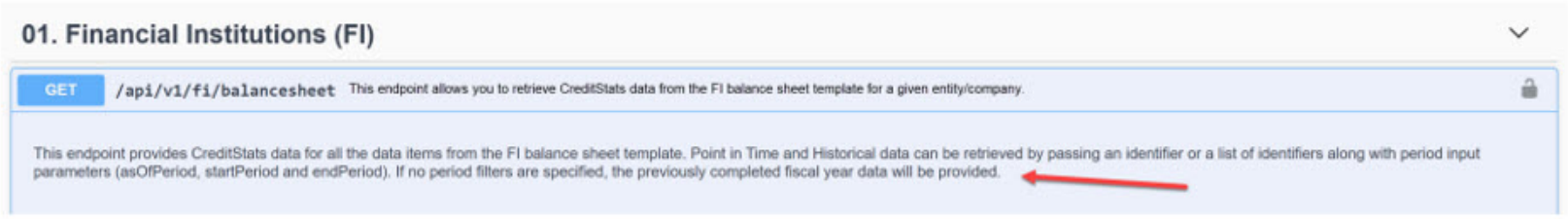
Using Swagger - Endpoints

In Swagger, dataset endpoints are clustered within logical groups with a similar business objective. Following is an example of three endpoint groups:

To expand (or collapse) a Swagger endpoint, click the endpoint. A brief description of the endpoint functionality appears immediately to the right of each GET and POST request.



Required filters and other pertinent details are displayed below the endpoint.



In a POST endpoint, additional details (e.g., mandatory filters and filter values) are provided.

POST `/api/v1/fi/incomestatement` This endpoint provides CreditStats data for all data items from the FI income statement template

The filters identifierType and identifierValues are mandatory.

identifierType: Refers to the type of identifier that uniquely identifies an entity or instrument. Available values: RXENTITYID, CUSIP6, CINS6.

identifierValues: Identifies is used to identify a unique entity or instrument. Please provide an identifier or a list of identifiers separated by a comma. You can pass up to 20 identifiers in one request

Point in Time and Historical data can be retrieved by passing an identifier or a list of identifiers along with period input parameters (**asOfPeriod, startPeriod and endPeriod**). Specify asOfPeriod to pull data as of a particular period (annual, semi-annual or quarterly) and range filters (startPeriod, endPeriod) to pull data between range of periods. If no period filters are specified, the previously completed fiscal year data will be provided.

The period filters are supported by filterValues in these period reference formats:

FYxxxx = Fiscal Year

CQ1Yxxxx to CQ4Yxxxx = First to Fourth Calendar Quarter

CH1Yxxxx to CH2Yxxxx = First to Second Calendar Half Year

FQ1Yxxxx to FQ4Yxxxx = First to Fourth Fiscal Quarter

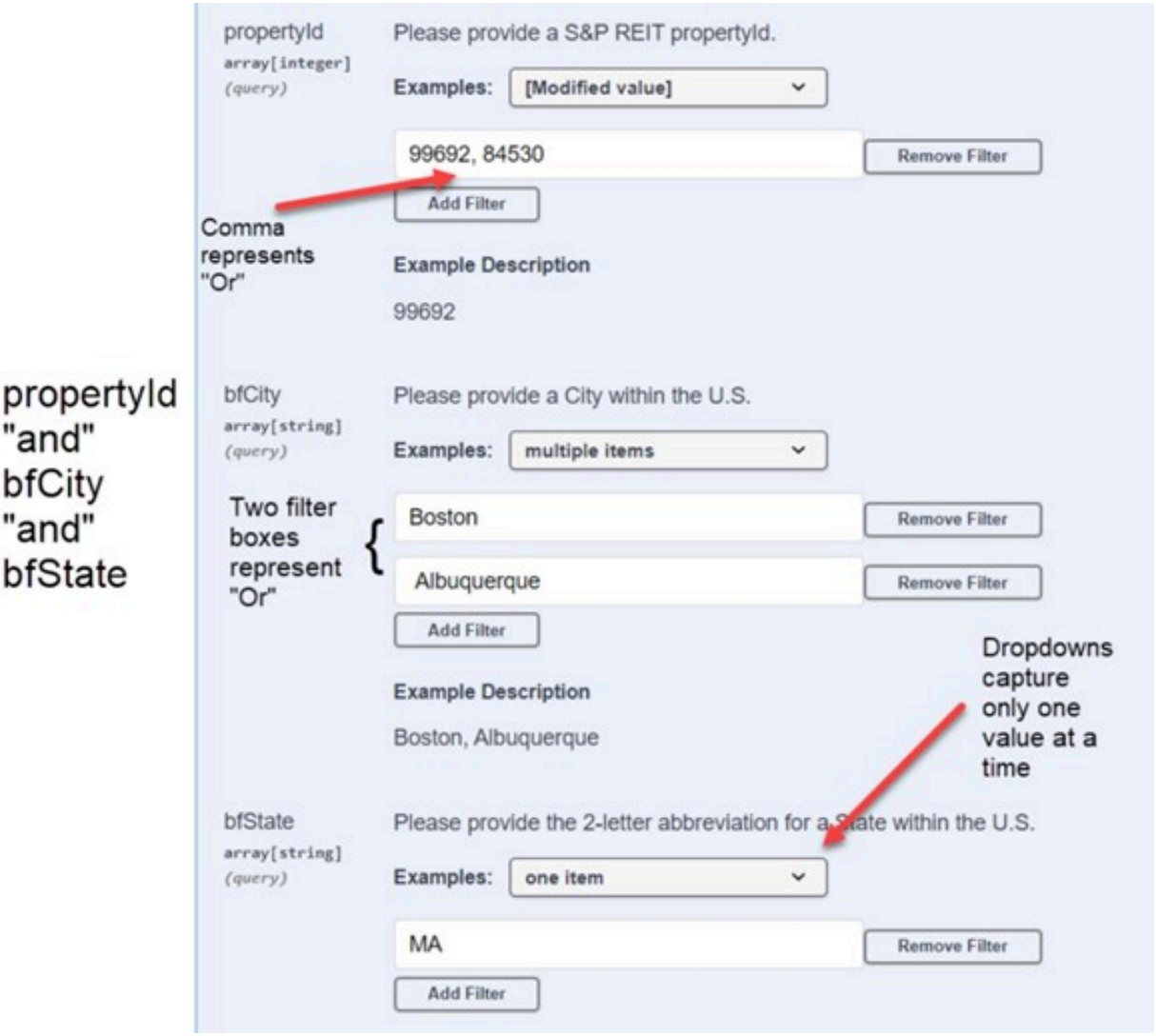
FH1Yxxxx to FH2Yxxxx = First to Second Fiscal Half Year

Replace xxxx with the year for which data is being requested for. Example CQ1Y2015.

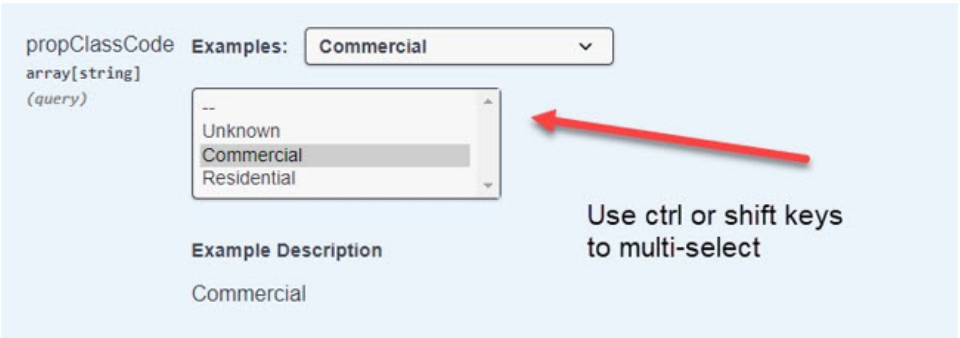
Using Swagger - Filters

In the Swagger interface, filters allow you to choose the query parameters (i.e., filter values) you wish to apply to your results.

