Mintec Import API

Use this API to import third party/supplier data and cost models in Mintec systems

Contact Info: support@mintecglobal.com
Document Version: v1.2

Introduction

The Mintec Import API is a RESTful API that provides programmatic access to allow clients to bring their own/third-party data and cost models into Mintec Analytics for further analysis and benchmarking

The API enables user imported series to be created, updated and deleted. Users can bring series one by one or in bulk into Mintec Analytics. The data points for these series can be added and updated using the API. These series can be used in the same way an imported series can today.

Cost Models can be created, updated and deleted. API enables user to bring cost models one by one or in bulk. The model components can be added and updated allowing the model to evolve and change over its life cycle. The API allows the model to be managed using familiar concepts such as Raw Materials and Packaging sections and can use all the existing representations, such as simple values, advanced options using series (mintec or imported) and formula definitions.

The REST API identifies customers using OAuth; all responses are in JSON the format.

Access

The API service endpoint will be https://public-api.mintecanalytics.com/Import

Swagger – Use below swagger end point to browse the API.
https://public-api.mintecanalytics.com/swagger/index.html

Authentication and Authorisation

The API uses OAuth 2.0 with a key that is provisioned via the user profile in Mintec Analytics to obtain a secure token and access is available over https only. The application must obtain a bearer token and submit this with every request.

The following information is required to obtain a token as an example:

Property	Value	Description
Authority	https://identity.mintecanalytics.com/connect/token	Endpoint used to authenticate the user against the service.
client_id	ebcc725e95d89b2bccf89b351471fb28	Available from Mintec Analytics, user profile, APIs Access
client_secret	ee094a61-8111-4479-bb6f-af61ca27b7ce	Generated from Mintec Analytics, user profile, APIs Access
grant_type	client_credentials	
scope	import_api	Scope must be set correctly

If you have already generated a secret key for the Mintec's Export API then you can utilise the same API key with the appropriate scope.

Auth Flow

- An application makes a request to the POST connect/token endpoint based on the Authority to exchange these credentials for a bearer token.
- When accessing the REST API, the application uses the bearer token to authenticate and is authorised to make the request.
- All requests must be made over https, any requests made over plain http will fail.

Refer to API Definitions document for access token details. Use the scope – import_api.

Methods

User series:

- Get list user series - GET/v1/import/series
- Get specific series - GET/v1/import/series/{code}
- Get specific series points - GET/v1/import/series/{code}/pointsrange
- Create new user series meta data - POST/v1/import/series
- Create new user series – POST/v1/import/series/{code}
- Create new user series points - POST/v1/import/series/{code}/points
- Update series – PUT/v1/import/series/{code}
- Update series points - PUT/v1/import/series/{code}/points
- Delete specific series - DELETE/v1/import/series/{code}
- Delete series points - DELETE/v1/import/series/{code}/points

Cost models:

- Get list of all available cost models GET/v1/import/costmodels
- Get specific cost model - GET/v1/import/costmodels/model/{code}
- Create new model - POST/v1/import/costmodels
- Create new model components - GET/v1/import/costmodels/{code}/components
- Update existing model – PUT/v1/import/costmodels/{code}
- Update cost model components - PUT/v1/import/costmodels/{code}/components
- Delete model – DELETE/v1/import/costmodels/{code}
- Delete model components- DELETE/v1/import/costmodels/{code}/components
- Rebalance the model to 100% - GET /v1/import/costmodels/model /{code}/rebalance

Additional methods:

- Get list of currencies – GET/v1/currencies
- Get list of units - GET/v1/units
- Get list of frequencies – GET/v1/frequencies

How to obtain an authentication token info and the import API

The post request to retrieve a token must set the **Content-Type** header to **application/x-www-form-urlencoded**, matching the encoding of the body of the request accordingly.

Requests will be rejected when obtaining a token if the Content-Type header is not set correctly. When using swagger or tools like postman, the content type is set automatically to match the requested body encoding.

Hints

- {code} - Should be replaced with the code of the user series or cost model required

  For example, to retrieve metadata and points for specific user series e.g. “mycoffeedata”, the request url would be formed as below.

https://public-api.mintecanalytics.com/v1/Import/series/mycoffeedata

- To utilise Mintec supported currency, frequency and unit in the API, use the "Abbr" abbreviation field is used for currency and unit representation and id or name for frequency. See currencies, frequencies and units below for details.

- Pagination support - The response is paginated in v1, set to 1000 records/page.

  Page can be indexed where the result has multiple pages, set {Pagination} to specific number (0, 1, 2, …) to specify number of pages to show. If not supplied, then default to 0.

- You can supply the frequency as an enum or the string value. For example, 2 or "Daily"

- Use the appropriate Content-Type header

Mintec Platform Data Points Import Rules

Frequency rules to adhere to when uploading user data via import API -

Example: if the date range is 1-Nov-2018 to 1-Nov-2019 then the following apply

- Daily
  - All dates except Saturday, Sunday
    e.g. 5-Nov-2018 (Mon), 6-Nov-2018 (Tue), 7-Nov-2018 (Wed), 8-Nov-2018 (Thu), 9-Nov-2018 (Fri)
- Weekly
  - All Wednesdays
    e.g. 7-Nov-2018 (Wed), 14-Nov-2018 (Wed), 21-Nov-2018 (Wed), 28-Nov-2018 (Wed)
- Monthly
  - All 1st of month
    e.g. 1-Nov-2018, 1-Dec-2018, 1-Jan-2019
- Quarterly
  - All 1st of Jan/ Apr/ Jul/ Oct
    e.g. 1-Jan-2019, 1-Apr-2019, 1-Jul-2019, 1-Oct-2019
- Annually
  - All 1st of Jan
    e.g. 1-Jan-2019

Import API Example Scenarios

The import API enables you to manage the life cycle of two important entities in Mintec Analytics.

- User Imported Series

- Cost Models

User Imported Series Walk Through

User Imported Series are series that represent prices from your own or third-party data sources, that you own or have the right to use. This data could come from many different sources, this could be a file, bespoke system, enterprise system or a third-party data provider. If you can read the data, then you can create a solution that uses the Import API to create and add points data to your own series. Series can be manually imported today, if you are familiar with the process, then you already know that the series can be used just like any other series in the Mintec Platform.

Here is an example of what a user imported series life cycle might look like.

You need to have decided on a few key points, what are the key attributes?
The code for the series, the name, specification, these will help users identify what the series represents and how-to identity it. This is what we often refer to as metadata, information about the data.

The other key decisions are what unit, currency, frequency to use, this will be driven by the data that you are importing. Having identified this information to best represent the series you can retrieve the data from the source and call the API having built the payload accordingly.

