

SUSTAINALYTICS API SERVICE

Technical Documentation

Version 1.9

Last updated: December 2020

Table of Contents

1. Introduction	2
2. Architectre	3
3. Data Model	5
4. Service Security	7
4.1 Transport Security	7
4.2 Authorization	7
5. API Calls Using Swagger UI	8
5.1 Perform AUTHORIZATION	8
5.2 Get Product Structure and Definitions	11
5.2.1 Field Definitions	11
5.2.2 Field Mappings	13
5.2.3 Field Mapping Definitions	14
5.3 Get Universe of Access	15
5.4 Get Sustainalytics Entity PDF Report	16
5.5 Get Sustainalytics Data	19
5.5.1 Identifier matching	22
6. API Integration	26
7. FAQs	27
Appendix 1 – Optional Parameters	30
Appendix 2 – API Response Codes	33

1. INTRODUCTION

The Sustainalytics API (Application Programming Interface) service provides developers with 24x7 programmatic access to current Sustainalytics' data. The API has been developed based on market standards with a primary focus on secure connectivity and ease of use. It allows users to retrieve and integrate Sustainalytics data into their own internal systems and custom or third-party applications. Default limits on the API are set to 50,000 requests in a day (24 hours) for the DataService service endpoints and 10,000 for the Reports endpoints. The API service can support up to 5 concurrent requests per account.

Historical datasets cannot be retrieved via the API. Once requested, they can be provided in a flat file format based on availability and can be placed directly on the FTP location of the client.

API permissions are managed by the Data Services team and any change requests to existing permissions can be routed to the assigned Client Advisor.

This document is meant to further provide readers with technical guidance to assist readers in understanding the general design of Sustainalytics' API service and how research data can be retrieved using the API.

Additional technical and interactive documentation can be found on the dedicated website for the API:

<https://api.sustainalytics.com/swagger/ui/index/index.html>

2. ARCHITECTRE

Sustainalytics’ REST (Representational State Transfer) API solution has been built as an OpenAPI following the specifications from **the Open API Initiative**, which aims to provide a standardized API description format based on the **OpenAPI specification**.

Registered clients can make requests to the Sustainalytics API. The response to a request will be delivered in **JSON (JavaScript Object Notation)** file format. The JSON file format is language agnostic but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others.

The below figure provides a graphical representation of the request process.

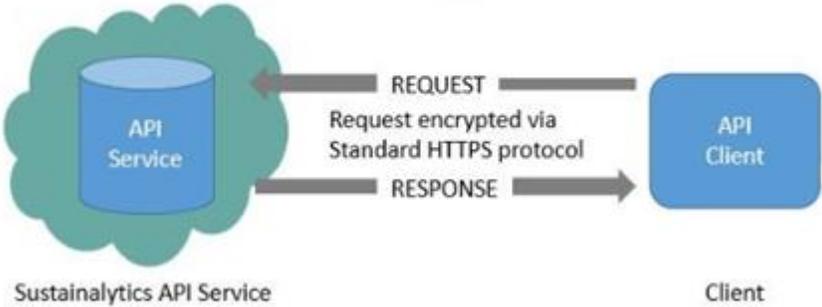
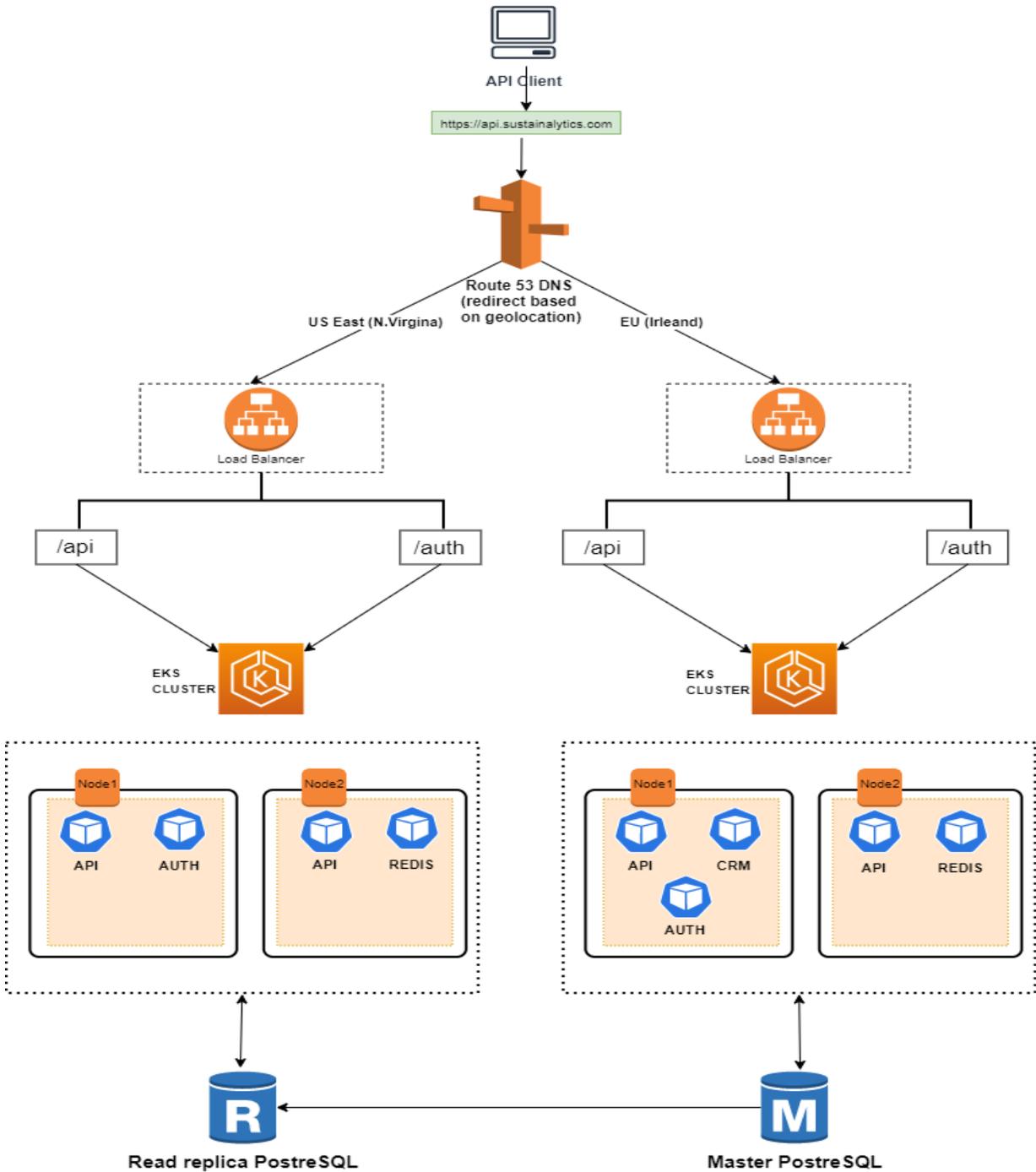


Figure 1. API Request Process Service Diagram

The API architecture illustrated in the diagram below denotes the various components involved in retrieving data via the API such as geo-locations of the load balancer, authorization components and retrieval route from postgresSQL databases.



3. DATA MODEL

The below figure 2 displays the functional data model of the API service. It explains the relationship between the various components of the API response.