- Filters applied to all query parameters in a single endpoint are considered “and”
- Filters applied to individual query parameters can be represented as “or” by using commas
- Dropdowns capture one query parameter at a time

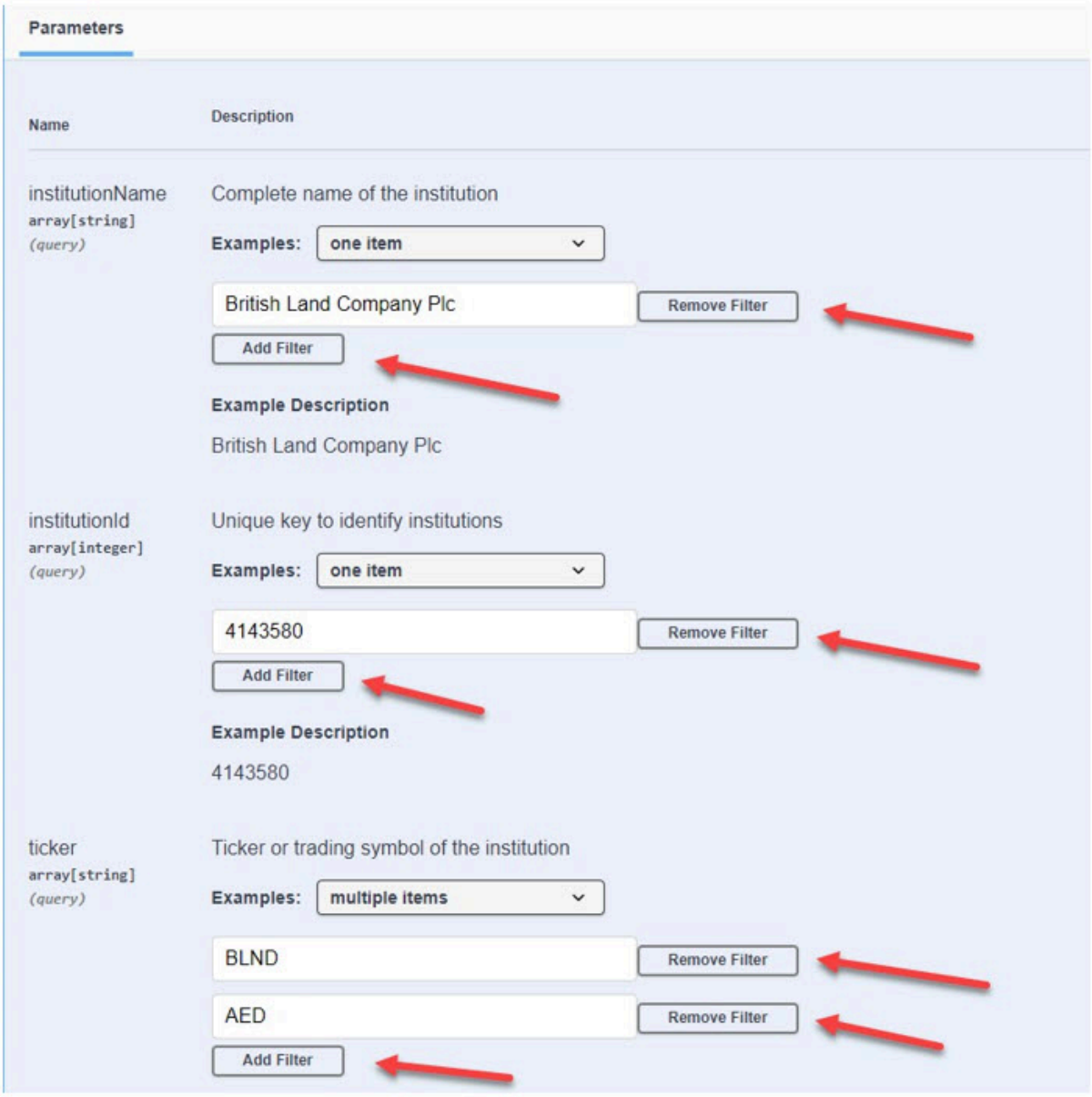


For scrolling lists, multi-select using control or shift keys.



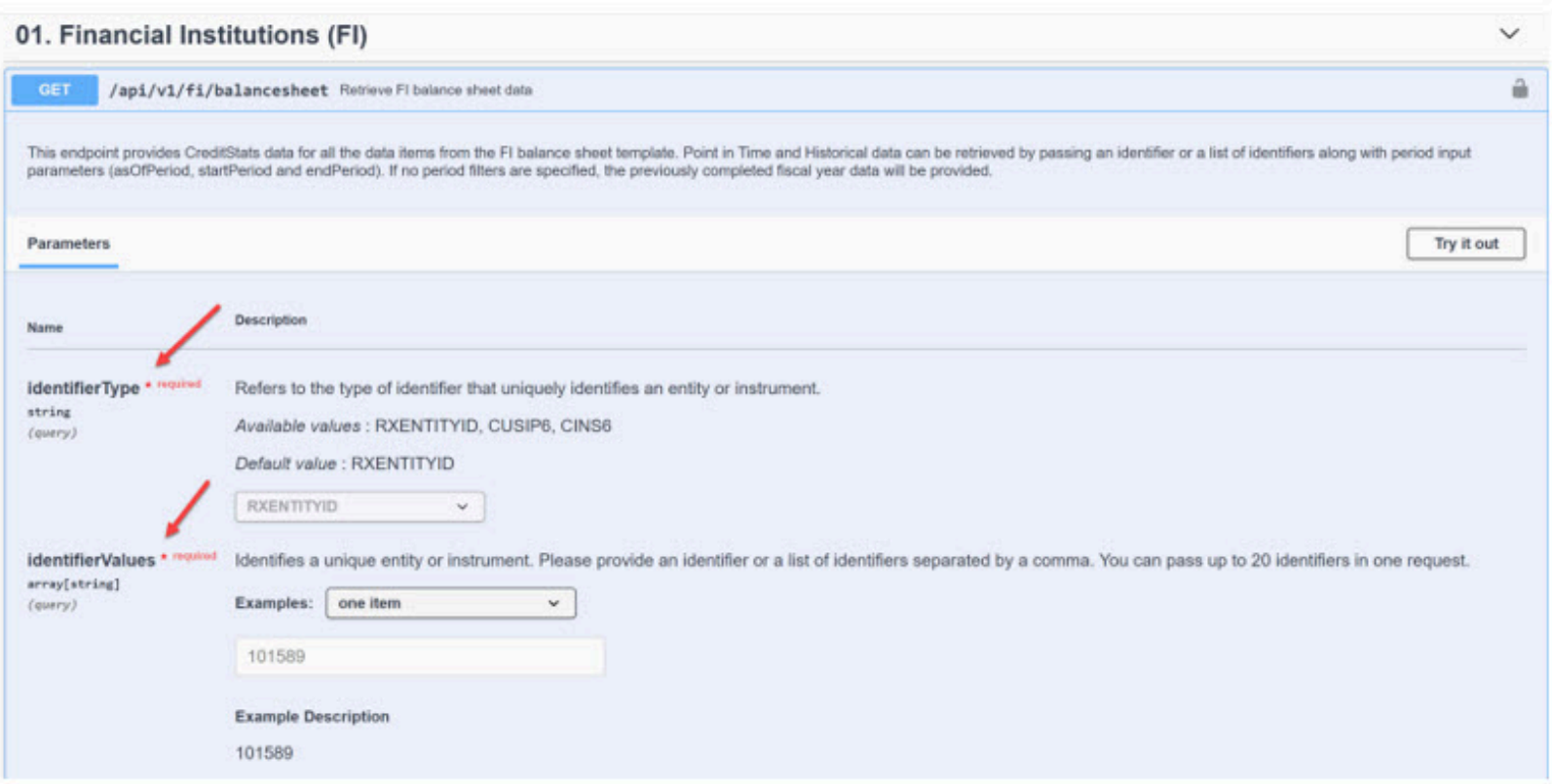
For dropdowns displayed in GET requests, you can multi-select filter values using control or shift keys. Not all scrolling lists allow multi-select.

Swagger endpoints are prepopulated with sample filter values. Click the **Remove Filter** or **Add Filter** buttons to revise.