The series must exist before you can add or update points, so the first step is to create it initially. For this example, we will create the metadata and points data in the first request. Use POST providing the following JSON payload in the body of the message. Posting to the import/series will add the new series, using the code you provided.

Example Payload in the body of the message:

```
[
{

"code": "MMM01",
            "name": "My Monthly example import series",
            "specification": "MMM Special Series | Monthly Data",
            "currency": "Eur",
            "unit": "kg",
           "frequency": "4",
 "points": [
                      {
                      "value": "12",
                      "date": "2019/01/01"
                      },
                                {
                      "value": "78",
                      "date": "2019/02/01"
                      }           ,
                      {
                      "value": "45",
                      "date": "2019/03/01"
                      },
                                {
                      "value": "68",
                      "date": "2019/04/01"
                      },
                      {
                      "value": "99",
                      "date": "2019/05/01"
                      },
                                {
                      "value": "41",
                      "date": "2019/06/01"
                      },
                      {
                      "value": "4",
                      "date": "2019/07/01"
                      },
                                {
                      "value": "66",
                      "date": "2019/08/01"
                      }

                      ]
           }
]
```

You can post one or more series to the /series collection to create them.

The body of the message is JSON, so the Content-Type request header should be set to application/json to indicate this.

If the request is successful, since you are creating a new resource the result will be 201 Created, the response body will also contain this information and confirmation of the outcome:

```
{
  "code": 201,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "MMM01",
        "statusInfo": [
          {
            "code": 12,
            "message": "OK",
            "path": null
          }
```

```
            ]
         }
      ]
   },
   "data": null,
   "links": []
}
```

The result format is used across the API when creating or updating multiple resources, there is a top-level code to indicate success or partial success and a count of the success/failure/total operations. Following this is a details array which a summary of the update for each entity provided. The code of the resource that was being managed and the statusInfo array containing details of the outcome. The error code provides a result based on validation rules along with the message, which is a textual representation of the issue or successful result (ok).

From the result above, the series was created successfully, as I provided only one in the request, the result indicates the same, success is 1 out of total 1.

The resource is created I can update the metadata name, description and even change the code if required. The typical life cycle for a series is to add new or updated points.

Here is an example, since my series is defined as having a Monthly Frequency you need to provide the data points representing a month, Mintec defines monthly series as having a point on the 1st of the month (see Mintec Platform Data Points Import Rules). As the series exists you need to specify the code and the points resource; /v1/import/series/MMM01/points

POST the data in the body of the request containing the points you would like to add.

```
[
        {
        "value": "10",
        "date": "2019/09/01"
        },
        {
        "value": "10",
        "date": "2019/10/01"
        }        ,
        {
        "value": "10",
        "date": "2019/11/01"
        },
                {
        "value": "10",
        "date": "2019/12/01"
        },
        {
        "value": "99",
        "date": "2020/01/01"
        },
                {
        "value": "41",
        "date": "2020/02/01"
        },
        {
        "value": "0.0023",
        "date": "2020/03/01"
        },
                {
        "value": "66.0012",
        "date": "2020/04/01"
        }
]
```

When providing series points, you need to provide the date (in ISO format YYYY/MM/DD) and the value as a number.

To update existing point data, you can use PUT and provide an array of “Date, Value pairs” for example to update points in the example, if will PUT to v1/import/series/MMM01/points with request message body containing the points to update.

```
[
        {
        "value": "10",
        "date": "2019/09/01"
        },
        {
        "value": "10",
        "date": "2019/10/01"
        }       ,
        {
        "value": "10",
        "date": "2019/11/01"
        },
                {
        "value": "10",
        "date": "2019/12/01"
        },
        {
        "value": "99",
        "date": "2020/01/01"
        },
                {
        "value": "41",
        "date": "2020/02/01"
        },
        {
        "value": "0.0023",
        "date": "2020/03/01"
        },
                {
        "value": "66.0012",
        "date": "2020/04/01"
        }
]
```

The response in this case will indicate partial success 207, this is due to the data being outside of the exiting date range. To update data it needs to exist, however the API will apply the data update to the valid range and ignore the data outside of the existing data range. The response will reflect this and indicate how many successful changes were applied and failure count. The statusInfo will also keep you informed.

Response indicating partial success and overlap.

```
{
  "code": 207,
  "message": {
    "success": 5,
    "failure": 3,
    "total": 8,
    "details": [
      {
        "code": "MMM01",
        "statusInfo": [
          {
            "code": 17,
            "message": "Update successfully. Dates out of existing range are ignored",
            "path": [
              "Date"
            ]
          }
        ]
      }
    ]
  },
  "data": null,
  "links": [
    {
      "href": "http://(base url}/v1/import/series",
      "rel": "getall-series",
      "method": "GET"
```

Now you have seen how to create a series with points, add new points and update existing points, the other useful methods:

Review your series using GET at the series collection level using a filter

GET: {{baseUrl}}/v1/import/series?filter=MMM01

Review your series using GET at the series/{code} level

GET: {{baseUrl}}/v1/import/series/MMM01

Check the date ranges that your series covers

GET: {{baseUrl}}/v1/import/series/MMM01/pointsrange

Which will return the result in the familiar format, providing the startDate and endDate that your points data ranges over.

```
{
   "code": 200,
   "message": {
      "success": 1,
      "failure": 0,
      "total": 1,
      "details": [
         {
            "code": "MMM01",
            "statusInfo": [
               {
                  "code": 12,
                  "message": "OK",
                  "path": null
               }
            ]
         }
      ]
   },
   "data": {
      "points": {
         "startDate": "2019/01/01",
         "endDate": "2019/08/01"
      }
   },
   "links": [
      {
         "href": "http:// {base}/v1/import/series",
         "rel": "getall-series",
         "method": "GET"
      },
      {
         "href": "http://{base}/v1/import/series",
         "rel": "bulkupload-series",
         "method": "POST"
      }, .....
```

The additional links provided let you know what other actions and URL to support them.

This completes the series creations and life cycle example walk through.

Cost Models Walk Through

The creation of cost models requires two key elements, cost model metadata and components (Raw Materials, Packaging, etc) that the model is composed from. Metadata describes the key attributes, models must have a unique code and a name, unit, currency and frequency defined. It is a good idea to familiarise yourself with the cost model UI in Mintec Analytics. Let's get started, to create a new code model you will define the core information and POST the data in the body of the message to https://public-api.mintecanalytics.com/v1/import/costmodels

```
[
{
  "code": "NEWCORN01",
  "name": "Pop Corn V2 Issue 01",
  "specification": "Made from more real corn created just now",
  "currency": "EUR",
  "unit": "kg",
  "frequency": 2,
  "gtin": "2218400000",
  "countryCode": "FR",
  "modelCategory": "Dairy",
  "region": "Europe"
}
]
```

The body of the message is a JSON payload, so the **Content-Type** request header should be set to **application/json**. To utilise Mintec supported currency, frequency and unit in the API, the "Abbr" abbreviation field is used for currency and unit representation and id or name for frequency. See currencies, frequencies and units below for details.

