**Reprocessing Data Baseline**

**API User Manual**

Reference RRPP-API-0013-CS

Issue/Revision 1.1

Date of Issue 14/04/2022

Status Issued

|  |  |  |
| --- | --- | --- |
| Description | Version | Date |
| Issued | 1.0 | 17/06/2021 |
| Limitations section added | 1.1 | 14/04/2022 |
|  |  |  |

**Change Log:**

Table of contents



[1. Introduction 4](#_Toc100850054)

[1.1 Purpose and scope 4](#_Toc100850055)

[1.2 Documents 4](#_Toc100850056)

[1.2.1 Reference documents 4](#_Toc100850057)

[1.2.2 Applicable documents 4](#_Toc100850058)

[1.3 Terminology 5](#_Toc100850059)

[1.3.1 Definitions 5](#_Toc100850060)

[1.3.2 List of Acronyms 5](#_Toc100850061)

[2. OVERVIEW 6](#_Toc100850062)

[3. INSTRUCTION SECTION 7](#_Toc100850063)

[3.1 Access point 7](#_Toc100850064)

[3.2 User registration 7](#_Toc100850065)

[3.3 Access token 7](#_Toc100850066)

[3.3.1 Via Postman 8](#_Toc100850067)

[3.3.2 Via Curl and Python for developers 9](#_Toc100850068)

[3.4 Requesting for reprocessing data baseline 9](#_Toc100850069)

[3.5 Requesting examples 10](#_Toc100850070)

[4. Limitations 12](#_Toc100850071)

[5. APPENDICES 13](#_Toc100850072)

[5.1 Appendix A – HTTP Status and Error Messages 13](#_Toc100850073)

[5.2 Appendix B – $metadata 13](#_Toc100850074)

# Introduction

## Purpose and scope

This document is the Reprocessing Data Baseline API user manual, it contains the necessary instructions for getting started with this service, the main functionality exposed by this API is getReprocessingDataBaseline function allowing clients to retrieve the applicable up-to-date auxiliary data files from the reprocessing preparation package database allowing users to perform reprocessing campaigns.

## Documents

### Reference documents

1. OData Documentation <http://www.odata.org/documentation>
2. OData Protocol <http://docs.oasis-open.org/odata/odata/v4.01/odata-v4.01-part1-protocol.html>

### Applicable documents

1. REPROCESSING REFERENCE PACKAGE PREPARATION

## Terminology

### Definitions

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **API** | A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service. |
| **ADF** | Auxiliary Data File used in a processing chain |
|  |  |

### List of Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Signification** |
| **AD** | Applicable Document |
| **UUID** | Universally Unique Identifier – Allows to identify an instance |
| **JSON** | JavaScript Object Notation, commonly used format in internet services |
| **URI** | Uniform Resource Identifier |

# OVERVIEW

The Reprocessing Data Baseline main functionality is to identify for each Level-0 product name and Level-1/2 product type to be reprocessed, the applicable current list of auxiliary data files to be used for reprocessing campaigns.

This document aims to teach new users how to use the Reprocessing Data Baseline API; it contains the following main instructions:

* The access point of the Reprocessing Data Baseline API.
* User registration.
* Access token generation.
* Data Requesting via the main functionality **getReprocessingDataBaseline**()

# INSTRUCTION SECTION

## Access point

As a part of the Reprocessing Preparation Package, Reprocessing Data Baseline API can be accessed via the following URL:

[**https://reprocessing-auxiliary.copernicus.eu/rdb.svc**](%20https://reprocessing-auxiliary.copernicus.eu/rdb.svc)

## User registration

New users of Reprocessing Data Baseline API should be registered first in order to be able to use the API and request for the accessToken.

|  |  |
| --- | --- |
|  |  |

Figure 1 : Sign in & registration Interface

## Access token

To get the Reprocessing Data Baseline API token for a client, an HTTP POST request should be sent to the following Token resource:

[**https://reprocessing-auxiliary.copernicus.eu/auth/realms/reprocessing-preparation/protocol/openid-connect/token**](https://reprocessing-auxiliary.copernicus.eu/auth/realms/reprocessing-preparation/protocol/openid-connect/token)