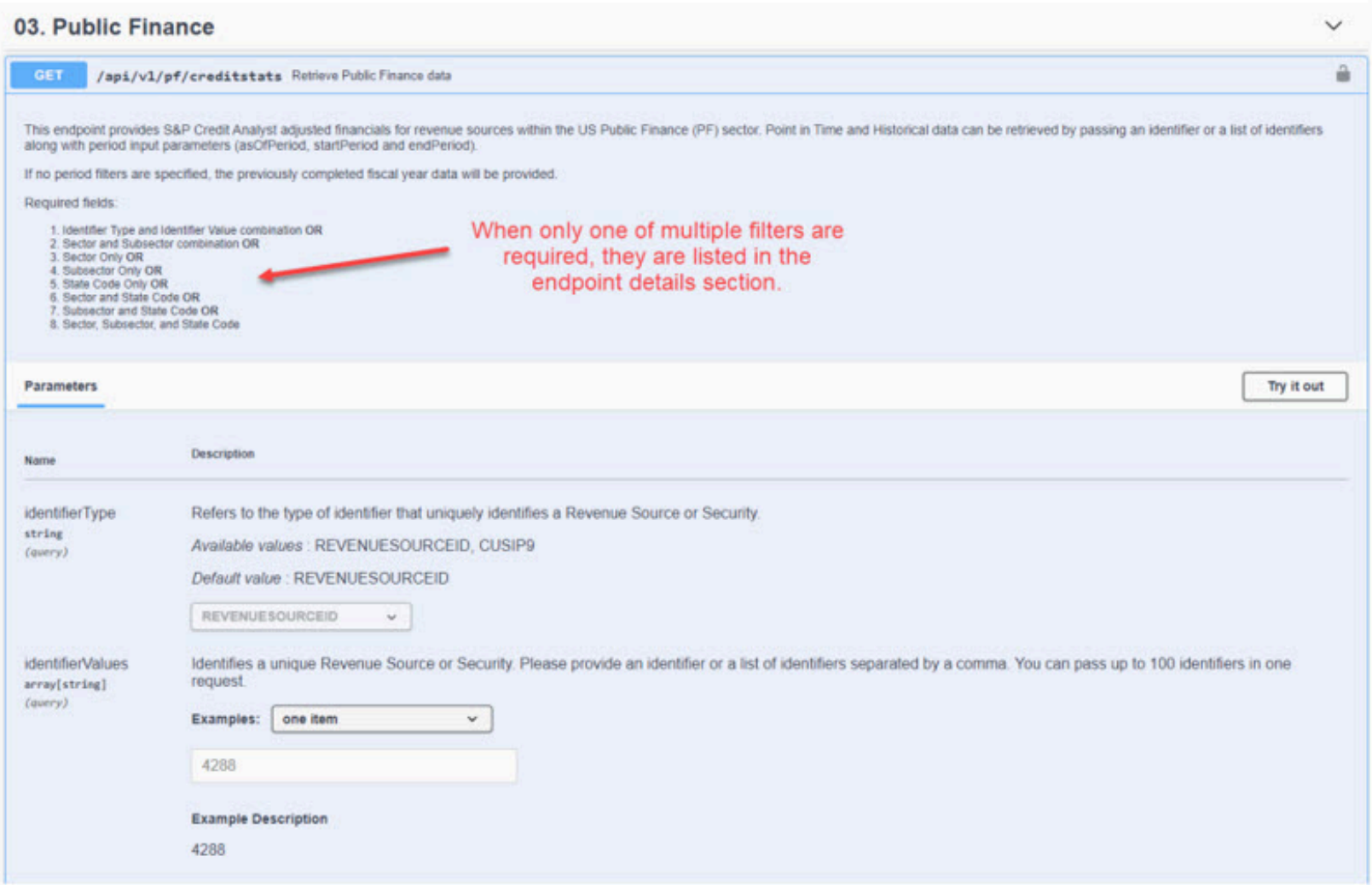
Required Filters

Required filters are indicated in the Swagger interface. In this example, the identifierType AND identifierValues filters are indicated with “* required” in red font.



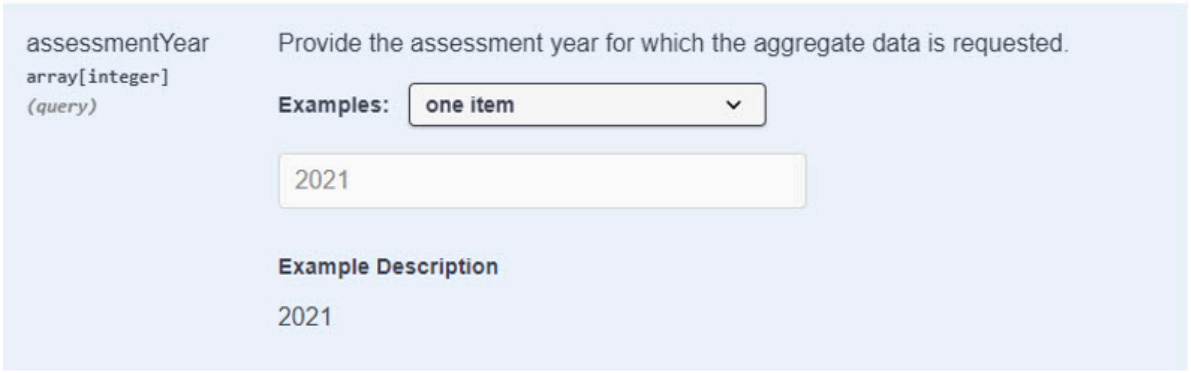
If only one or a distinct combination of multiple filters is required, the required fields are listed below in the endpoint detail section.

Additional validations may apply. If the request requirements are not met, an error message will display. Please see the error message section for more details.



Optional Filters

If optional filters are included, additional details will further explain the data that will appear in a response.



Using Swagger - Dynamic Field Selection

Some endpoints include a '**fields**' or a '**fieldGroups**' filter that allow you to choose specific fields (or group of fields) to retrieve in the response. This is known as 'Dynamic Field Selection'.

Examples of endpoints that use Dynamic Field Selection include:

- **S&P Global ESG Scores**
- **Trucost Carbon Earnings At Risk**
- **Trucost Environmental**
- **Trucost Paris Alignment**
- **Trucost Physical Risk**

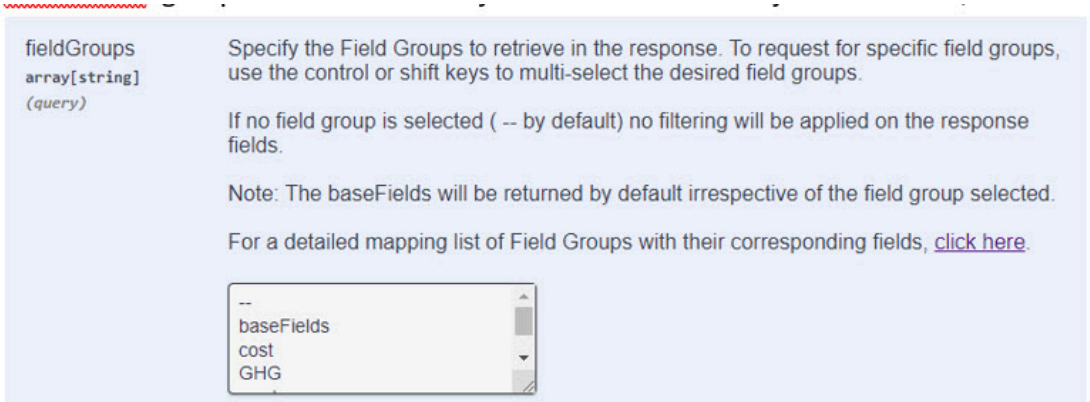
GET Requests

In the Swagger interface, both 'fields' and 'fieldGroups' filters appear in a combo box and are sorted alphabetically.

To request:

- All fields, select " -- "
- Specific fields, use the control or shift keys to multi-select the desired fields

A 'baseFields' group is defined for every dataset and will always be returned, even if not selected.



The click here link provides additional details for fields and field groups.