If the model was successfully created, you will get a 201 response code, the body of the result will return a JSON payload providing additional details. See below:

```
{
   "code": 201,
   "message": {
      "success": 1,
      "failure": 0,
      "total": 1,
      "details": [
         {
            "code": "NEWCORN01",
            "statusInfo": [
               {
                  "code": 12,
                  "message": "OK",
                  "path": null
               }
            ]
         }
      ]
   },
   "data": null,
   "links": []
}
```

You have defined the metadata for the model, it does not have any components. The model will not display in the Mintec Analytics Application, until it has components defined.

To add components to the model, you will need to initially create them using POST, supply a message body detailing which section to place them (Raw Ingredients, Packaging, ..), what percentage or weight to use, the Mintec series code if using a Mintec Series and a name for the ingredient you are adding.

Add some ingredient to the Raw Materials section of the model, you will need to POST the body of the message to https://public-api.mintecanalytics.com /v1/import/costmodels/model/NEWCORN01/components

This path represents the path to the model reference by its code that was created in the initial step and addresses the components.

```
[
  {

   "compSection": 1,
   "type": 2,
   "useWeight": false,
   "series": [
                                   {
            "name": "Wheat",
            "code": "QI19",
            "type": "mintec",
            "percentage" : 35
           },
           {
            "name": "Butter",
            "code": "J134",
            "type": "mintec",
            "percentage" : 35
           }
          ]
   }
]
```

Let’s break down the request.

The “compSection” defines where we want to place these new elements in the model the 1 = Raw Materials and other sections follow the same order, so 2 = Packaging and so on.

Type represents how you want to define the items, these are:

- 1 = Simple – A value
- 2 = Advanced – Use a Series
- 3 = Formula – Use a formula

In the example “type” 2 (Advanced) is using a series

“useWeight” false tells the API to expect the series to contain percentage value to determine their contribution to the overall model.

The series are defined as an array, each series is defined by code and a name. the type within the series block indicates that the series code is for a Mintec series.

```
  "series": [
          {
            "name": "Wheat",
            "code": "QI19",
            "type": "mintec",
            "percentage": 35
           },
           {
            "name": "Butter",
            "code": "J134",
            "type": "mintec",
            "percentage": 35
           }
          ]
```

You can use other types of series, specify scope of the series, the platform will resolve the code looking for a series that matches. Please note the series must exist before you reference it in a model.

- mintec
- customised
- imported
- costmodel
- shared

Note: It is also possible to create the cost model and components in a single request, to do this the components array can add to the request payload. Below is a full example that combines the data supplied into a single step:

Create Model with Components:

```
[
          {
          "code": "NEWCORN01",
          "name": "Pop Corn V2 Issue 01",
          "specification": "Made from more real corn created just now",
           "currency": "EUR",
          "unit": "kg",
          "frequency": 2,
          "gtin": "2218400000",
           "countryCode": "FR",
          "modelCategory": "Dairy",
          "region": "Europe",
           "components":
           [
                      {
                      "compSection": 1,
                      "type": 2,
                      "useWeight": false,
                      "series": [
                                  {
                                 "name": "Wheat",
                                 "code": "QI19",
                                "type": "mintec",
                                "percentage" : 35
                                },
                                 {
                                 "name": "Butter",
                                "code": "J134",
                                 "type": "mintec",
                                "percentage" : 35
                                 }
                                ]
                  }
          ]
     }
]
```

Note: Once a model and components have been created you can update the metadata properties and components by using PUT. You can verify the current state of a model using GET, either at a collection level using https://public-api.mintecanalytics.com /v1/import/costmodels or at a model level using the code in the path:

https://public-api.mintecanalytics.com /v1/import/costmodels/model/NEWCORN01

To navigate to a cost models component with path for the model + code and components for example:

https://public-api.mintecanalytics.com
/v1/import/costmodels/model/NEWCORN01/components

To complete the walkthrough, you can update the cost models components, with a list that spans a few more sections and types. If your subscription does not have access to all the categories, substitute the series codes.

POST/v1/import/costmodels would also allow you to create cost model with components and meta data together.

Using PUT, the message body with the following payload will replace the list of components for the model.

PUT: https://public-api.mintecanalytics.com/v1/import/costmodels/model/NEWCORN01/components

```
[
{
   "compSection": 2,
   "type": 2,
   "useWeight": false,
   "series": [
                                {
           "name": "Packaging Liner",
           "code": "EX29",
           "type": "mintec",
           "percentage" : 0.56
          },
          {
           "name": "Film Covering",
           "code": "PI35",
           "type": "mintec",
           "percentage" : 2
          }
         ]
  } ,
  {
   "compSection": 1,
   "type": 2,
   "useWeight": false,
   "series": [
                                {
           "name": "Wheat",
           "code": "QI19",
           "type": "mintec",
           "percentage" : 35
          },
          {
           "name": "Butter",
           "code": "J134",
           "type": "mintec",
           "percentage" : 35
          }
          ,
          {
           "name": "Oil",
           "code": "SFOR",
           "type": "mintec",
           "percentage" : 2
          }
         ]
   },
         {


                   "compSection": 3,
            "type": 3,
                   "formula": "(@XI05 * 0.05) + 9.3"
          }
]
```

This conclude the cost model walkthrough.

Cost Model Fields Reference

The following indicates the mandatory and optional fields for a cost model, metadata definition.

Parameter	Data Type	Required /Optional	Description
code	String	Required	Cost model code. Alphanumeric. Max length: 50 characters.
name	String	Required	Cost model name. Max length: 50 characters.
specification	String	Optional	Specification.
currency	String	Required	Currency. Value list: Refer to list of currencies
unit	String	Required	Unit Value list: Refer to list of units
frequency	Enum	Required	Frequency. Value list: 2 - 6 **Note**: The String value can also be used in place of the enum 2 = Daily 3 = Weekly 4 = Monthly 5 = Quarterly 6 = Annually You can use number or name in frequency parameter E.g. “2” or “Daily
GTIN	String	Optional	Alphanumeric 30 chars. GTIN (Global Trade Item Number)
countryCode	String	Optional	countryCode must be provided, for example: DE, FR, GB, etc.
systemRefKey	String	Optional	Alphanumeric 20 chars (future)
modelCategory	String	Optional	Alphanumeric 30 chars.

region	String	Optional	Alphanumeric 30 chars.
businessUnit	String	Optional	Alphanumeric 30 chars. (future)

* future – The data may not be persisted, and the UI will not yet support the field.

