

# Importing data into Everest using the Web API

Technical Documentation

Exported on 07/12/2022

## Table of Contents


1 Overview.....	3
2 Option 1: JSON POST - /api/B2B/ProcessExtraction.....	4
3 Option 2: File POST - /api/B2B/ProcessExtraction .....	5
4 Option 3: <Any> POST - /api/B2B/ProcessExtractionData?codeName=<import code name> .....	6
5 Option 4: JSON Post to InvokeExtraction - /api/B2B/InvokeExtraction .....	7

# 1 Overview

This article will give you a couple examples of how you can post data to the Everest Web API and have that data imported into Everest. There are a few methods described below but all of them have the following prerequisites:

1. **Authentication** - The requests will need to be authenticated as an Everest user that is authorized to perform the *ImportData* operation. See [this article](#)<sup>1</sup> for some example code for authentication.
2. **Import Configuration** - An import will need to be configured via the Everest Data Imports Manager with the correct handler type for the data being posted (eg CSV, JSON, XML). You will need to know the CodeName from the Data Import Manager to post to the API.

- [Overview](#)(see page 3)
- [Option 1: JSON POST - /api/B2B/ProcessExtraction](#)(see page 4)
- [Option 2: File POST - /api/B2B/ProcessExtraction](#)(see page 5)
- [Option 4: JSON Post to InvokeExtraction - /api/B2B/InvokeExtraction](#)(see page 7)

 Option #3 is recommended if you are on a version of Everest that supports it (> 10.1)

---

<sup>1</sup> <https://lift.allvuesystems.com/display/Everest/Everest+Web+API+Authentication>

## 2 Option 1: JSON POST - /api/B2B/ProcessExtraction

This API (at <http://<yourEverestUrl>/api/B2B/ProcessExtraction>) allows posting of JSON-formatted data and requires an import already be configured to process the data using our JSON handler. For the sample code below the CodeName from the Data Import Manager for that import is **Security Master (Generic JSON)**.

```
{
  "extractionPackageCodeName" : "Security Master (Generic JSON)",
  "data" : [{
    "Asset_EverestID" : 4260,
    "Asset_Name" : "Test Loan A - API"
  }, {
    "AssetID_CUSIP" : "TBB12345",
    "Asset_Name" : "Test Bond B - API"
  }
]
```

Note that the JSON data inside the data array in this example will be able to be directly imported using our Generic Security Master specification. The same structure, with different field names can be used to comply with our other generic specification.


### 3 Option 2: File POST - **/api/B2B/ProcessExtraction**

Using the same endpoint as Option 1 above (at *http://<yourEverestUrl>/api/B2B/ProcessExtraction*) you can post an entire file. The file contents using by POSTing a form-style request with the following form fields:

- **DataBytes** - The file with the data that you're trying to import
- **DataName** - The file name of the file you are uploading
- **extractionPackageCodeName** - The CodeName from Data Import Manager of your import (eg. **Security Master (Generic JSON)** in the example above.

## 4 Option 3: <Any> POST - /api/B2B/ProcessExtractionData?codeName=<import code name>

With this API you can post any type of data (XML, CSV, JSON, etc) and have it processed by the import engine.

 This API is available as of v10.1

To perform the same import as Option 1 with this API you would use the URL `http://<yourEverestUrl>/api/B2B/ProcessExtractionData?codeName=Security%20Master%20%28Generic%20JSON%29` and post the following JSON data (note that it is the contents of the *data* array from Option 1

```
[{
  "Asset_EverestID" : 4260,
  "Asset_Name" : "Test Loan A - API"
}, {
  "AssetID_CUSIP" : "TBB12345",
  "Asset_Name" : "Test Bond B - API"
}]
```

This option also allows other formats to be posted. For example, if we had an import setup to process XML (CodeName=Security Master XML you would POST the following to `http://<yourEverestUrl>/api/B2B/ProcessExtractionData?codeName=Security%20Master%20XML`

```
<Extraction>
  <Row>
    <Asset_EverestID>4260</Asset_EverestID>
    <Asset_Name>Test Loan A - API</Asset_Name>
  </Row>
  <Row>
    <AssetID_CUSIP>TBB12345</AssetID_CUSIP>
    <Asset_Name>Test Bond B - API</Asset_Name>
  </Row>
</Extraction>
```

You could do the same with the following CSV (using a CSV import):

```
Asset_EverestID,AssetID_CUSIP,Asset_Name
4260,, "Test Loan A"
,TBB12345,"Test Bond B - API"
```

## 5 Option 4: JSON Post to InvokeExtraction - /api/B2B/InvokeExtraction

This API (at <http://<yourEverestUrl>/api/B2B/InvokeExtraction>) can be used to kick off an existing import's listener. The JSON Post'ed instructs B2B which package to run and what parameters to use during the execution.

- **codeName** or **packageKey** - The CodeName from Data Import Manager of your import (eg. **SecurityMaster (SQL Listener)** in the example below.
- **ignoreQueue** - Should this import proceed with the
- **listenOnlyOnce** - Instruct the listener to poll only a single time for data
- **parameters** (optional) - an array of parameters to be passed to the SQL Listener as arguments.

Similar to Option 1, you would post the following JSON to kick off a specific extraction package:


```
{
  "codeName": "SecurityMaster (EDM Listener)",
  "ignoreQueue": true,
  "listenOnlyOnce": true,
  "parameters": [
    {
      "Name": "AsOfDate",
      "Value": "2017-05-12"
    }
  ]
}
```

For those instances where you need to produce the JSON from SQL, the following SQL query will produce the above JSON message:

```
SELECT
  codeName='SecurityMaster (EDM Listener)',
  ignoreQueue=CAST('true' AS BIT),
  listenOnlyOnce=CAST('true' AS BIT),
  [parameters] = (SELECT Name='AsOfdate', Value='2017-05-12' FOR JSON PATH)
FOR JSON PATH, WITHOUT_ARRAY_WRAPPER
```

### Sample SQL Listener configuration:

Name	Value
SQL	Exec EDM.pExtractSecurityMaster @AsOfDate=@AsOfDate
IsXMLQuery	true
OutputCSV	false
param:@AsOfDate (enter a default value if desired)	2016-12-31

 If the endpoint responds with a 404 but you've correctly entered the URL, check to see that the ImportData role is available to the api user.