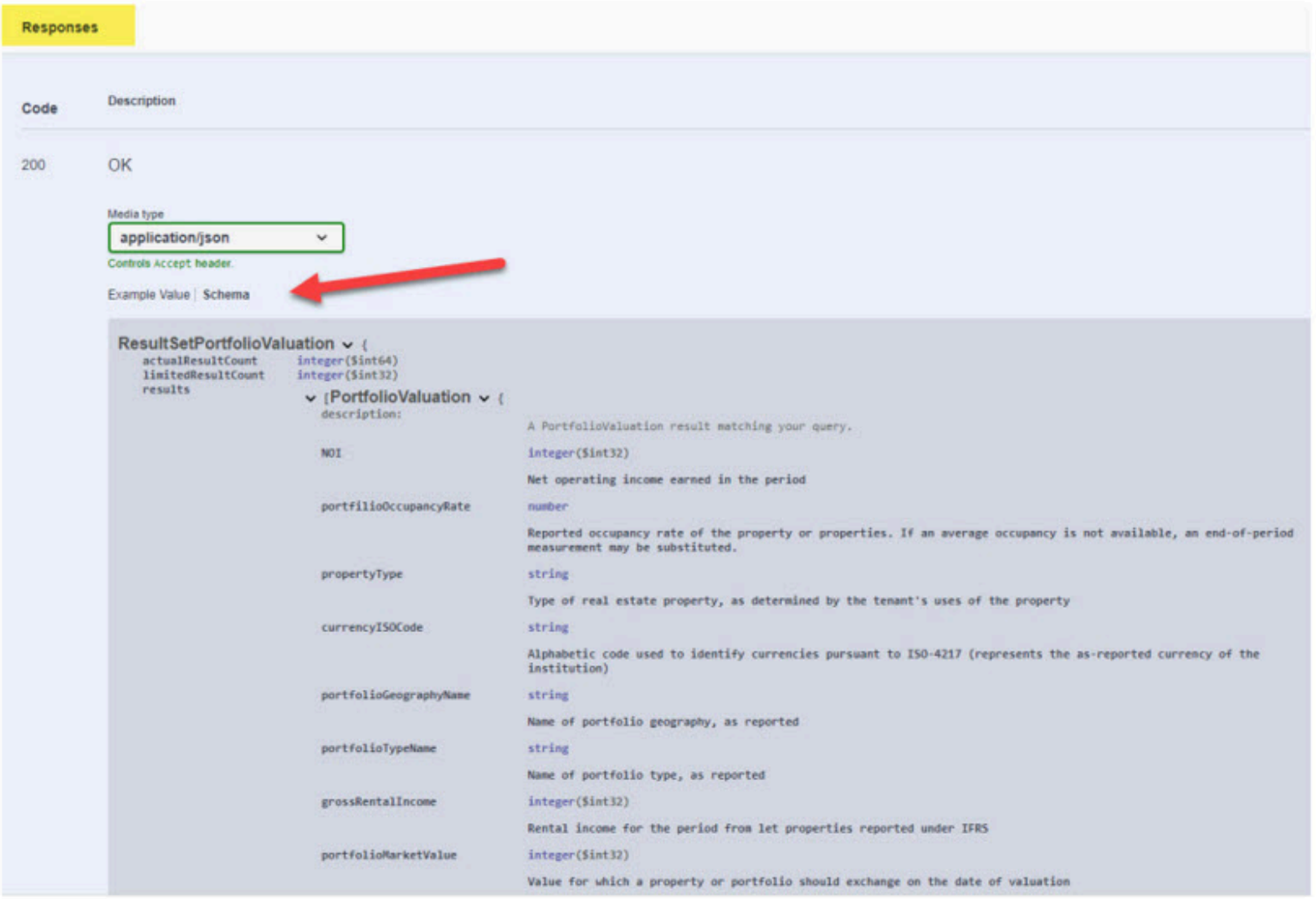
Using Swagger - Reference Endpoints

When a filter accepts a large number of filter values, Reference endpoints provide the acceptable values.

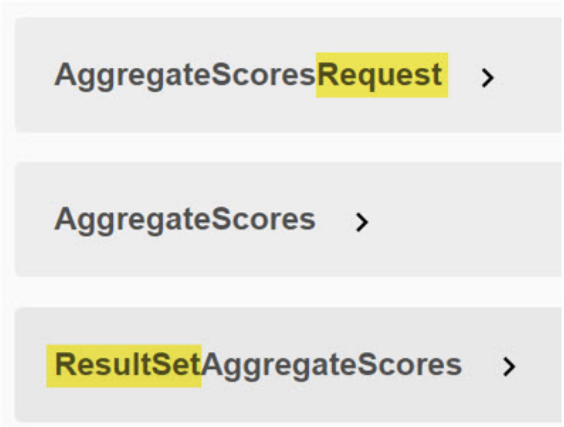
GET method

Using Swagger - Schemas

Swagger Schemas provide the technical structure for a given endpoint. Endpoint schemas are displayed in the **Responses** section. To view a schema, click **Schema**.



Each endpoint should have three schemas:



The schema layout for each is as follows:

Request Schema

The Request schema helps you build the API request by providing instructions for each filter.

AggregateScoresRequest {

aggregationLevel	string Select the level at which the aggregation is requested. Assessment year and Industry level aggregate scores are at a dimension level and the criteria level aggregate scores are at a criteria level. Only 1 aggregation level can be specified in a single request. By default all levels of aggregation will be returned. For a list of Aggregation Levels with their corresponding fields click here.
assessmentYear	[Provide the assessment year for which the aggregate data is requested. integer($int32) Provide the assessment year for which the aggregate data is requested.]
minAssessmentYear	integer($int32) Provide the minimum assessment year.
maxAssessmentYear	integer($int32) Provide the maximum assessment year.
csaScoreTypeName	string Select the CSA score type. By default all score types will be returned.
criteriaId	[Provide requested list of criteria Id(s). By default data for all criteria will be returned. This can be found in the /api/v1/reference/samcriteria reference endpoint. integer($int32) Provide requested list of criteria Id(s). By default data for all criteria will be returned. This can be found in the /api/v1/reference/samcriteria reference endpoint.]
csaIndustryId	[Provide the requested list of industry Id(s). This can be found in the /api/v1/reference/csaindustry reference endpoint. integer($int32) Provide the requested list of industry Id(s). This can be found in the /api/v1/reference/csaindustry reference endpoint.]
score	[Score > {...}]
isLatest	string *example: Yes* Select No to get all historic data. Default value is Yes which returns most recent information available for a given assessment year.
page	integer($int32) The requested page number.
pageSize	integer($int32) The requested number of results per page. Max 200

}

Endpoint Schema

The Endpoint schema provides a list of field definitions.

AggregateScores {

assessmentYear	integer($int32) Year of Corporate Sustainability Assessment campaign.
aggregateScoreDate	string($date-time) As of date for a particular campaign-level aggregation.
csaScoreTypeName	string Name of the CSA score type.
criteriaId	integer($int32) Unique Identifier for a criteria.
criteriaName	string The sustainability criteria are selected based on a materiality analysis. This analysis results in a materiality matrix for each ESG Score industry, which serves as the basis for determining the applicability and weights of the sustainability criteria in the CSA questionnaire. The criteria focus on financially material, industry-specific sustainability factors that have a link to long-term financial performance and drive business value. The criteria are composed of underlying questions. The CSA questionnaire is composed of both cross-industry and industry-specific criteria. Each industry-specific questionnaire is composed of -on average- 20 to 25 ESG criteria.
csaIndustryId	integer($int32) Unique Identifier for a CSA industry.
csaIndustryName	string Name of the CSA industry.
coverageCount	integer($int32) Number of companies covered within specific year, either for a certain industry or the entire coverage universe.
criteriaScoreMin	number Minimum ESG criteria score at the ESG score industry level in a methodology year.
criteriaScoreMax	number Maximum ESG criteria score at the ESG score industry level in a methodology year.
criteriaScoreAvg	number Average ESG criteria score at the ESG score industry level in a methodology year.
esgScoreMin	number Yearly minimum ESG score at the CSA industry level.
esgScoreMax	number Yearly maximum ESG score at the CSA industry level.
esgScoreAvg	number Yearly average ESG score at the CSA industry level.

ResultSet Schema

The ResultSet schema shows the response structure and provides field definitions.

ResultSetAggregateScores {

currentPage	integer($int32) The provided page number
currentPageSize	integer($int32) The number of results on the provided page
totalPages	integer($int32) The total number of pages available
totalResults	integer($int64) The total number of results available in all pages
results	[AggregateScores { assessmentYear: integer($int32) Year of Corporate Sustainability Assessment campaign. aggregateScoreDate: string($date-time) As of date for a particular campaign-level aggregation. csaScoreTypeName: string Name of the CSA score type. criteriaId: integer($int32) Unique Identifier for a criteria. criteriaName: string The sustainability criteria are selected based on a materiality analysis. This analysis results in a materiality matrix for each ESG Score industry, which serves as the basis for determining the applicability and weights of the sustainability criteria in the CSA questionnaire. The criteria focus on financially material, industry-specific sustainability factors that have a link to long-term financial performance and drive business value. The criteria are composed of underlying questions. The CSA questionnaire is composed of both cross-industry and industry-specific criteria. Each industry-specific questionnaire is composed of -on average- 20 to 25 ESG criteria. csaIndustryId: integer($int32) Unique Identifier for a CSA industry. csaIndustryName: string Name of the CSA industry. coverageCount: integer($int32) Number of companies covered within specific year, either for a certain industry or the entire coverage universe. criteriaScoreMin: number Minimum ESG criteria score at the ESG score industry level in a methodology year. criteriaScoreMax: number Maximum ESG criteria score at the ESG score industry level in a methodology year. criteriaScoreAvg: number Average ESG criteria score at the ESG score industry level in a methodology year.