Cost Model Components Fields reference

Components Fields

Parameter	Data Type	Required /Optional	Description
code	String	Required	Cost model code. Alphanumeric. Max length: 50 characters.
components	Array of objects	Required	At least one component is required.
compSection	Enum	Required	Code of Component. Value: 1 = Raw materials 2 = Packaging 3 = Transport 4 = Energy 5 = Labour 6 = Duty/Tariff 7 = Other Cost
type	Enum	Required	Type of Component. Value: 1 = Simple 2 = Advanced 3 = Formula builder
useWeight	Boolean		Determine whether portion of series to the component is to be specified by weight or percentage. Required if ComponentType = 2 AND (ComponentCode = 1 OR 2). Value: True/False. If UseWeight is True, series must supply a weight field. Otherwise, series must supply a percentage field and value.
percentage	Number		Component percentage. Applicable if ComponentType = 1. Positive number =<100. If IsPercentage is True, Percentage is required. Otherwise, set Percentage as Null.
costValue	Number		Component value. Applicable if ComponentType = 1. Positive number. If IsPercentage is False, CostValue is required. Otherwise, set CostValue as Null.
series	Array of object		
name	String		User's defined name. Max length: 50 chars. Required if ComponentType = 2.
code	String		Series code. Max length: 50 chars. Required if ComponentType = 2.

			Alphanumeric. Max length: 50 chars.
type	String		Values: mintec, customised, imported, costmodel, shared. Used to match specified series with existing series in MA.
weight	Number		Series weight. Applicable if ComponentType = 2 AND (ComponentCode = 1 OR 2). Must be a positive number. If Useweight is True, SeriesWeight is required. Otherwise, set SeriesWeight as Null.
percentage	Number		Series weight. Applicable if ComponentType = 2 AND (ComponentCode = 1 OR 2). Must be a positive number =<100. If Useweight is False, SeriesPercentage is required. Otherwise, set SeriesPercentage as Null.
factor	Number		Series factor. Required if ComponentType = 2 AND (ComponentCode = 3 OR 4 OR 5 OR 6 OR 7). Max length: 15 chars.
formula	String		Formula builder. Required if ComponentType = 3.

Method Details

GET Https://public-api.mintecanalytics.com/v1/import/series

Returns list all available user series in Mintec Analytics

Return type
JSON response

Path parameters
N/A

Query parameters
Filter by Code/Name/Description (optional)

Query Parameter — filter Enter some text e.g. Coffee to search for all series which contains that text in series Code or Name or Description. If not supplied, then no filter will be applied.

To change the number of items per page use **pagesize**
To page trough the results use **pageindex** (zero based)

For example: /v1/import/series?filter=CoDe2&pageSize=2&pageindex=2

Example data for response

```
{
```

```
"code": 200,
"message": {
  "success": 6,
  "failure": 0,
  "total": 6,
  "details": [
    {
      "code": null,
      "statusInfo": [
        {
          "code": 12,
          "message": "OK",
          "path": null
        }
      ]
    }
  ]
},
"data": {
  "items": [
    {
      "code": "Tuna",
      "name": "Tuna",
      "specification": null,
      "currency": "US Dollar",
      "unit": "Kilogram",
      "frequency": "Daily",
      "points": [
        {
          "date": "2017/09/15",
          "value": 1234
        },
        {
          "date": "2017/09/18",
          "value": 1234
        },
        {
          "date": "2017/09/19",
          "value": 1234
        },
        {
          "date": "2017/09/20",
          "value": 1234
        },
        {
          "date": "2017/09/21",
          "value": 1234
        },
```
Content-Type: application/json

Responses

200
OK

Refer to Error Codes section more details

GET Https://public-api.mintecanalytics.com/v1/import/series/{code}

Returns metadata and data points for a single user series. Get individual series code based on the provided Code.

Path parameters

code (required)
Path Parameter — Series code of the series to retrieve. Should form part of the URL

Return type
JSON response

Example data for response

```
 {
"code": "MySalmon",
"name": "Salmon Norway",
"specification": null,
"currency": "Norway",
"unit": "Kilogram",
"frequency": 2,
"points": [
 {
  "date": "2017/09/15",
  "value": 1234
 },
 {
  "date": "2017/09/18",
  "value": 1234
 },
 {
  "date": "2017/09/19",
  "value": 1234
 },
 {
  "date": "2017/09/20",
  "value": 1234
 },
 {
  "date": "2017/09/21",
  "value": 1234
 },
 {
  "date": "2017/09/22",
  "value": 1234
```

```
      },
```
Content-Type: application/json

Responses
200
OK

Use GET/v1/import/series/{code}/pointsrange to return the date range of the points available for specific series.

```
{
  "code": 200,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "Salmon Norway",
        "statusInfo": [
          {
            "code": 12,
            "message": "OK",
            "path": null
          }
        ]
      }
    ]
  },
  "data": {
    "points": {
      "startDate": "2017/09/15",
      "endDate": "2017/12/15"
    }
  },
```

Refer to Error Codes section more details

POST Https://public-api.mintecanalytics.com/v1/import/series

Create new user series into Mintec Analytics. This end point will also allow you insert series in bulk. Use below end point of you need to create specific series POST Https://public-api.mintecanalytics.com/v1/import/series/{code}

Return type
JSON response

Path parameters
N/A

Query parameters
N/A

Example data

```
[
  {
    "code": "IMPORT01",
    "name": "Import series",
    "specification":null,
    "currency": "GBP",
    "unit": "kg",
    "frequency": "Daily"
    "points": [
      {
        "date": "2020/05/13",
        "value": 10
      }
    ]
  }
]
```

Content-Type: application/json

Date Format is: YYYY/MM/DD.

Responses
201
OK

Refer to Error Codes section more details

POST Https://public-api.mintecanalytics.com/v1/import/series/{code}/points

Add series points to the series {code}, where {code} is the user imported series.

Path parameters
code (required)
Path parameter – code of the series to import into Mintec Analytics.

Query parameters
code (required)
Query Parameter — Code of the series to import

Example data

```
[
 {
    "date": "2020/05/12",
    "value": 10
 }
 {
    "date": "2020/05/13",
    "value": 10
  }

]
```

Content-Type: application/json

Responses
201
OK

Refer to Error Codes section more details

PUT Https://public-api.mintecanalytics.com/v1/import/series/{code}

This end point could be used to modify user series or to update the imported series

Path parameters
code (required)
Path parameter – code of the series to import into Mintec Analytics.