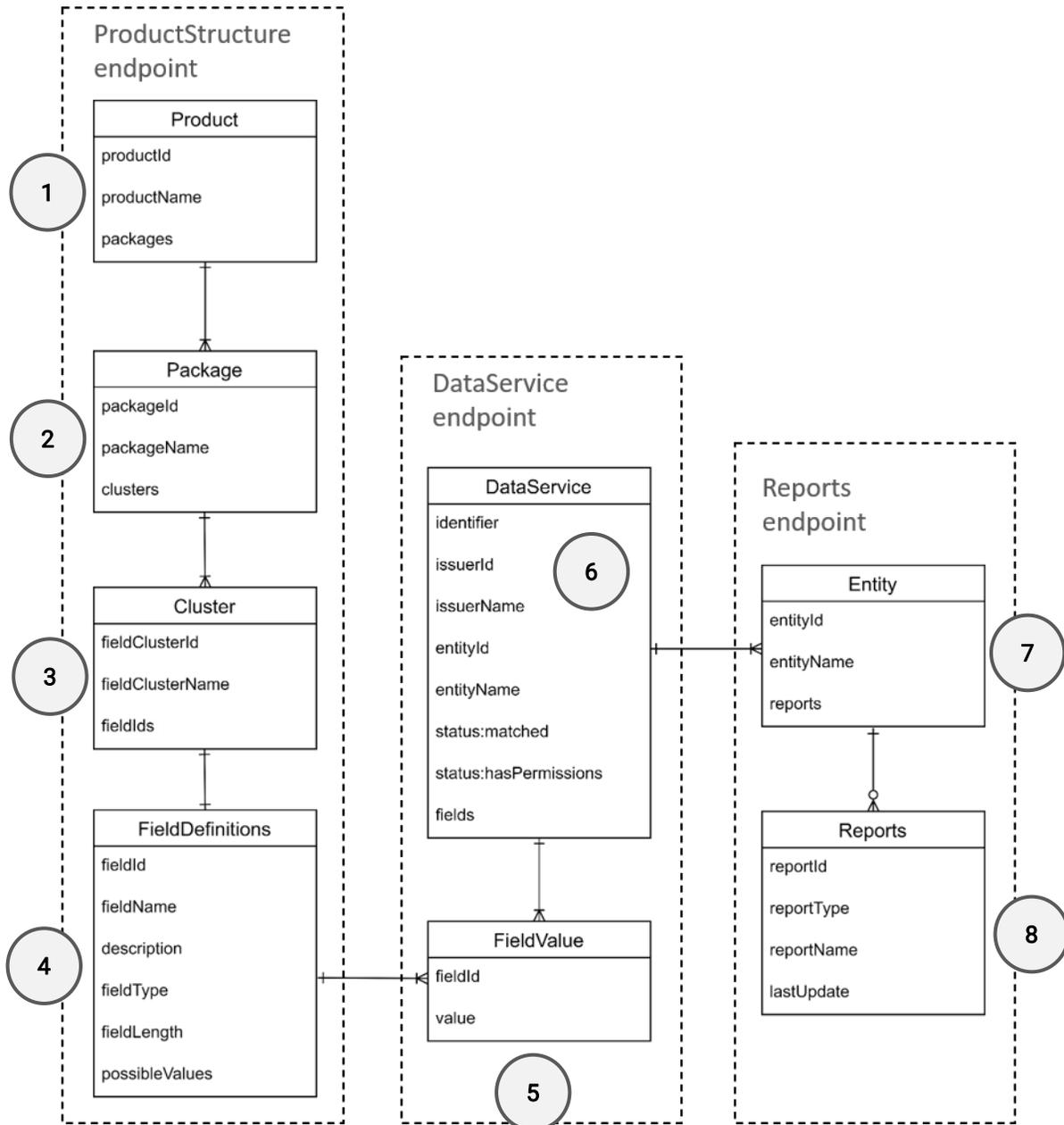


Figure 2 – API Service Data Model

The API Data model consists of the following information clusters:

- 1) Product:** describes the Research Products permissioned to the user in Sustainalytics' API. Research Products available include:
 - Corporate Governance
 - ESG Data
 - ESG Risk Rating
 - Controversies Research
 - Controversial Weapons Radar
 - Product Involvement
 - Carbon Emissions
 - Carbon Risk Rating
 - Country Risk Rating
 - Country Screening
 - Global Standards Engagement
 - Material Risk Engagement
 - Impact Metrics
 - Thematic Engagement
- 2) Package:** grouping by topic of product fields. E.g. "Corporate Data" research product includes the following packages:
 - General
 - Industry classification
 - Open source identifiers
 - Licensed identifiers
- 3) FieldCluster:** grouping of product fields by type. E.g. "Open source identifiers" package includes the following clusters:
 - Exchange ticker
 - Legal Entity Identifier
 - Bloomberg Ticker
 - Bloomberg Global Composite ID
 - Bloomberg Global ID
- 4) FieldDefinitions:** the grouping describing each research field. E.g. "Market capitalization" field is defined as follows:
 - "fieldId": 101011152799,
 - "fieldName": "Market cap. (mln. USD)",
 - "description": "Market capitalization in mln. USD",
 - "fieldType": "Numerical",
 - "fieldLength": "12",
 - "possibleValues": "NULL|min. 0.00 max. 9999999999.99"
- 5) FieldValue:** raw value of a field, identified by "fieldId".
- 6) DataService:** grouping of the relationship between the business entity ID and its corresponding research field IDs and field values based on user input and filters applied.
- 7) Entity:** the business entity ID for which the research is focused on.

- 8) Report: PDF document encapsulating product-specific information available for one entity, including visual representations of the data provided through the data services endpoints.

4. SERVICE SECURITY

4.1 TRANSPORT SECURITY

Connections to the Sustainalytics API are established over HTTPS.

The transfer of data is secured using an SSL certificate owned and managed by Sustainalytics. The SSL certificate uses 2048-bit encryption which is the most secure encryption available and supports SHA-2 algorithms.

4.2 AUTHORIZATION

Sustainalytics' API uses the **OAuth2** protocol for secure authorization. Successful authorization requires the generation of an access token which can be generated by using valid `client_id` and `client_secret` keys.

Refer to **section 5.1** to learn more about how the authorization end point can be accessed and used using Swagger.

5. API CALLS USING SWAGGER UI

In line with the OpenAPI protocol, Sustainalytics has created a website based on the Swagger framework that can be used to test the Sustainalytics API. The website can be accessed using the below URL:

```
https://api.sustainalytics.com/swagger/index.html
```

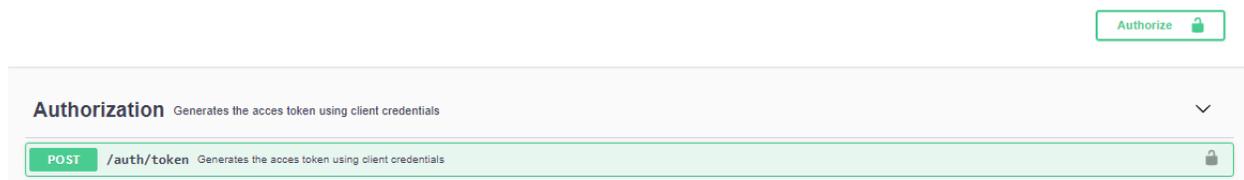
The website provides an overview of all the available endpoints for the API and allows users to interact with and familiarize themselves with sending requests and interpreting API responses. Each response content type received from the Sustainalytics API is in **JSON** format.