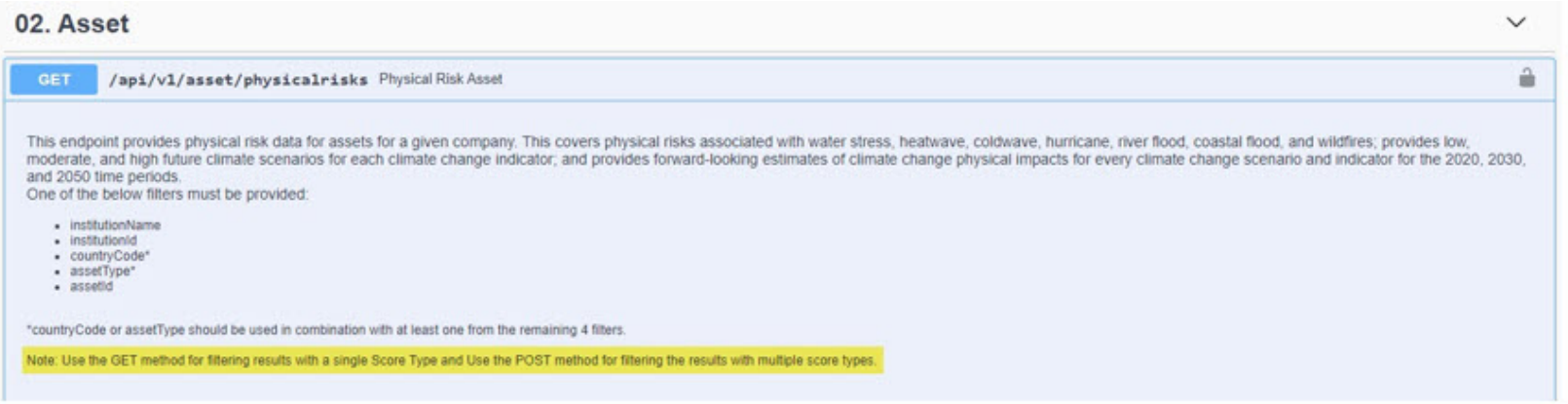
Using Swagger - GET vs. POST

When deciding whether to use GET or POST, please consider the following:

GET Method

In the GET method, key/value pairs are visible in the URL. The URL character length in the GET method is limited to a maximum of 2,048 characters, minus the number of characters in the actual path.

If you could be impacted by this limitation in the GET request, language is provided on the Swagger page. Following is an example:



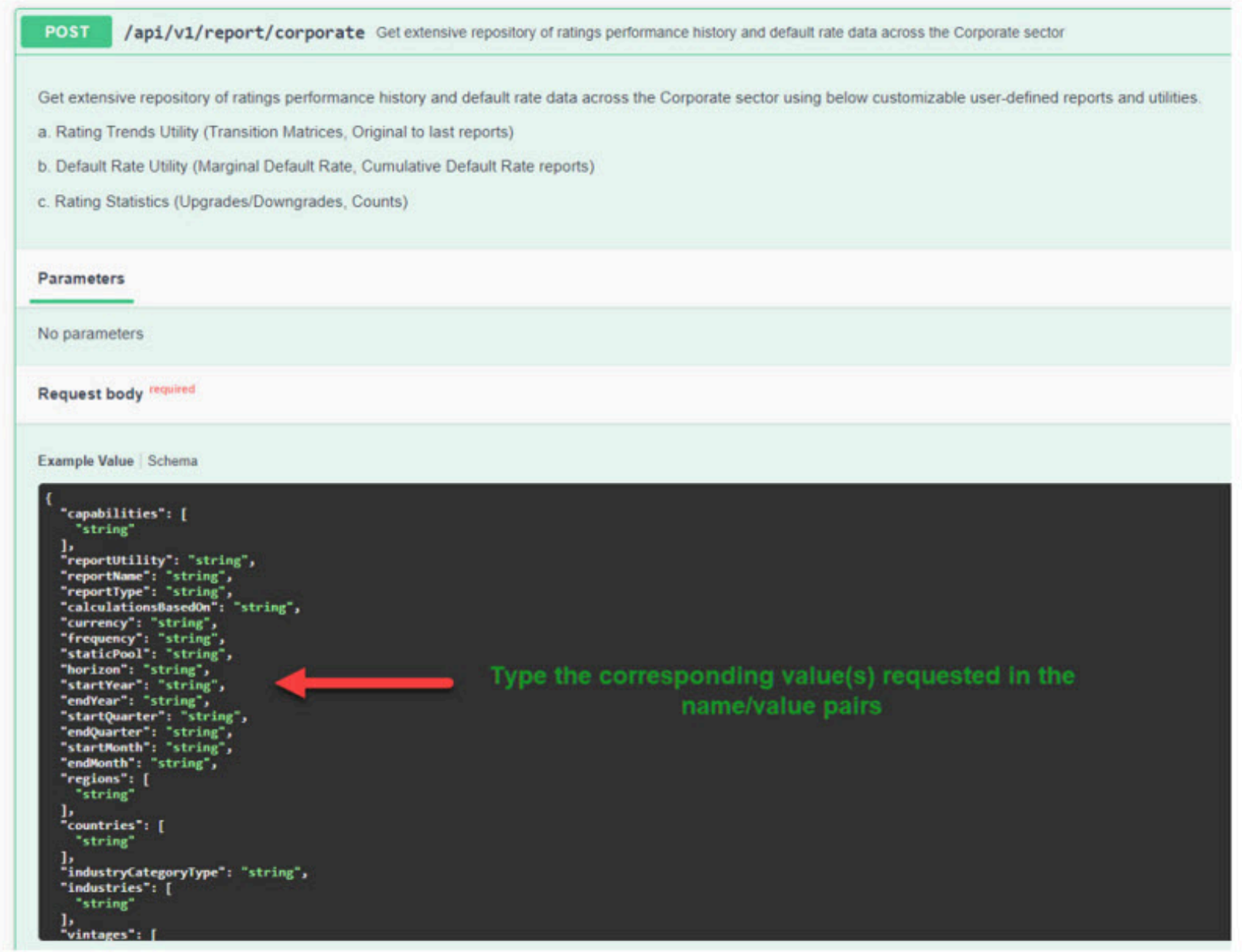
Post Method

In the POST method, key/value pairs are not visible in the URL. Therefore, the POST method is not limited by the size of the URL for submitting key/value pairs. These pairs are transferred in the request body and not in the URL.