Query parameters
code (required)
Query Parameter — Code of the series to import

Example data

```
{
  "code": "wheatdata",
  "name": "mywheatsupplierdata",
  "specification": "wheatmilling",
  "currency": "1",
  "unit": "21",
  "frequency": 2,
  "points": [
    {
      "date": "2020/01/01",
      "value": 150
    }
  ]
}
```

Content-Type: application/json

Responses
200
OK

Refer to [Error Codes]() section more details

PUT Https://public-api.mintecanalytics.com/v1/import/series/{code}/points

This end point could be used to update the data points for specific series code. This will allow to add/modify data points in history or to continue line in Mintec Analytics.

Path parameters
code (required)
Path parameter – code of the series to modify or update .

Query parameters
code (required)
Query Parameter — Code of the series to modify or update

Example data

```
[
  {
    "date": "2020/10/03"
    "value": 100
  }
  {
    "date": "2020/10/04"
    "value": 200
  }

]
```
Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

GET https://public-api.mintecanalytics.com/v1/import /series/{code}/pointsrange

Returns the data range as dates start – end

Path parameters
code (required)
Path parameter – code of the series to manage points data

Query parameters
N/A

Example data for response
```
[
{
   "code": 200,
   "message": {
      "success": 1,
      "failure": 0,
      "total": 1,
      "details": [
         {
            "code": "MMM01",
            "statusInfo": [
               {
                  "code": 12,
                  "message": "OK",
                  "path": null
               }
            ]
         }
      ]
   },
   "data": {
      "points": {
         "startDate": "2019/01/01",
         "endDate": "2019/08/01"
      }
   },
]
```
Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

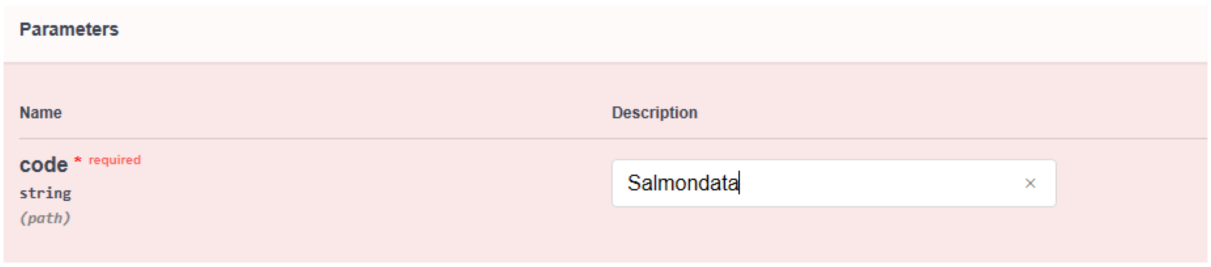
DELETE Https://public-api.mintecanalytics.com/v1/import/series/{code}

Removes the specific user series and the points data

Path parameters
code (required)
Path parameter – code of the series to delete from Mintec Analytics.

Query parameters
code (required)
Query Parameter — Code of the series to delete

Example data


Parameters

Name	Description
code * required string (path)	Salmondata

```
{
  "code": 200,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "Salmondata",
        "statusInfo": [
          {
            "code": 19,
            "message": "Delete successfully.",
            "path": null
          }
        ]
      }
    ]
  },
  "data": null,
  "links": []
}
```

Content-Type: application/json

Responses
200
OK

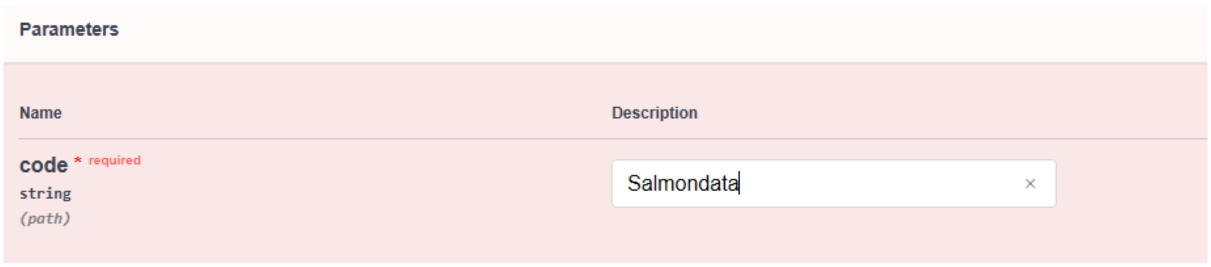
Refer to Error Codes section more details

DELETE Https://public-api.mintecanalytics.com/v1/import/series/{code}/points

Delete user series and removes all points data
Path parameters
code (required)
Path parameter – code of the series to delete

Query parameters
code (required)
Query Parameter — Code of the series to delete

Example data



Parameters

Name	Description
code * required string *(path)*	Salmondata

```
{
  "code": 200,
  "message": {
    "success": 262,
    "failure": 0,
    "total": 262,
    "details": [
      {
        "code": "Salmondata",
        "statusInfo": [
          {
            "code": 19,
            "message": "Delete successfully.",
            "path": null
          }
        ]
      }
    ]
  },
  "data": null,
  "links": []
}
```
Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

Import Cost models –

GET Https://public-api.mintecanalytics.com/v1/import/costmodels

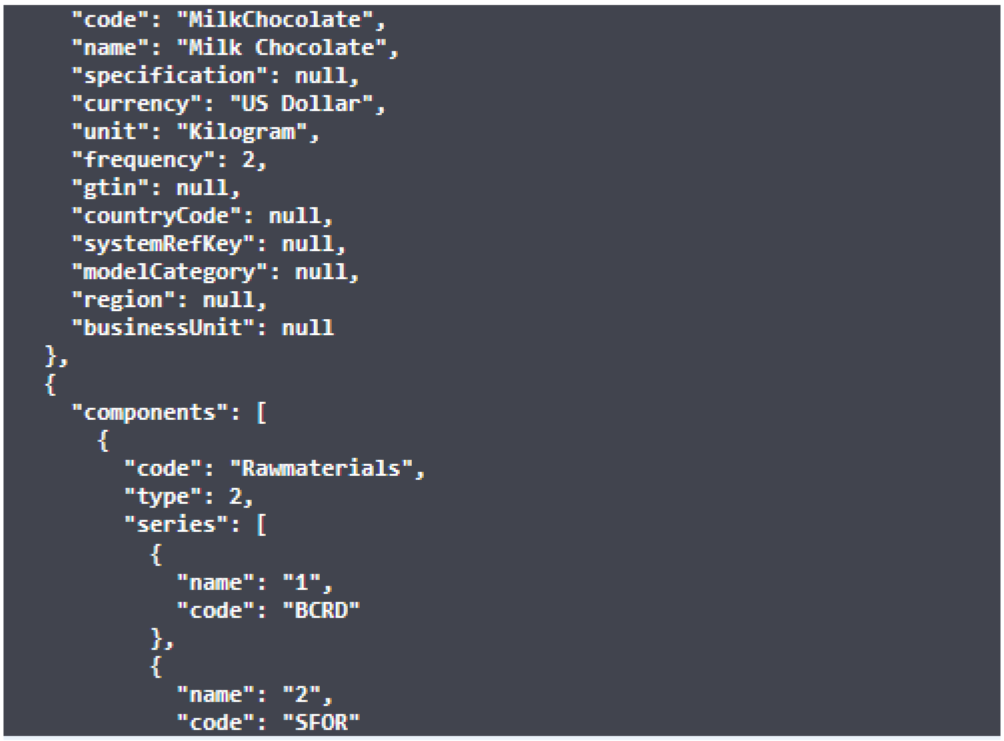
Get list of existing cost models in Mintec Analytics system