The sequence of steps below describes how data can be retrieved by using the Sustainalytics API.

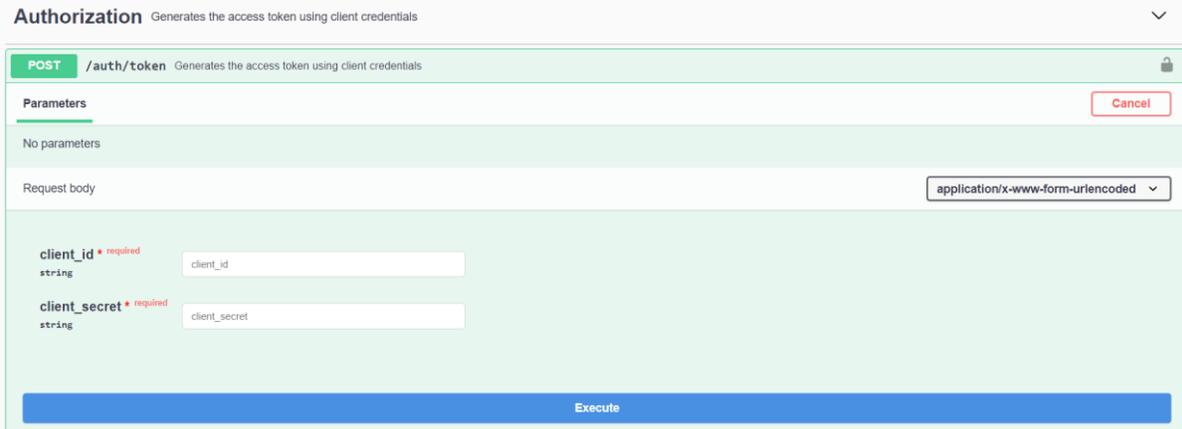
5.1 PERFORM AUTHORIZATION

To make requests to the API, an active `access_token` is required. Follow the steps listed below to retrieve the access token:

Step 1: Click on the green ribbon below to expand the Authorization section



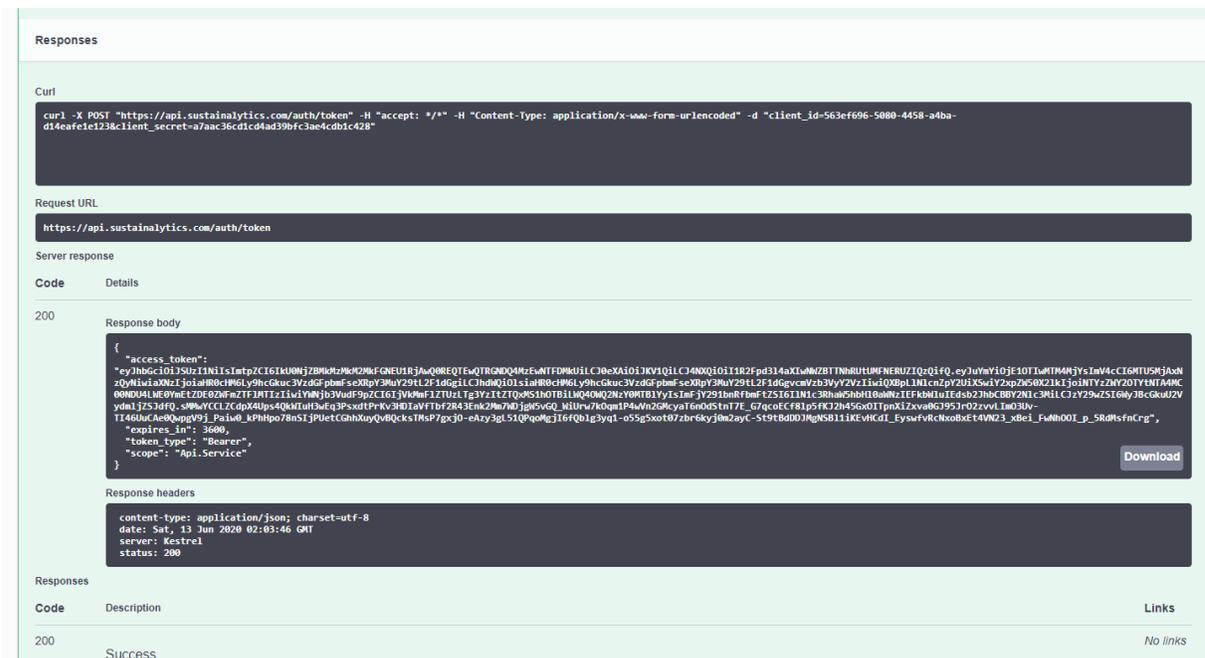
Step 2: Click in the **Try it out** button on the top right-hand side of the section. Once it is clicked, the user should be able to enter the client id and client secret in the corresponding fields as denoted in the screen shot below.



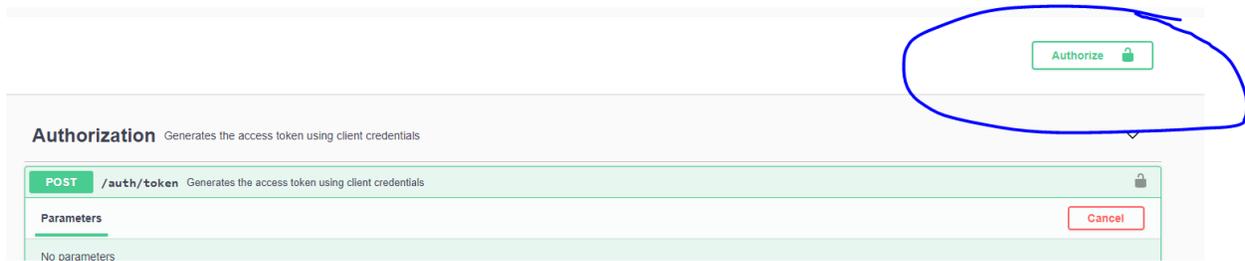
Step 3: Click on the blue colored Execute button upon which the **Responses** section will appear underneath the button

Step 4. From the Response body section, refer to the value corresponding to the access_token key, select and copy the entire value string ONLY without the double quotes (" "). This is the token and it is valid for a period of 60 minutes.