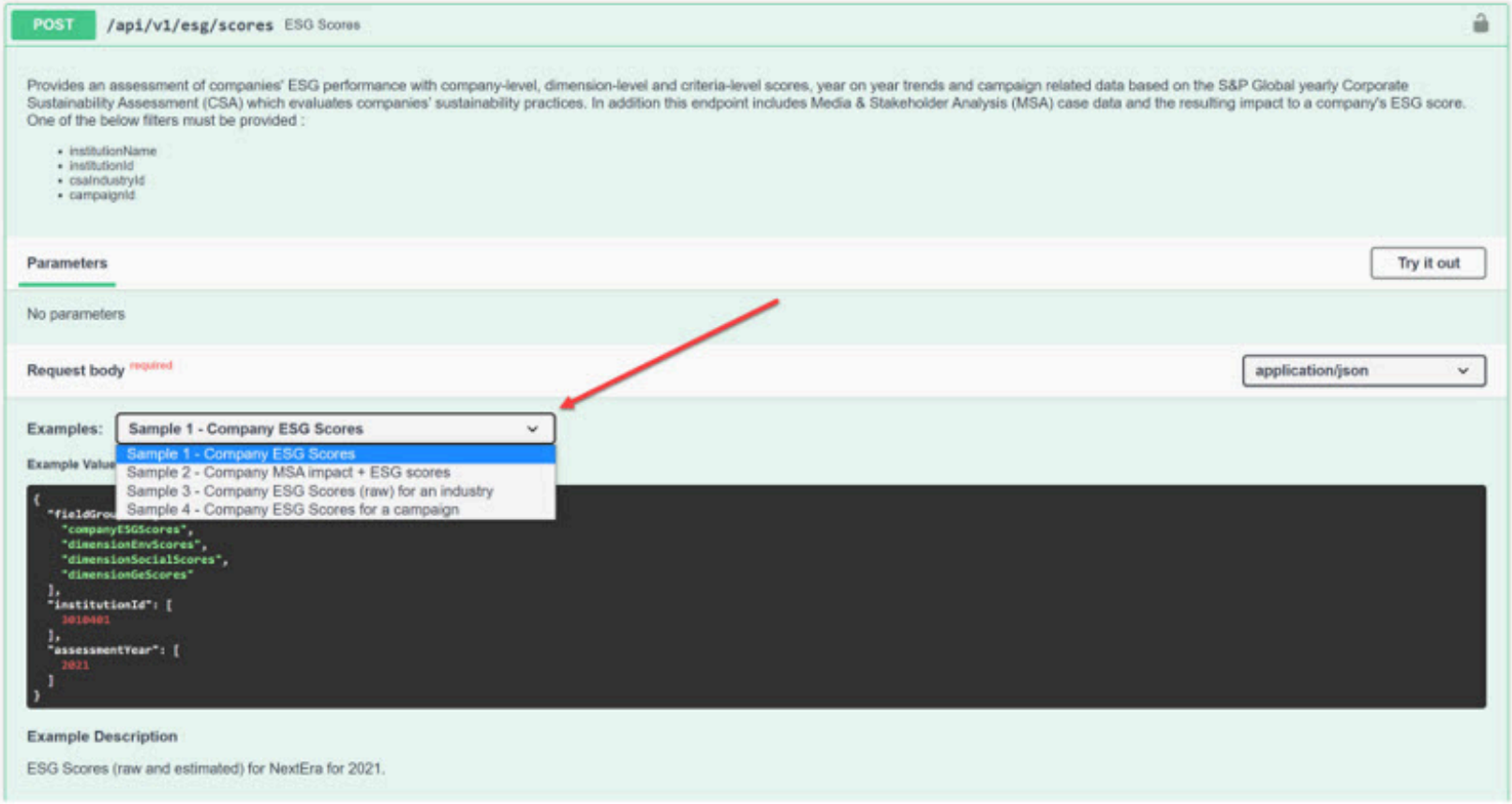
Using Swagger - POST Requests

With certain datasets, POST requests are available. When both GET and POST methods are available, the URL is the same:

In Swagger, simply type the value requested in the corresponding key/value pair.



Sample POST requests are listed in a dropdown menu. Following is an example in **S&P Global ESG Scores** dataset:



Using Swagger - Responses

After each request, the Swagger UI generates both the Request URL and the cURL command.

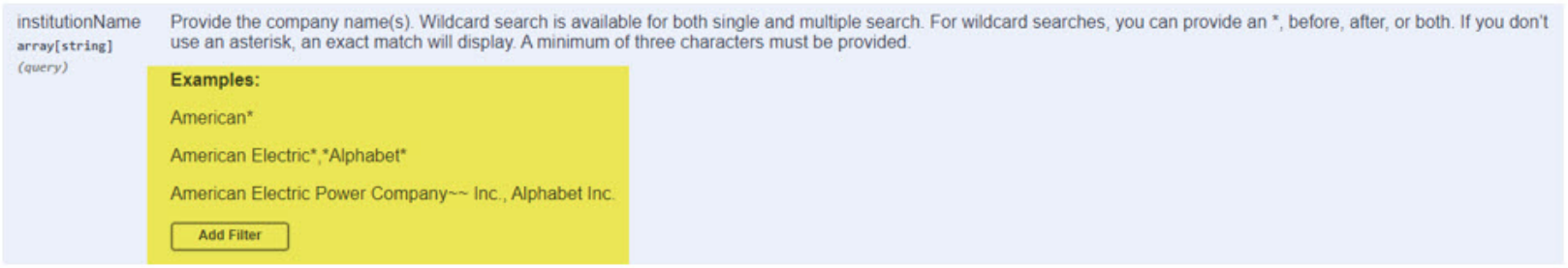
This functionality allows you to test XpressAPI requests in Swagger and copy this information into other API Development tools.

Using Swagger - 'Try it out' Functionality

In Swagger, the Try it out functionality allows you to immediately begin experimenting with endpoints. In XpressAPI, 'Try it out' is active for every dataset.

When you first click Try it out

- If you haven't previously authenticated your access,
  - 'Try it out' allows you to retrieve a bearer token. Please see the Authentication section for more details.
- If you have already authenticated your access,
  - Filter input functionality is activated
  - Default filters are prepopulated with examples of filter values
  - Each filter has at least one filter value that is either pre-populated or suggested. When a filter value is not automatically applied (i.e., prepopulated), it will appear in the 'Example Description' as shown below:



In POST requests, prepopulated filter values appear in the request body.

Request body required

Example Value | Schema


```
[
  {
    "identifierValue": 112350,
    "identifierType": "companyId",
    "relationshipType": [
      "currentInvestment",
      "currentSubsidiary",
      "mergedEntity",
      "currentInvestmentArm",
      "currentFundSponsor",
      "pendingAcqOrInvest",
      "currentAffilGovtInst"
    ],
    "minPercentageOwned": 1,
    "maxPercentageOwned": 100,
    "minLevel": 1,
    "maxLevel": 3,
    "ownershipType": [
      "Majority",
      "Minority"
    ]
  }
]
```


After clicking Try it out

That button is replaced with the following three buttons:

- Clear All - Removes all default filters and filter values
- Reset - Prepopulates the filters with default filter values
- Cancel - Cancels the changes and reverts back to the 'Try it out' button

Building Requests Outside of Swagger - GET Requests Constructing URLs

REST APIs have a base URL to which the endpoint paths are appended. The XpressAPI request URL has three significant parts:

1. Base URL
2. Resource path
3. Query string

Sample Raw URL

https://xpressapi.marketplace.spglobal.com/fiscalnote/api/v1/bills/industries?billNumber=HR2500&congressSession=116

Sample Raw URL broken down into its component parts

[1]https://xpressapi.marketplace.spglobal.com/
base URL

[2]fiscalnote/[3]api/[4]v1/[5]bills/industries
endpoint

[6]?billNumber=HR2500&congressSession=116
query string (i.e., key/value pair)

The following table explains each URL element in the example provided.

Element	Name	Part	Comment
1	xpressapi.marketplace.spglobal.com	Base URL	Always terminates in a forward slash
2	fiscalnote	Endpoint	
3	api		
4	v1	Endpoint version	
5	bills/industries	The category/path of the endpoint	
6	?billNumber=HR2500&congressSession=116	Query string	? specifies the filter parameters

Building Requests Outside of Swagger - Formatting Query Parameters

Query parameters are formatted in standard https REST syntax. They appear at the end of the request URL after a question mark (`?`), with different key/value pairs separated by ampersands (&). Query parameters can be required and optional.

Case sensitivity in Key/Value Pairs

Filters (i.e., keys) are case ***in****sensitive.*

Filter Values (i.e., values) are case *sensitive*.

When a filter value includes a comma

When a filter value includes a comma "," (such as " Phoenix-Mesa-Chandler, AZ (Metro) ") the comma "," would need to be replaced by a double tilde "~~" (Phoenix-Mesa-Chandler~~ AZ (Metro) ").

A comma "," can be used as a delimiter to separate distinct filter values, however, in a GET request, please utilize the 'Add filter' functionality in Swagger.

When a filter value contains a tilde (~)

The tilde (~), if present within a filter value, can continue to be passed as tilde (~) without any change.

Special characters

: / ? # [ ] @ ! $ & ' ( ) * + , ; = " < > % { } | \ ^ `

In XpressAPI, use **standard HTTP encoding**. For special characters other than a comma, always encode when forming requests. Special characters that are not encoded are handled as errors.

Separating distinct filter values

Use a comma "," to separate distinct filter values.

Building Requests Outside of Swagger - Swagger API docs (OpenAPI Specification)

In the Swagger UI, a link is provided to access OpenAPI specifications as presented in XpressAPI.