Path parameters
N/A

Query parameters
Filter by Code (optional)

Query Parameter — {filter} Enter some text e.g. MilkChocolate to search for all model which contains that text in model Code. If not supplied, then no filter will be applied.

Example data for response



```
      "code": "MilkChocolate",
      "name": "Milk Chocolate",
      "specification": null,
      "currency": "US Dollar",
      "unit": "Kilogram",
      "frequency": 2,
      "gtin": null,
      "countryCode": null,
      "systemRefKey": null,
      "modelCategory": null,
      "region": null,
      "businessUnit": null
    },
    {
      "components": [
        {
          "code": "Rawmaterials",
          "type": 2,
          "series": [
            {
              "name": "1",
              "code": "BCRD"
            },
            {
              "name": "2",
              "code": "SFOR"
```

Content-Type: application/json

Responses

200

OK

Refer to Error Codes section more details

GET Https://public-api.mintecanalytics.com/v1/import/costmodels/model/{code}

Return specific model in Mintec Analytics

Path parameters

code (required)

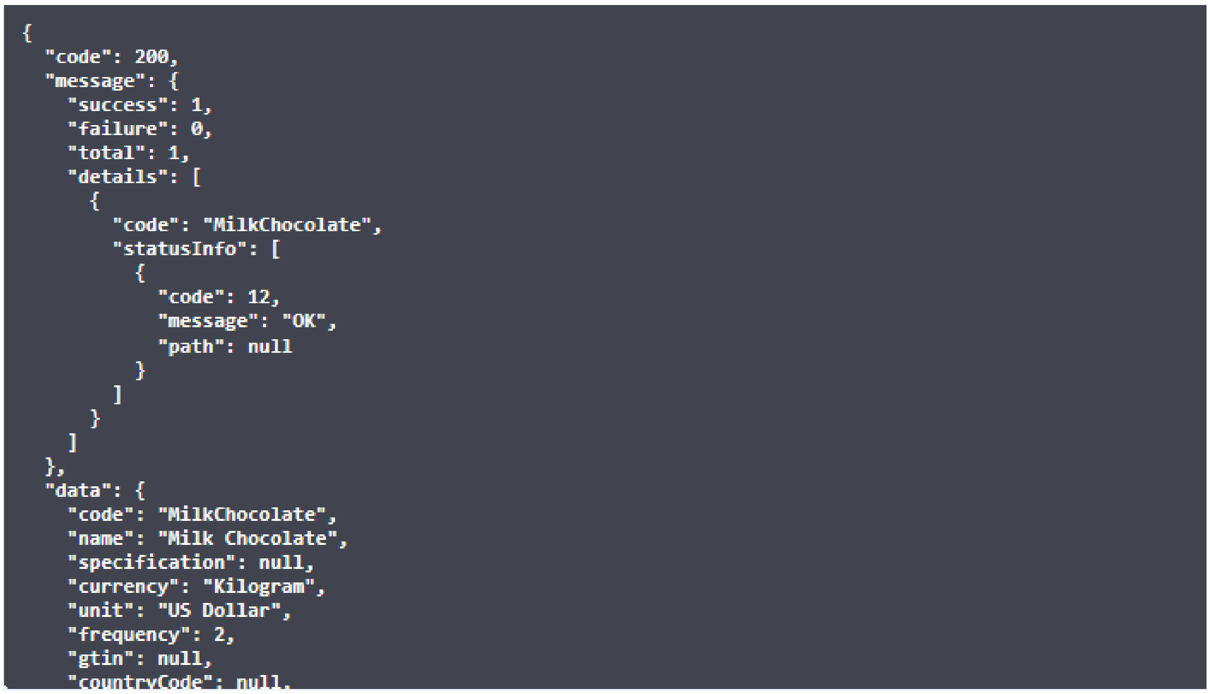
Path parameter – code of the model available in Mintec Analytics.

Query parameters

code (required)

Query Parameter — Code of the series available in Mintec Analytics.

Example data for response



```
{
  "code": 200,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "MilkChocolate",
        "statusInfo": [
          {
            "code": 12,
            "message": "OK",
            "path": null
          }
        ]
      }
    ]
  },
  "data": {
    "code": "MilkChocolate",
    "name": "Milk Chocolate",
    "specification": null,
    "currency": "Kilogram",
    "unit": "US Dollar",
    "frequency": 2,
    "gtin": null,
    "countryCode": null,
```

Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

POST Https://public-api.mintecanalytics.com/v1/import/costmodels

Use this end point to create one or more cost models in Mintec Analytics

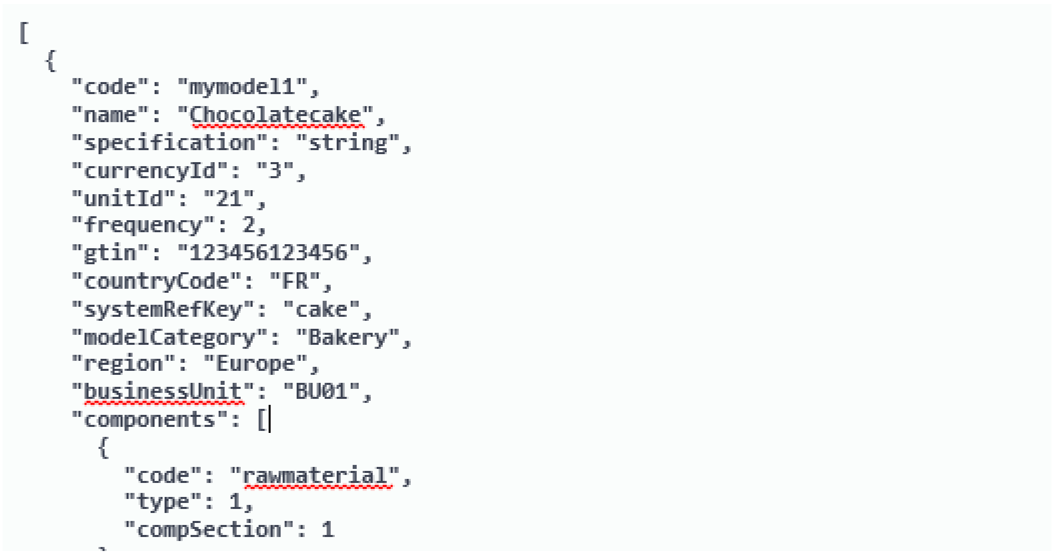
Path parameters

N/A

Query parameters

N/A

Example data



```
[
  {
    "code": "mymodel1",
    "name": "Chocolatecake",
    "specification": "string",
    "currencyId": "3",
    "unitId": "21",
    "frequency": 2,
    "gtin": "123456123456",
    "countryCode": "FR",
    "systemRefKey": "cake",
    "modelCategory": "Bakery",
    "region": "Europe",
    "businessUnit": "BU01",
    "components": [
      {
        "code": "rawmaterial",
        "type": 1,
        "compSection": 1
```

Content-Type: application/json

Responses
201
OK

Refer to Error Codes section more details

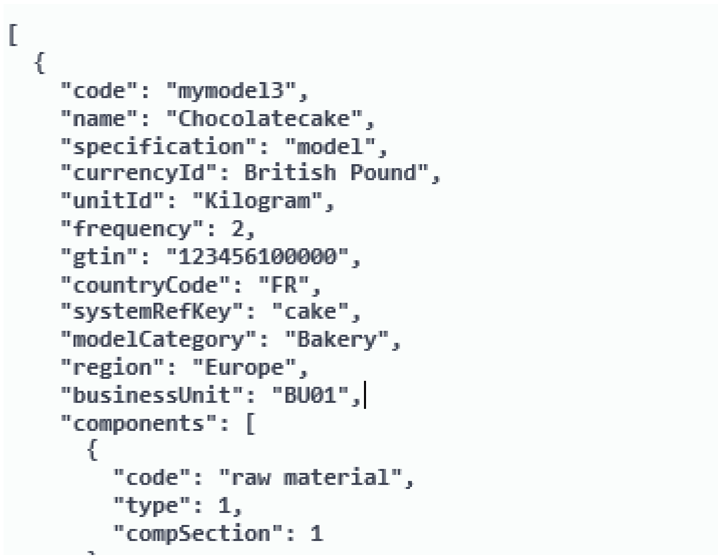
PUT Https://public-api.mintecanalytics.com/v1/import/costmodels/model/{code}

Update specific model information using PUT

Path parameters
code (required)
Path parameter – code of the model available in Mintec Analytics.

Query parameters
code (required)
Query Parameter — code of the model available in Mintec Analytics

Example data



```
[
  {
    "code": "mymodel3",
    "name": "Chocolatecake",
    "specification": "model",
    "currencyId": British Pound",
    "unitId": "Kilogram",
    "frequency": 2,
    "gtin": "123456100000",
    "countryCode": "FR",
    "systemRefKey": "cake",
    "modelCategory": "Bakery",
    "region": "Europe",
    "businessUnit": "BU01",|
    "components": [
      {
        "code": "raw material",
        "type": 1,
        "compSection": 1
```

Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

PUT Https://public-api.mintecanalytics.com/v1/import/costmodels/{code}/components

Update the components for cost models

Path parameters
code (required)
Path parameter – code of the model to update the components

Query parameters
N/A

Example data

```
[
 {

   "compSection": 1,
   "type": 2,
   "useWeight": false,
   "series": [
                         {
          "name": "Wheat",
          "code": "QI19",
          "type": "mintec",
          "percentage" : 35
         },
         {
          "name": "Butter",
          "code": "J134",
          "type": "mintec",
          "percentage" : 35
         }
        ]
  }
]
```
Content-Type: application/json

Responses

200

OK

Refer to [Error Codes](#) section more details

GET Https://public-api.mintecanalytics.com/v1/import/costmodels/model/{code}/components

Get a list of existing components for specific cost model

Path parameters

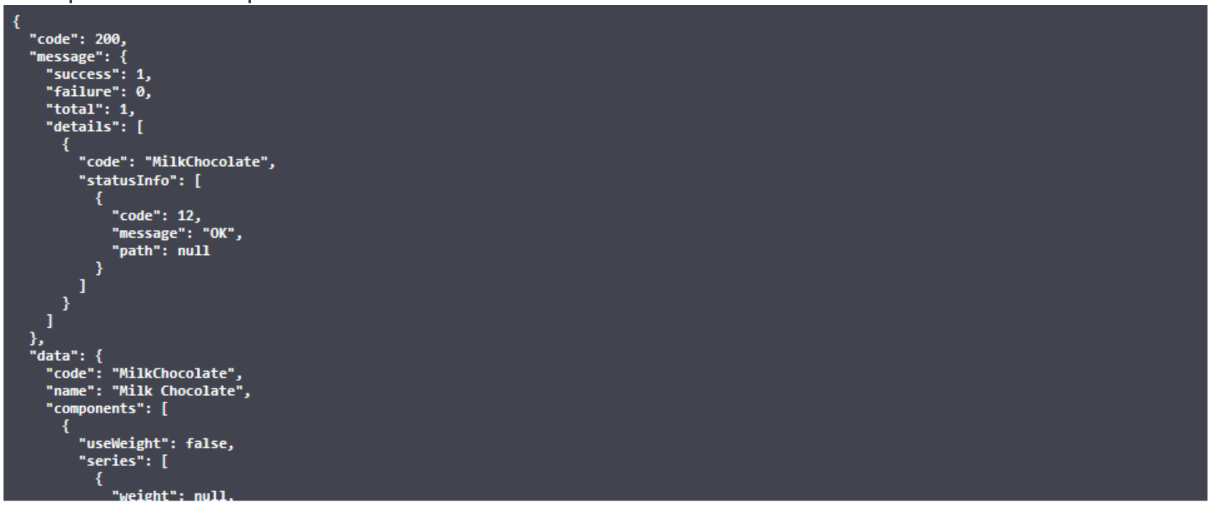
code (required)

Path parameter – code of the model

Query parameters

N/A

Example data for response



```
{
  "code": 200,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "MilkChocolate",
        "statusInfo": [
          {
            "code": 12,
            "message": "OK",
            "path": null
          }
        ]
      }
    ]
  },
  "data": {
    "code": "MilkChocolate",
    "name": "Milk Chocolate",
    "components": [
      {
        "useWeight": false,
        "series": [
          {
            "weight": null,
```