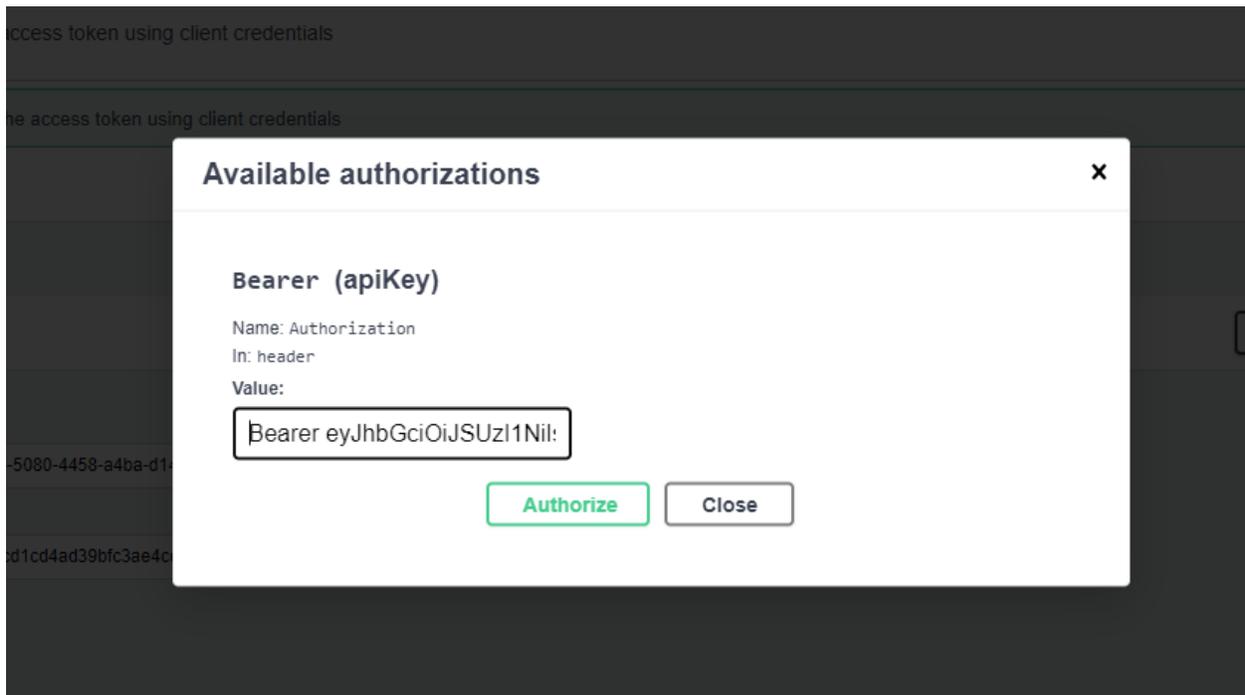
For example, from the screenshot below, copy and select the entire access_token value string starting from lower case e to lower case g



Step 6. Scroll up on the page, refer to the right-hand section of the page and select the green bordered Authorize button.



Step 7. In the Value text area, type Bearer, leave a space and then paste your access token value retrieved from step 5 above. See screenshot below to see the field text syntax.



Step 8. Click Authorize and Close the overlay windowpane and collapse the Authorization section by clicking on its accordion ribbon.

Authorization is now complete, and all the API endpoints listed on the Swagger page can now be used. As an example, refer below for steps to send a request via the DataService API endpoint.

Once the access token is received, the token will remain valid for 60 minutes. When this time has elapsed, a new access token will need to be generated by making a new request to the authorization endpoint.

5.2 GET PRODUCT STRUCTURE AND DEFINITIONS

The product structure endpoint provides Sustainalytics research organized by a specific data structure reflected in the name of the endpoint, for example FieldDefinitions, FieldMappings, etc. Each product is broken down into data packages and each data package is broken down into field clusters, which contain sets of data fields as explained in the [data model section](#). Each data field has a corresponding unique field id and the value associated with the data field is a currently valid value.

The ProductStructure endpoint provides an overview of the data fields available in the Sustainalytics API and the unique FieldIds linked to each of these data fields. Within this service there are three endpoints, as described below.

ProductStructure		Product structure endpoint
GET	/v1/FieldDefinitions	Get field definitions
GET	/v1/FieldMappings	Get product structure
GET	/v1/FieldMappingDefinitions	Get product structure with field definitions

5.2.1 FIELD DEFINITIONS

<https://api.sustainalytics.com/v1/FieldDefinitions>

The FieldDefinitions endpoint allows users to retrieve the definitions of the datafields that have been permissioned. Each Sustainalytics data field has the following defined properties: fieldName; fieldId; description; fieldType; fieldLength; possibleValues.

Response body

```
[
  {
    "fieldId": 101011112999,
    "fieldName": "EntityId",
    "description": "ID to uniquely identify an entity",
    "fieldType": "Integer",
    "fieldLength": "10",
    "possibleValues": "min. 0 max. 9999999999"
  },
  {
    "fieldId": 101011122899,
    "fieldName": "EntityName",
    "description": "Entity name",
    "fieldType": "String",
    "fieldLength": "200",
    "possibleValues": "n/a"
  },
  {
    "fieldId": 101011132899,
    "fieldName": "Country",
    "description": "Country of domicile",
    "fieldType": "String",
    "fieldLength": "50",
    "possibleValues": "n/a"
  }
],
```

Implementation Notes

Get the mapping between Sustainalytics research data and assigned field ids.

```
Inline Model [
  Inline Model 1
]
Inline Model 1 {
  fieldId (integer),
  fieldName (string),
  description (string),
  fieldType (string),
  fieldLength (string),
  possibleValues (string)
}
```

Response Content Type: JSON

5.2.2 FIELD MAPPINGS

<https://api.sustainalytics.com/v1/FieldMappings>

The FieldMappings endpoint allows users to retrieve the following Product constituents structure: ProductName (name;id); DataPackages (name;id); FieldClusters (name;id); FieldIds.

Response body

```
[
  {
    "productId": 10,
    "productName": "Corporate data",
    "packages": [
      {
        "packageId": 1010,
        "packageName": "General",
        "clusters": [
          {
            "fieldClusterId": 101010,
            "fieldClusterName": "Company info",
            "fieldIds": [
              101011112999,
              101011122899,
              101011132899,
              101011142899,
              101011152799,
              101011162799,
              101011182899,
              101011193199,
              101011272899,
              101011293099,
              101011313099,
              101011322799,
              .....
            ]
          }
        ]
      }
    ]
  }
]
```

Implementation Notes

Get the product structure constituents: products, data packages, field clusters, field ids.

Inline Model [
Inline Model 1
]

Inline Model 1 {
productId (integer),
productName (string),
packages (Array[PackageModel], optional)
}

PackageModel {
packageId (integer),
packageName (string),
clusters (Array[FieldClusterModel], optional)
}

FieldClusterModel {
fieldClusterId (integer),
fieldClusterName (string),
fieldIds (Array[integer]),
}

Response Content Type: JSON

5.2.3 FIELD MAPPING DEFINITIONS

<https://api.sustainalytics.com/v1/FieldMappingDefinitions>

The FieldMappingDefinitions endpoint allows users to call the following Product constituents structure: ProductName (name;id); DataPackages (name;id); FieldClusters (name;id); FieldIds (fieldname; fieldid; description; fieldType; fieldLength; possibleValues).

This is the most comprehensive Product Structure endpoint as it will return the entire data structure for the data fields that the account has permissions on.

Response body

```
[
  {
    "productId": 10,
    "productName": "Corporate data",
    "packages": [
      {
        "packageId": 1010,
        "packageName": "General",
        "clusters": [
          {
            "fieldClusterId": 101010,
            "fieldClusterName": "Company info",
            "fieldDefinitions": [
              {
                "fieldId": 10101112999,
                "fieldName": "EntityId",
                "description": "ID to uniquely identify an entity",
                "fieldType": "Integer",
                "fieldLength": "10",
                "possibleValues": "min. 0 max. 9999999999"
              }
            ]
          }
        ]
      }
    ]
  }
]
```

Implementation Notes

Get the product structure constituents: products, data packages, field clusters, field ids, field definitions.