Example: **https://xpressapi.marketplace.spglobal.com/trucostenvironmental/v3/api-docs**

This file renders the Swagger UI page in JSON, provides endpoints, expected results, and errors.

Importing the API docs into Postman

You can **import the OpenAPI Specification into Postman** by uploading a file, entering the XpressAPI API docs URL, or directly copying the JSON.

When importing your OpenAPI specification, Postman follows the endpoint hierarchy defined in the specification to create a collection organized into folders (if that dataset contains multiple levels of hierarchy). Postman uses the schemas defined in the OpenAPI to generate request and response bodies.

Building Requests Outside of Swagger - Dynamic Field Selection

Some endpoints include a '**fields**' or a '**fieldGroups**' filter. These filters allow you to choose specific fields (or group of fields) to retrieve in the response. This is known as 'Dynamic Field Selection'.

Examples of endpoints that use Dynamic Field Selection include:

- **S&P Global ESG Scores**
- **Trucost Carbon Earnings At Risk**
- **Trucost Environmental**
- **Trucost Paris Alignment**
- **Trucost Physical Risk**

To request	Append the endpoint with	Example
Only Base fields	?fieldGroups=baseFields	?fieldGroups=baseFields
Specific fieldGroups	?fieldGroups and follow with standard Rest syntax to append the query parameters	?fieldGroups=Group1 ?fieldGroups=Group1,basefields
All fieldGroups and All fields	No additional parameter is required	
Specific fields	?fields and follow with standard Rest syntax to append the query parameters	?fields=field1 ? fieldGroups=baseFields&fields=field1,field2
Specific fieldGroups and Specific fields	?fieldGroups & ?fields and follow with standard Rest syntax to append the query parameters	?fieldGroups=Group1 &fields=field1 ? fieldGroups=Group1,baseFields&fields=field1,field2

'baseFields' will always be returned, even if not selected.

POST Requests

If a GET endpoint has fields and fieldGroups implemented, the equivalent POST endpoint will also have fields and fieldGroups implemented.

- Both 'fields' and 'fieldGroups' filters are sorted alphabetically.
- 'baseFields' will always be returned, even if not selected.
- The click here link opens the dataset user guide which provides additional details for fields and field groups.

Following are some examples of how fields and field groups appear in a request and in a response for a POST endpoint. These samples represent POST requests in the **S&P Global ESG Scores** dataset.

To request	In the body of the request	Example

Only Base Fields	Include only the required parameters	(institutionId is the required parameter)
		{
		"institutionId": [4006321]
		}

Specific fieldGroups	Include the desired fieldGroup (e.g., "GHG")	{
		"fieldGroups":["GHG"]
		}

All fieldGroups and All fields	Include only the required filters. “fieldGroups” and “fields” do not need to be specified.	Example A (countryCode is the required parameter)
		{
		"countryCode": ["GB"]
		}
		
		Example B (institutionId is the required parameter)
		
		{
		"institutionId": [4006321]
		}

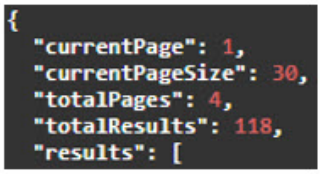
Specific fields	Include the desired individual fields within ‘Fields’ (e.g.,"street1", "street2", etc.). Please note the ["baseFields"] parameter does not need to be included.	Example (institutionName is the required parameter. Morgan is indicated as a wildcard search.)
		{
		"fields": [
		"street1",
		"street2",
		"zipCode",
		"state",
		"countryCode",
		"countryShortName"
		],
		"institutionName": [
		"*Morgan*"
		]
		}

Specific fieldGroups and Specific fields	Include the desired fieldGroup ["GHG"] and the desired individual fields within ‘Fields’ (e.g.," companyType","currencyISOcode”).	{
		"fieldGroups":["GHG"],
		"fields":
		["companyType","currencyISOCode"],
		"institutionName": ["*Morgan*"]
		}

Responses - Paginated Results

XpressAPI pagination limits the maximum results that you can see with each request. Pagination appears in the Body of the response as follows:

Body



currentPage

currentPage reflects the page number that you are currently viewing. Page number starts with 1 and ends with the number of pages found on "totalPages".

currentPageSize

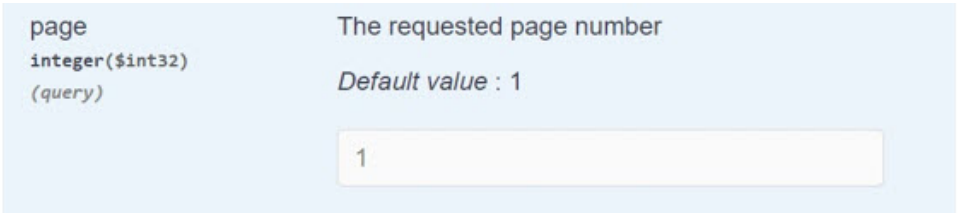
This reflects the number of results you expect to see in a given page. You can set the requested number of results per page in some endpoints.

Note: The maximum XpressAPI pagination (**pageSize**) is 1,000 root level results for each endpoint. Please note that an endpoint may explicitly mention a different value as shown in the above example. If you enter a **pageSize** value greater than the maximum allowed, XpressAPI defaults to the maximum. If you input a value less than 1, you will get an Invalid Range error.

totalPages

totalPages reflects the total number of pages generated in the response based on applied filters/parameters.

The **page** filter within the endpoint allows you to view all results through multiple API calls. Simply indicate the page number you would like to see in your request using the **page** filter. Please note that if you input a value less than 1, you will get an Invalid Range error.



totalResults

This reflects the total number of records in the subscription given the filters or parameters requested. The totalResults value is based on the record count at the root level. Please note that some datasets deliver data as multi-level nested results.

- Curly braces “{“ hold root level nested results.
- Square brackets "[" hold arrays; values are separated by "," (comma).
- Each field name is followed by ':' (colon) and the name/value pairs are separated by "," (comma).

Responses - Non-Paginated Results

Certain datasets may have been implemented without pagination. This typically happens when results are displayed in a nested structure, however, there are instances where non-nested results are implemented without pagination. Following is partial results set of a non-paginated, nested structure for the Company v2 endpoint in **S&P Global ESG Scores**.

<table>
<tr><td>


```
{
  "currentPage": 1,
  "currentPageSize": 1,
  "totalPages": 1,
  "totalResults": 1,
  "results": [
```


</td><td>"totalPages" will always display 1</td></tr>
<tr><td>


```
{
      "institutionId": 4245529,
      "institutionName": "FinecoBank Banca Fineco S.p.A.",
      "companyType": "Public",
      "assessmentYear": 2021,
      "csaIndustryId": 7,
      "csaIndustryName": "BNK Banks",
      "csaIndustryGroupName": "Banks",
      "csaSectorName": "Financials",
      "csaIndustryClassificationName": "SAM GICS",
      "campaignId": 1035,
      "campaignName": "DJSI CA 2021",
      "campaignTypeName": "SURVEY_RESPONDENT",
      "csaSharingLevelName": "All",
      "scoreDate": "2022-03-18",
      "mostRecentSequence": 1,
      "csaScoreTypeName": "Raw",
      "esgScore": 65,
      "esgScoreYoY": 11,
      "esgIndustryRank": 60,
      "envScore": 56,
      "envScoreWeight": 13,
      "envScoreYoY": 13,
      "envIndustryRank": 117,
      "socialScore": 61,
      "socialScoreWeight": 32,
      "socialScoreYoY": 17,
      "socialIndustryRank": 75,
      "geScore": 69,
      "geScoreWeight": 55,
      "geScoreYoY": 6,
      "geIndustryRank": 43,
      "dimensionScoreDate": "2022-03-18",
      "criteria": [
```


</td><td>Root level results</td></tr>
<tr><td>


```
{
        "criteriaDimension": "ENVIRONMENTAL",
        "criteriaId": 1012,
        "criteriaName": "Climate Strategy",
        "criteriaIndustryRank": 106,
        "criteriaScoreDate": "2022-03-18",
        "criteriaScore": 52,
        "criteriaWeight": 7,
        "criteriaScoreYoY": 26
      },
```


</td><td>Second level results</td></tr>
</table>

Responses – Other Formats

Some datasets have been implemented with output responses other than JSON. The **RatingsXpress: Research Article Content endpoint** is an example. The response formats in this dataset are HTML, XML, and PDF delivered with Base64 encoding. A PDF download option is also available.