Content-Type: application/json

Responses

200

OK

Refer to Error Codes section more details

DELETE Https://public-api.mintecanalytics.com/v1/import/costmodels/{code}

Delete cost model. Deleting the model will remove any components that are part of the model

Path parameters

code (required)

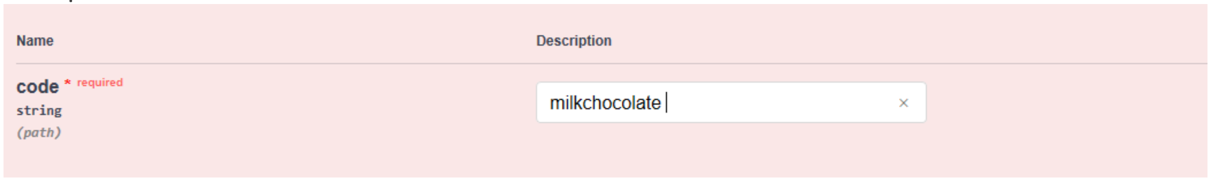
Path parameter – code of the model to delete

Query parameters

code (required)

Query Parameter — Code of the model to delete

Example data



Name	Description
code * required string *(path)*	milkchocolate

```
{
  "code": 200,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "milkchocolate",
        "statusInfo": [
          {
            "code": 19,
            "message": "Delete successfully.",
            "path": null
          }
        ]
      }
    ]
  },
  "data": null,
  "links": []
}
```

Content-Type: application/json

Responses

200

OK

Refer to Error Codes section more details

DELETE Https://public-api.mintecanalytics.com/v1/import/costmodels/model/{code}/components

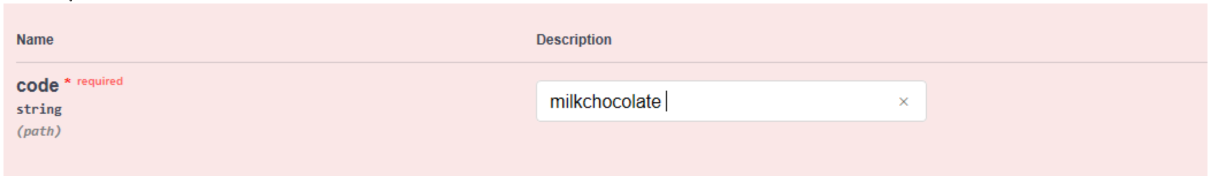
Removes the components for specific model

Return type
JSON response

Path parameters
code (required)
Path parameter – code of the model

Query parameters
code (required)
Query Parameter — Code of the model to delete model and its components

Example data



Name	Description
code * required string *(path)*	milkchocolate

```
{
  "code": 200,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "milkchocolate",
        "statusInfo": [
          {
            "code": 19,
            "message": "Delete successfully.",
            "path": null
          }
        ]
      }
    ]
  },
  "data": null,
  "links": []
}
```

Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

GET Https://public-api.mintecanalytics.com/v1/import/costmodel/model/{code}/rebalance

Use this to rebalance specific cost model to 100%

Return type

JSON response

Example data for response

```
{
  "code": 200,
  "message": {
    "success": 1,
    "failure": 0,
    "total": 1,
    "details": [
      {
        "code": "FP0022",
        "statusInfo": [
          {
            "code": 12,
            "message": "OK",
            "path": null
          }
        ]
      }
    ]
  },
  "data": {
    "code": "FP0022",
    "name": "Fish Pie",
    "components": [
      {
        "useWeight": true,
        "series": [
          {
            "weight": 4.68,
            "percentage": null,
            "name": "Cod",
            "code": "TS11",
            "type": "mintec"
          },
          {
            "weight": 2.34,
            "percentage": null,
            "name": "Mash",
            "code": "POBE",
            "type": "mintec"
          },
          {
            "weight": 2.34,
            "percentage": null,
            "name": "Italian Tomato",
            "code": "7S24",
            "type": "mintec"
          },
          {
            "weight": 0.38,
            "percentage": null,
            "name": "Salt L1",
            "code": "IH41",
            "type": "mintec"
          },
          {
            "weight": 8.57,
            "percentage": null,
            "name": "Cheese",
            "code": "JS22",
            "type": "mintec"
          }
        ],
        "code": "Rawmaterials",
        "type": 2
      },
      {
        "useWeight": false,
        "series": [
          {
            "weight": null,
            "percentage": 8.42,
            "name": "Box1",
```
Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

GET Https://public-api.mintecanalytics.com/v1/curriences

Returns a list of supported currencies. Use the short name “abbr” when defining currencies in series or cost models. For example, “USD” for US Dollar.

Return type
JSON response
Example data for response
```
{
"content": [
  {
   "id": 1,
   "name": "US Dollar",
   "abbr": "USD"
  },
  {
   "id": 3,
   "name": "British Pound",
   "abbr": "GBP"
  },
```
Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

GET Https://public-api.mintecanalytics.com/v1/frequencies

Returns a list of supported Frequencies. Use the id or name for series or cost models, they can be used interchangeably. For example: “Daily” to define a daily frequency or the integer 2.

Return type
JSON response

Example data for response

```
{
 "content": [
  {
   "id": 2,
   "name": "Daily"
  },
  {
   "id": 3,
   "name": "Weekly"
  },
```
Content-Type: application/json

Responses
200
OK

Refer to Error Codes section more details

GET Https://public-api.mintecanalytics.com/v1/units

Return a list of supported units. Use the "abbr" abbreviation short name for units in series and cost models. For example, "MT" for Metric Tonne.

Return type
JSON response

Example data for response

```
{
 "content": [
  {
   "id": 1,
   "name": "Metric Tonne",
   "abbr": "MT"
  },
  {
   "id": 21,
   "name": "Kilogram",
   "abbr": "kg"
  },
```
Content-Type: application/json

Responses
200
OK

Error Codes

The API returns appropriate standard response codes and statuses for every request.

Code	Text	Description
200	OK	Success, standard response.
201	Created	Standard response for a request when an item is created.
207	Multi-Status	Outcome of multiple resources, an array of responses with codes indicating the outcome of each resource request.
400	Bad Request	The request was invalid or cannot be otherwise served.
401	Unauthorized	Authentication credentials were missing or incorrect.
403	Forbidden	The request is understood, but it has been refused or access is not allowed.
404	Not Found	The URI requested is invalid or the resource requested.
429	Too Many Requests	Returned when a request cannot be served due to the application's rate limit having been exhausted for the resource.
500, 501, 502, etc.	Internal Server Error	Something is broken. Please contact Mintec Support so the Mintec team can investigate.