Inline Model [

Inline Model 1

]

Inline Model 1 {

productId (integer),

productName (string),

packages (Array[PackageModel])

}

PackageModel {

packageId (integer),

packageName (string),

clusters (Array[FieldClusterModel], optional)

}

FieldClusterModel {

fieldClusterId (integer),

fieldClusterName (string),

fieldDefinitions (Array[FieldDefinitionModel])

}

FieldDefinitionModel {

fieldId (integer),

fieldName (string),

description (string),

fieldType (string),

fieldLength (string),

possibleValues (string)

}

Response Content Type: JSON

5.3 GET UNIVERSE OF ACCESS

The UniverseOfAccess endpoint allows users to retrieve the list of EntityIds contained in the universes of access permitted to their account..

UniverseOfAccess Universe of access endpoint

GET /v1/UniverseOfAccess Get universe of access

<https://api.sustainalytics.com/v1/UniverseOfAccess>

```
{
  "securityListId": "SL_40",
  "securityListName": "Ratings",
  "entityIds": [
    1017162097,
    1032187584,
    1017164437,
    1008607053,
    2004746354,
    1380312069,
    1043105073,
    1016061011,
    2000489802,
    1016882415,
    2000702352,
    1008107758,
    1012818068,
    1011478697,
    1048310362,
    2001123030,
    1548930312,
    1008011104,
    2000000000
  ]
}
```

Implementation Notes

Get all companies in the universe of access.

Response Class (Status 200)

OK

ModelExample Value

Inline Model [

Inline Model 1

]

Inline Model 1 {

securityListId (string),

securityListName (string),

EntityIds (Array[integer])

}

Response Content Type: JSON

5.4 GET SUSTAINALYTICS ENTITY PDF REPORT

The ReportService provides the ability to determine the PDF reports available for retrieval through the Sustainalytics API. Within this service there are four endpoints, as described below.

Reports Reports endpoint	
GET	/v1/ReportService Get available report types
GET	/v1/ReportService/{identifier} Get available report types by entity identifier
GET	/v1/ReportService/{identifier}/{reportId} Get report
GET	/v1/ReportService/url/{identifier}/{reportId} Get report url

We recommend this endpoint as it has the fastest response time.

<https://api.sustainalytics.com/v1/ReportService>

The ReportService endpoint allows users to retrieve a list of all available PDF report types by ReportId, ReportType, and ReportName for companies belonging to the universe of access.

Response body

```
{
  "entityId": 1007902331,
  "entityName": "1-800-Flowers.com Inc",
  "reports": [
    {
      "reportId": 111210113299,
      "reportType": "Core Company ESG PDF report",
      "reportName": "1-800-Flowers.com_Inc_ESGRatingsReport_02042020.pdf"
    },
    {
      "reportId": 111210123299,
      "reportType": "Core Summary ESG PDF report",
      "reportName": "1-800-Flowers.com_Inc_ESGSummaryReport_02042020.pdf"
    },
    {
      "reportId": 121110113299,
      "reportType": "Company Controversy PDF report",
      "reportName": "1-800-Flowers.com_Inc_ControversyReport_02042020.pdf",
      "lastUpdate": "2018-02-15"
    },
    {
      "reportId": 181413113299,
      "reportType": "Core Company Risk Rating PDF report",
      "reportName": "1-800-Flowers.com_Inc_CoreRiskRatingsReport_02042020.pdf",
      "lastUpdate": "2019-11-15"
    }
  ]
}
```

Implementation Notes

Get available report types for companies in the universe of access

```
Inline Model [
  Inline Model 1
]
Inline Model 1 {
  EntityId (integer),
  EntityName (string),
  reports (Array[ReportDetails])
}
ReportDetails {
  reportId (integer),
  reportType (string),
  reportName (string),
  lastUpdate (string)
}
Response Content Type: JSON
```

<https://api.sustainalytics.com/v1/ReportService/{Identifier}>

The ReportService/{Identifier} endpoint allows users to retrieve a list of available reportIds for a specific entity. Using the EntityId (or another security identifier) in the identifier parameter, the response will return the list of available PDF reports and related information.

Response body

```
{
  "issuerId": 1007896995,
  "issuerName": "Amazon.com Inc",
  "entityId": 1007896995,
  "entityName": "Amazon.com Inc",
  "reports": [
    {
      "reportId": 111110113299,
      "reportType": "Comprehensive Company ESG PDF report",
      "reportName": "Amazon.com_Inc_ESGRatingsReport_02042020.pdf",
      "lastUpdate": "2019-11-12"
    },
    {
      "reportId": 111110123299,
      "reportType": "Comprehensive Summary ESG PDF report",
      "reportName": "Amazon.com_Inc_ESGSummaryReport_02042020.pdf",
      "lastUpdate": "2019-11-12"
    },
    {
      "reportId": 111110133299,
      "reportType": "Comprehensive TearSheet PDF report",
      "reportName": "1007896995.pdf"
    },
    {
      "reportId": 121110113299,
      "reportType": "Company Controversy PDF report",
      "reportName": "Amazon.com_Inc_ControversyReport_02042020.pdf",
      "lastUpdate": "2020-03-31"
    }
  ]
}
```

Implementation Notes

Get available report types for the entity specified by identifier.

ModelExample Value
 GetEntityListResponse {
 EntityId (integer),
 EntityName (string),
 reports (Array[ReportDetails])
 }
 ReportDetails {
 reportId (integer),
 reportType (string),
 reportName (string),
 lastUpdate (string)
 }

Response Content Type: JSON

`https://api.sustainalytics.com/v1/ReportService/{identifier}/{ReportId}`

The `ReportService/{identifier}/{ReportId}` endpoint allows users to retrieve the specified report for a specific entity. Using the `EntityId` (or another security identifier) in the `identifier` parameter, and a `reportId` in the `report` parameter, the response body will include a stream to a PDF document.

Response body

[Download file](#)

Implementation Notes

Get a report for the entity by identifier.

```
Stream {
  canRead (boolean, read only),
  canSeek (boolean, read only),
  canTimeout (boolean, read only),
  canWrite (boolean, read only),
  length (integer, read only),
  position (integer, optional),
  readTimeout (integer, optional),
  writeTimeout (integer, optional)
}
```

Response Content Type: application/pdf

`https://api.sustainalytics.com/v1/ReportService/url/{Identifier}/{ReportId}`

The `ReportService/{identifier}/{ReportId}` endpoint allows users to retrieve a specified PDF report for a specific entity by accessing a URL returned in the service response. Using the `EntityId` (or another security identifier) in the `identifier` parameter, and a `reportId` in the `report` parameter, the response body will include a URL that links to the PDF document. This URL is valid for 24 hours, starting from the API call. After 24 hours, loading the URL will throw an error and a new API call is required for a new URL. This is the recommended endpoint for retrieving PDF reports as it has the fastest response time.