Best Practices

Wildcard searches

- Always use an asterisk (*) for wildcard searches. If you don't use an asterisk (*), an exact match will display.
- For wildcard searches, you can provide an asterisk (*), before, after or both.
- You can pass a single exact match item, a list of exact match items, or a single wild card search item. Please note that a list cannot have a wildcard search item.

When a filter value contains an asterisk (*)

- When a filter value contains an asterisk (*), use an underscore to represent the asterisk. For example, a wildcard search on E*Trade would be formatted as "E_Trade*".

If the same filter is provided multiple times

A list can be passed as *either* "keys=1,2,3" *or* "keys=1&keys=2&keys=3". Please do not use a combination of these formats (e.g., *keys=1,2&keys=3*) as you won't receive accurate results.

Range fields

Range fields have 3 separate inputs:

= (equals)

Min (inclusive) - prefix

Max (inclusive) - prefix

Mapping across endpoints

Each dataset in XpressAPI provides common identifiers that may exist in more than one endpoint. To map these common identifiers across endpoints, use XpressAPI **Cross Reference Services**.

API Access, Usage, and Limits

Throttling Limits

Throttling depends on the number of calls made to each endpoint. There are three kinds of throttling limits on XpressAPI:

1. Hourly limit: Number of requests that a user can send within an hour, regardless of the endpoint being called.
2. Daily limit: Number of requests that a user can send within a day, regardless of the endpoint being called.
3. Monthly limit: Number of requests that a user can send within a month, regardless of the endpoint being called.

Rate Limiter

The rate limiter is put in place to ensure performance. It protects XpressAPI from sudden bursts of requests from a client, which in turn can affect performance for other clients. The current configuration has a burst capacity of 10 and a replenish rate of 5. This means that each user shouldn't be sending more than 5 requests within a second to any of our dataset APIs. But sudden bursts of 10 requests within a second are allowed, that is, if you suddenly send 10 requests within a second, we will allow it, but in the subsequent second, you won't be able to send anything more than 5 requests again. If you do, you will get a 429 (Too Many Requests) error. You will also get this same error if you ever send more than 10 requests within a second.

Rate limiter has another feature in that it includes this information in the below response headers with each call.

- x-ratelimit-burst-capacity: Value for burst capacity, which is currently set to 10.
- x-ratelimit-replenish-rate: Value for replenish rate, which is currently set to 5.
- x-ratelimit-requested-tokens: Value for tokens requested with each call, which is currently set to 1.
- x-ratelimit-remaining: Value for number of remaining requests allowed within that second for that user. So after the first call within a second, this will return with a value set to 9 (burst capacity - 1). And if you breach the replenish rate of 5 requests within a second, then in the subsequent second, the value will be 4 (replenish rate - 1).

Caching

The same request is cached and served back from the cache but you will be charged for it. Caching helps improve performance in the subsequent request.

Content Types

All requests returned by XpressAPI are encoded in JavaScript Object Notation (JSON), regardless of content-types specified in the Accept header.

API Usage Fees

XpressAPI access and call limits are determined by your subscription's contract. If you have questions about your contract or API charges, please contact your Account Representative.

Generating Client Code

Within Postman, you can generate code snippets in various languages and frameworks. Please visit this **Postman site** for more information.

Error Messages

The following table lists error messages you may encounter while using XpressAPI:

Error Code	Error Message	Scenario	Example
200	Data Unavailable	If a value does not exist in the database	This could be the result of: • an invalid identifier for a company or security • a company or security identifier that does not exist in the database (e.g., an unrated issuer) Please note that if multiple input filters are provided and any are incorrect, the incorrect (i.e., NULL) fields are removed and only the valid values (i.e., those that exist) will display.
400	Invalid data type	Data type is incorrect	As an example, ISINs consist of 12 alphanumeric digits. If fewer or greater than 12 digits are provided, or if all digits are numeric, a message will appear.
400	Invalid Value	Invalid input value for “period” or dateTime	• Dates must be formatted as YYYY-MM-DD. If a request input for a period = 2019 Q5, an error message will appear. • If a user attempts to input a future dateTime, an error message "value for Field X cannot be a future date."
400	Mandatory parameter Field X missing	If a filter is misspelled, and Field X is a mandatory parameter	For example, “Ag” is requested instead of “Age” and this is a mandatory parameter, a message will display “mandatory parameter ‘Age’ missing.”
400	Field X is not a valid filter	If a filter is misspelled, and Field X is an optional parameter	For example, “Ag” is requested instead of “Age” and this is an optional parameter, a message will display “Ag is not a valid filter.”
400	Invalid Range	Invalid range – Field X must be between A and B	The value specified is out of range or a range is being enforced. For example, a user inputs 1.1 for the parameter minLowerPreFloorProb and the value must be a whole number. The message "minLowerPreFloorProb (1.1): must be between 0 and 1" would appear.
400	Invalid search criteria	If the minimum string length for a search is not met	If less than n characters are input in a wildcard search, a message will appear "Must be at least n characters."
401	User Unauthorized	The user doesn’t have correct entitlements	The following messages may appear: • "No capability is found" • "Additional capability is required"
401	User unauthorized	The refresh token is invalid or expired	This message appears when an incorrect token or refresh token is used.
423	Account Locked	Account is locked after 5 failed attempts	Your account has been temporarily locked due to multiple failed login attempts. Please try again in 30 minutes. If the issue persists, contact our support team at **support.api.mi@spglobal.com** for assistance.

Error Code	Error Message	Scenario	Example
429	Too many requests	Too many requests per second	The message "Too many requests" may appear
429	Throttling Limit reached	Throttling Limits have been exceeded	The following messages may appear: • Daily Throttling limit reached • Monthly Throttling limit reached
500	Internal server error	Application specific errors e.g., Database connectivity errors	Unknown error in processing request. Please try again. If the problem persists, please reach out to the API support team.
504	Request timed out	Time out	Default is set to 60 seconds

Revision History

Revision

Version	Date	Changes
2.8	10/28/2024	• Addition of Unlock Safety
2.7	01/19/2024	• Removed all content related to Basic Authentication as the service has been deprecated
2.6	08/28/2023	• Updated terminology throughout the guide to be consistent and to further clarify. The use of the generic term "field" was replaced with filter, filter value, query parameter, or key/value pair. • Added content addressing case sensitivity
2.5	07/06/2023	• Updated to include token-based authentication • Added new Try it out 'clear all' functionality
2.4	04/10/2023	• Updated the section titled "Responses - Non-Paginated Results" to reflect v2 endpoint in S&P Global ESG Scores
2.3	02/14/2023	• Updated two screenshots • Revised content and screenshots in the section titled "Responses - Paginated Results"
2.2	01/23/2023	• Updated Error Messages section
2.1	01/12/2023	• Cosmetic updates
2.0	11/15/2022	Major update. Revisions include: • Adding an Overview section • Updating the Authorization and Login Credentials section • Including more details on Dataset subscriptions • Providing more content for Dynamic Field Selection in GET and POST requests • Added a section for Reference Endpoints • Providing details for Schemas • Adding a new section for 'Try it out' functionality • Including more details for POST requests • Providing information for non-paginated results and other result formats besides JSON • Updating Error Messages Create PDF functionality is also enabled
1.9	04/01/2022	• Updated the Swagger API docs (OpenAPI Specification) section

Version	Date	Changes
1.8	01/27/2022	• Updated the Best Practices section to state that if the same filter is provided multiple times, do not use a combination of formats (e.g., keys=1,2&keys=3).
1.7	09/15/2021	• Added Python sample code in textual form • Added an example of how to format a field containing an asterisk (*) in a wildcard search
1.6	04/29/2021	• Updated error messages • Updated screenshot for Dynamic field selection
1.5	04/26/2021	Added content that addresses the following: • Swagger authorization message reflecting entitlements • Swagger filter option for endpoints • Dynamic field selection (in Swagger and formatting request URLs) • Schema hierarchy (i.e., multi-level nested results) • Making Post requests • When a filter supports an array Updated: • totalResults section to explain root level nested results and multi-level nested results
1.4	04/12/2021	• API support email and support website update.
1.3	01/05/2021	• Added sample error message for No Data Available • Removed content explaining Actual Result Count and Limited Result Count since these no longer appear
1.2	12/16/2020	• Included more detail on throttling and results • Added content for pagination • Restructured guide
1.1	11/19/2020	• Added clarifying language
1.0	11/04/2020	• Initial version