Response body

```
{
  "filename": "Amazon.com_Inc_RiskRatingsReport_02042020.pdf",
  "url": "https://live-globalaccess-us-east.s3.amazonaws.com/RiskRatingsReport_B/Amazon.com_Inc_RiskRatingsReport_02042020.pdf?AWSAccessKeyId=AKIAJKGZZMOFAVOVLHJA&Expires=1585969350&Signature=7zsGcJPF%2FB4fmADsNi2FH5te44Q%3D"
}
```

5.5 GET SUSTAINALYTICS DATA

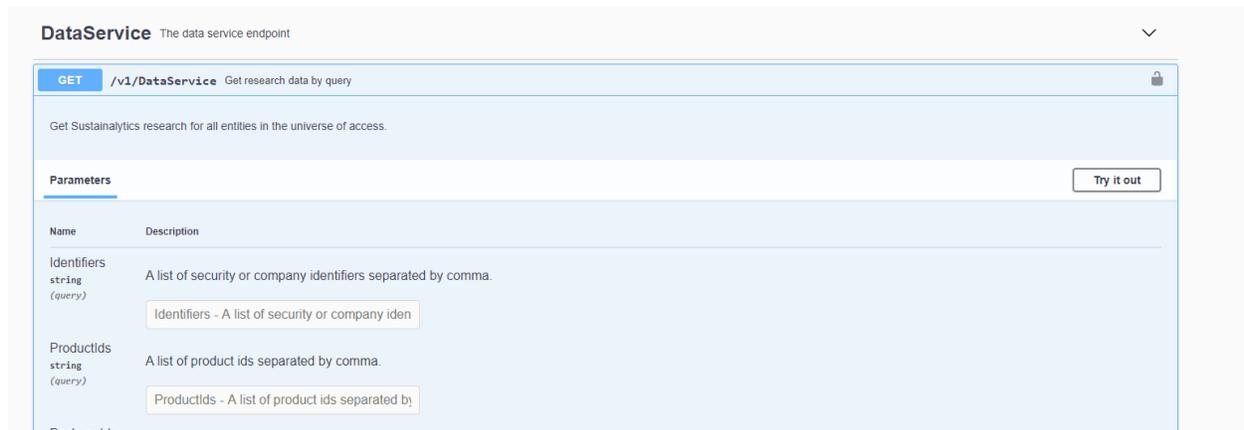
The DataService endpoints enable the user to retrieve the research data for companies included in the universe of access. Within this service there are two endpoints, as described below.



The screenshot shows a list of two API endpoints for the DataService. Each endpoint is represented by a blue ribbon with a 'GET' method and a description of the endpoint's function.

- Endpoint 1:** Method: GET, Path: `/v1/DataService`, Description: Get research data by query.
- Endpoint 2:** Method: GET, Path: `/v1/DataService/{identifier}`, Description: Get research data by identifier.

Expand the `/v1/DataService` section by clicking on its blue-colored ribbon to see the endpoint structure and associated filters.



The screenshot shows the expanded details for the `/v1/DataService` endpoint. It includes a description, a 'Try it out' button, and a table of parameters.

DataService The data service endpoint

GET `/v1/DataService` Get research data by query

Get Sustainalytics research for all entities in the universe of access.

Parameters Try it out

Name	Description
Identifiers string (query)	A list of security or company identifiers separated by comma. <input type="text" value="Identifiers - A list of security or company iden"/>
ProductIds string (query)	A list of product ids separated by comma. <input type="text" value="ProductIds - A list of product ids separated by"/>

Click on the **Try it out** button in the right-hand side of the section. Then scroll down and click on the blue colored **Execute** button, upon which the Responses section will appear.

Refer to the Response body section to see the JSON formatted response of the DataService API endpoint call. This API response filters out all records with NULL values.

Response body

```

},
{
  "entityId": 1007896995,
  "entityName": "Amazon.com Inc",
  "fields": {
    "181110101499": 0.173943750000003,
    "181110112399": 30.56944375,
    "181110121199": 57.5,
    "181110121499": 2.68503958771635,
    "181110131299": {
      "rank": 7005,
      "peers": 12273
    },
    "181110141499": 1.45,
    "181110142899": "High",
    "181110151199": 100,
    "181110161299": {
      "rank": 62,
      "peers": 62
    },
    "181110161499": "The company's risk score has not changed sign",
    "181110171499": "The company's management score has increased",
    "181110172399": 24.475659472422066,
    "181110181499": "The company's exposure score has not changed",
    "181110182899": "Weak",
    "181110192399": 39.5,
    "181110202899": "Medium",
    "181110212399": 36.4875,
  }
}

```

Implementation Notes

Get Sustainalytics research for all companies in the universe of access.

Response Class (Status 200) OK

ModelExample Value

Inline Model [

Inline Model 1

]

Inline Model 1 {

EntityId: (integer),

EntityName: (string),

fields: (inline_model, optional)

inline_model {

}

Response Content Type: JSON

<https://api.sustainalytics.com/v1/DataService/{identifier}>

The DataService/{identifier} endpoint allows users to receive Sustainalytics Research Data for a specific entity. Using the EntityId (or another security identifier) in the identifier parameter, the response body will include all data referring to that specific entity or security, within the universe of access.

The response structure is similar to the DataService endpoint when using the optional parameter for *identifier*, as it contains the issuer id/name and the entity id/name.

Issuer = the identified issuing entity of the security

Entity = the linked research entity for which data is provided

Note that the response structure is different from the DataService endpoint when using the optional parameter for *identifier*, as it does not provide the full response on a security with multiple matches since response is only provided for the specific inputted identifier value.

Response body

```
{
  "identifier": "amz",
  "issuerId": 1007896995,
  "issuerName": "Amazon.com Inc",
  "entityId": 1007896995,
  "entityName": "Amazon.com Inc",
  "fields": {
    "181110101499": 0.173943750000003,
    "181110112399": 30.56944375,
    "181110121199": 57.5,
    "181110121499": 2.68503958771635,
    "181110131299": {
      "rank": 7005,
      "peers": 12273
    },
    "181110141499": 1.45,
    "181110142899": "High",
    "181110151199": 100,
    "181110161299": {
      "rank": 62,
      "peers": 62
    }
  }
}
```

Implementation Notes

Get Sustainalytics research data by entity identifier.

Response Class (Status 200) OK

ModelExample Value

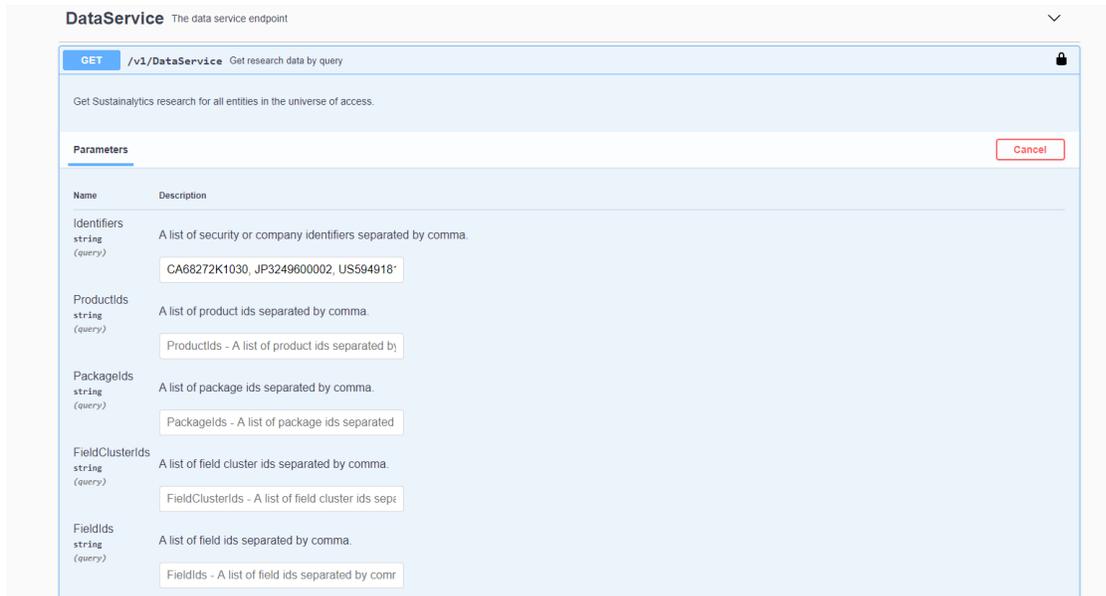
```
Inline Model [
  Inline Model 1
]
Inline Model 1 {
  Identifier (string),
  IssuerId: (string),
  IssuerName: (string),
  EntityId (integer),
  EntityName (string),
  fields (inline_model)
}
inline_model {}
```

Response Content Type:JSON

5.5.1 IDENTIFIER MATCHING

API calls to the DataService and the Reports endpoint routes allow querying data for any identifier as described in the chapter above. Depending on matching result (yes/no/multiple) and user's permissions, the API provides various response types.

As a maximum of 100 identifiers can be entered in the Identifier field in a comma separated format as denoted in the screenshot below.



Listed below are all the accepted identifiers with ISIN being the preferred identifier:

1. ISIN
2. TICKER
3. SEDOL
4. CUSIP

Depending on the type of identifier inputted by the user, the matching functionality built into the API returns research associated to the entity ID that most closely aligns and maps to the identifier entered by the user.

The table below depicts the corresponding response structures for each combination of the status message component values.

STATUS MESSAGE		JSON RESPONSE STRUCTURE
matched	hasPermissions	
YES	YES	<pre>{ Identifier: (string), IssuerId: (string), IssuerName: (string), EntityId: (integer), EntityName: (string), status: { matched: "Yes", hasPermissions: true } fields:(inline_model)</pre>

		<pre> } </pre>
YES	NO	<pre> { Identifier: IssuerId: IssuerName: EntityId: EntityName: status: { matched: "Yes", hasPermissions: false } } </pre>
MULTIPLE	YES (see additional details in table below)	<pre> { Identifier: IssuerId: IssuerName: EntityId: EntityName: status: { matched: "multiple", hasPermissions: true } fields: { } } { Identifier: IssuerId: IssuerName: EntityId: EntityName: status: { matched: "multiple", hasPermissions: true } fields: { } } </pre>
NO	n/a	<pre> { Identifier status: { matched: "No", hasPermissions: false } } </pre>

		}
--	--	---

In case of multiple matches, the API's response can have different structures, based on user's permissions.

STATUS MESSAGE		RESPONSE STRUCTURE / COMPONENTS	
matched	hasPermissions	Entity 1	Entity 2
MULTIPLE	Entity1: YES Entity2: YES	Identifier Issuer Entity Status Fields	Identifier Issuer Entity Status Fields
MULTIPLE	Entity1: YES Entity2: NO	Identifier Issuer Entity Status Fields	Identifier Issuer Entity Status
MULTIPLE	Entity1: NO Entity2: NO	Identifier Issuer Entity Status	Identifier Issuer Entity Status

As noted above, there is a slight difference in the response structure for entities with multiple matches, depending on if the parameter for *identifier* is optional or mandatory. The Field value keypairs will only be included in the response when the user makes an API request using identifiers with multiple matches within the **DataService endpoint**, whereas a similar request using identifiers with multiple matches within the **DataService/{identifier} endpoint** will return a 409 error code with a response in the following structure:

Error:

Response body

```
{
  "message": "More than one company has been found for the following identifier 'tmx'.",
  "companies": {
    "1007970368": "Teléfonos de México, S.A.B. de C.V.",
    "1008757215": "BinckBank N.V.",
    "1008761036": "Thermax Limited",
    "1008761998": "Asseco Poland S.A."
  }
}
```

6. API INTEGRATION

Users can use the Python pip package and Swagger Codegen frameworks to easily integrate API responses with their internal systems.

Python pip Package

Go to the Sustainalytics python pip package webpage to setup and configure direct access to the API using python. The page provides code snippets and sample python commands to access various end points.

```
https://pypi.org/project/sustainalytics/
```

Swagger Codegen

Listed below are steps for using Swagger Codegen -

Step 1 – Go to Swagger Codegen project page and sign in. For the “Select your product” option, click on “SwaggerHub”. Create a GitHub account to sign up or you can also sign up using an email address. There is no charge to sign up.

```
https://swagger.io/tools/swagger-codegen/
```

Step 2 – Import Sustainalytics Data Services API.

Go to left menu, “Create new” – “Import and Document API. Use the Sustainalytics API URL for the “Path or URL” input field. Click Import.

```
https://api.sustainalytics.com/swagger/docs/v1
```

The import result should be displayed like in the window below. Click “IMPORT OPENAPI”.

Import API

Update the name and version below, or keep the values from the spec.

Name: sustainalytics-api

Version: v1

✓ The resource is valid JSON
It will automatically be converted to YAML

CANCEL IMPORT OPENAPI

Step 3 – Click “Export” (top right menu) and select your desired output. There are four options available: Client SDK / Server Stub / Documentation / Download API. After clicking your output type, select your technology. There are over 25 options, technologies and frameworks: java, .NET, python, PHP etc.

7. FAQs

1) Does the API currently have any limits on requests?

Yes, we do impose limits on our API usage. Default caps are set to 50,000 requests in a day (24-hour period) for the DataService end points and 10,000 for the Reports end points.

2) How many concurrent requests can the API support?

The API can support up to 5 concurrent requests per account.

3) Can I retrieve large datasets in bulk using the API?

We do not recommend retrieving large datasets in bulk via the API. The API is designed to be used as an ad-hoc data retrieval tool. For downloading large datasets, we recommend using the Data Feeds solution that offers direct file delivery on a secure FTP location.

4) What is an API endpoint?

An API endpoint is a channel that is exposed to enable clients to directly retrieve and fetch our data, rather than us sending them data files.

Listed below are the four main API access endpoint routes that we currently offer. Each endpoint route has its own separate child endpoints that clients can further use to access API data.

1) DataService

- /v1/DataService
- /v1/DataService/{identifier}

2) ProductStructure

- /v1/FieldDefinitions
- /v1/FieldMappings
- /v1/FieldMappingDefinitions

3) Reports

- /v1/ReportService
- /v1/ReportService/{identifier}
- /v1/ReportService/{identifier}/{reportId}
- /v1/ReportService/url/{identifier}/{reportId}

4) UniverseofAccess

- /v1/UniverseOfAccess

5) How does the API service handle authorization?

Before being able to use the API endpoints, API users will need to authenticate their using a client ID and client secret. The API will then generate a token that will be valid for a period of 1 hour. The API is secured using OAuth 2.0 with Client Credentials flow.

6) How do clients get notified about any upcoming and scheduled API maintenance activities or metadata updates?

Clients are notified by email in advance of any upcoming API scheduled maintenance activities. Additionally, these notifications also get posted on the Service Announcements page within the Data Services tab on Sustainalytics Academy.

In cases of any unexpected API errors or issues, clients are also notified as soon as these get discovered.

Any changes to data points are communicated to clients at least 3 months in advance.

7) How do I request access for more product data using the API than what I currently have access to?

Your dedicated client service representative or client advisor can further assist you with your request on accessing any more products than what you may currently have access to.

8) Is the data retrieved by the API encrypted?

No, data retrieved by the API does not contain any PII or sensitive information hence it is not encrypted. The connection is secured using HTTPS.

9) Can product historical data be retrieved by the API solution?

No, historical datasets cannot be accessed by the API solution at this time. These can only be made available as a data feed on a secure FTP folder.

10) What is the date of validity of the data retrieved by the API?

Data retrieved by the API is latest data and with the same data you would find in our Global Access platform on that day.

11) Is there a preferred endpoint for retrieving PDF reports?

Yes, GET /v1/ReportService/url/{Identifier}/P{ReportId} – Get report url is the preferred endpoint for retrieving PDF reports as it has the fastest response time. It is important to note that the generated URL is valid for a 24-hour period.

12) What does the Field ID indicate?

Every datapoint is assigned a unique field ID which is a static ID that uniquely identifies a particular data field.

13) What is the geo-location of the API?

The API service has an EU and US East Coast geographic location. The service selects the nearest server based on geo-location of the IP address sending the request.

APPENDIX 1 – OPTIONAL PARAMETERS

Within many of the endpoints made available by the Sustainalytics team, users have the ability to input several optional parameters that allow for filtering by various attributes such as Identifier, Product, Fields, or by Date. Filtering will be applied based on the values entered in these parameters.

Count: If true, response will return the number of elements specified

Skip: If used, the response will skip the specified number of results. Default value is 0

Take: If used, only the specified number of results will be retrieved. Default value is 100

Code sample of how to use Skip/Take parameters

Users can apply the skip and take pagination methods, which can be used to bypass a specific number of search results and to specify the maximum number of search results to be returned.

Using the default value of 100 for 'take', the URL in the 'for' should look like this:

```
https://api.sustainalytics.com/v1/DataService?count=false&skip=%(x)s&take=100
```

The second part of the equation is knowing when to stop the loop. To do so, you would first need to identify the correct number of items, and place it in a variable from the following url:

```
https://api.sustainalytics.com/v1/DataService?count=true&skip=%s&take=100
```

This request will give you the exact count, so you can then divide it by 100 to get the number of requests+1 for your range. For example, if the universe of access has 1000 entities then $1000/100 = 10+1 = 11$ for requests will be required.

ProductIds: Refer section [1. Get Product Structure and Definitions](#)

PackageIds: Refer section [1. Get Product Structure and Definitions](#)

FieldClusterIds: Refer section [1. Get Product Structure and Definitions](#)

FieldIds: Refer section [1. Get Product Structure and Definitions](#)

LastUpdate: This parameter allows the user to call only reports that were updated after the specified date. The format of this parameter is: *yyyy-mm-dd'T'hh:mm:ss.SSSX*, in line with the **ISO8601 date format**.

Identifier: For service calls where the identifier parameter is *optional*, users can retrieve data for multiple securities within one call. The security matching service is embedded within the query, so the response contains an additional object. This object provides the user with more granular data related to the Sustainalytics' matching service and includes various components such as the issuer matched to the identifier in Sustainalytics' securities database, the entity matched to the issuer with coverage in Sustainalytics' research database, and a two-part status.

The status indicates if an identifier has been successfully 'matched' in Sustainalytics' database, and if the user account 'haspermissions' to access the data for this entity. (See section *5.4.1 Identifier Matching Rules* for further information)

Issuer = Entity matched in securities database

Entity = Entity matched in research database

In many cases the entity will be the same as the issuer, however, in some cases, it may be an entity related to the issuer, such as a parent or a sibling.

Response body

```
[
  {
    "identifier": "XS0763122578",
    "issuerId": 1107726746,
    "issuerName": "ABB Finance B.V.",
    "entityId": 1007896773,
    "entityName": "ABB Ltd",
    "status": {
      "matched": "Yes",
      "hasPermissions": true
    },
    "fields": {
      "181110101499": 0.698025625,
      "181110112399": 19.504575000000003,
      "181110121199": 18.24,
      "181110121499": -2.69476547231758,
      "181110131299": {
        "rank": 2138,
        "peers": 12273
      },
      "181110141499": -0.700000000000003,
      "181110142899": "Low",
      "181110151199": 6.47,
      "181110161299": {
        "rank": 10,
        "peers": 164
      }
    }
  }
]
```

Implementation Notes

Get Sustainalytics research for identifiers listed in the optional parameter.

Response Class (Status 200)

OK

ModelExample Value

Inline Model [

Inline Model 1

]

Inline Model 1 {

Identifier: (string),

IssuerId: (string, optional),

IssuerName: (string, optional),

EntityId: (integer, optional),

EntityName: (string, optional),

status: {

matched: (string),

hasPermissions: (Boolean)

}

fields: (inline_model, optional)

}

inline_model {

}

Response Content Type:JSON

APPENDIX 2 – API RESPONSE CODES

Status Code	Status Code Message	Behavior	Message	Example
304	Not Modified	User introduces a Last Update date as a parameter.	"Report has not changed since {lastUpdate}."	E.g. Comparison between the LastUpdate date provided by the user and the one from the Database.
400	Bad Request	Request is made for an entity that does not have research for that product.	"There is no {researchType} research for the entity with this {identifier}."	E.g. Request is made for a Core pdf report where the entity has Comprehensive research.
400	Bad Request	Request is made for a reportId that does not exist.	"'{reportId}' is not a valid fieldId."	E.g. Request is made for a different ReportId than the one available in DS Internal Product Structure.
401	Unauthorized	User does not have the right credentials.	"Please use a valid authorization token."	E.g. Incorrect Token, Token has expired.
403	Forbidden	Request is made for an identifier for which the user does not have access.	"You currently do not have access to the research for this entity, please contact your client advisor for more information'."	E.g. Request is with an identifier that links to an entity for which research is available, but the account does not have permissions for it.
403	Forbidden	User does not have permissions for the accessed reportId.	"You currently do not have access to this reporttype '{reportId}', please contact your client advisor for more information."	E.g. Account does not have permissions for that report.
403	Forbidden	Quota is set to a limited number, which is exceeded.	"You have reached the limit of {MaxPermittedRequest} requests per {duration} (d.hh:mm:ss). Please contact your client advisor for more information."	E.g. Default caps are set to 50,000 requests in a day (24 hours) for the DataService route and 10,000 for the Reports route.
404	Not Found	Request is made for an identifier which is not available in the system.	"The identifier '{identifier}' is not recognized."	E.g. Request is made for an identifier that is not available in our Database.
404	Not Found	Report is not available in Amazon S3.	"Report '{fileName}' is currently not available."	E.g. Pdf generation service has failed

409	Conflict	This code is used in situations where the user might be able to resolve the conflict and resubmit the request.	"More than one entity has been found for the following identifier '{identifier}'." "companies": { "EntityId": "EntityName", "EntityId": "EntityName" }	E.g. Identifier returns multiple conflicting values - different companies linked to the same identifier.
429	Too Many Requests	User has sent too many requests in a given amount of time ("rate limiting").	"You have reached the limit of {MaxConcurrentRequest} concurrent requests."	E.g. The API allows for 5 concurrent requests per account.
500	Internal Server Error	The server could not fulfill the request due to some unexpected condition.	"A technical error has occurred. We will look into the issue and fix it as soon as possible."	E.g. All other unhandled errors.