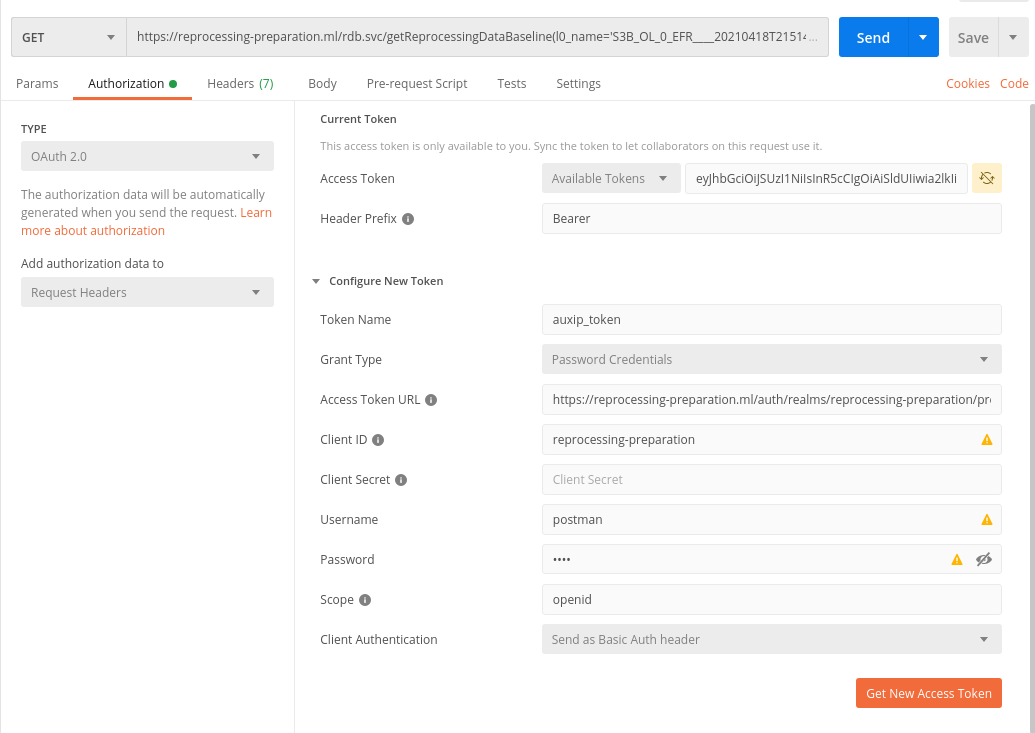
In the post body, username and password are specified in JSON format, and the response body contains a token key with an actual API Token as the value. The token should be used in an HTTP Authorization header while communicating with the Reprocessing Data Baseline service.

It is necessary to get a token using a tool (a browser or any other application that can send http requests). Depending on the client that is used, there are different ways to send HTTP headers.

This chapter covers the following examples:

* Sending requests via Postman
* Curl
* Python

## Via Postman



4

21A

8

11A

7

6

5

3

Figure 2 : Getting New Access Token via Postman

To be able to send requests via Postman, one should configure the authorization settings first, as showing in the above steps:

* Steps 2 and 3 : choose authorization via OAuth 2.0
* Steps 4 to 7 :
  + Access Token URL : set the the resource where to post for the accessToken,
  + Client ID : should be set to “**reprocessing-preparation**”
  + Username and Password : Client credentials from the registration.
  + Scope : **openid**
* Step 8 : Get New Access Token

## Via Curl and Python for developers

* **Curl**

RAWTKN=$(curl -s -X POST \

-H "Content-Type: application/x-www-form-urlencoded" \

-d "username="$USER \

-d "password="$PASSWORD \

-d 'grant\_type=password' \

-d "client\_id=reprocessing-preparation" \

https://reprocessing-auxiliary.copernicus.eu/auth/realms/reprocessing-preparation/protocol/openid-connect/token \

|jq . )

TOKEN=$(echo $RAWTKN | jq -r '.access\_token')

echo "Token to be used to access to our service : ${TOKEN} "

* **Python**

import requests

def get\_token\_info(user,password):

headers = {'Content-Type': 'application/x-www-form-urlencoded'}

data={"username":user, "password":password,"client\_id":" reprocessing-preparation","grant\_type":"password"}

token\_endpoint="https://reprocessing-auxiliary.copernicus.eu/auth/realms/reprocessing-preparation/protocol/openid-connect/token"

token\_endpoint="https://reprocessing-auxiliary.copernicus.eu/auth/realms/reprocessing-preparation/protocol/openid-connect/token"

response = requests.post(token\_endpoint,data=data,headers=headers)

return response.json()

## Requesting for reprocessing data baseline

The Reprocessing Data baseline API is an ODATA Restfull API, which gives access to the data baseline via only one unbounded function called ***getReprocessingDataBaseline***(), this function is overloaded and can be called in the following two ways :

* With a comma separated list of level 0 products names + mission + unit + product\_type :
  + [https://reprocessing-auxiliary.copernicus.eu/rdb.svc/getReprocessingDataBaseline(l0\_names='L0\_ProductName\_1, L0\_ProductName\_2,L0\_ProductName\_3',mission='mission',unit='unit',product\_type='product\_Type')](https://reprocessing-auxiliary.copernicus.eu/rdb.svc/getReprocessingDataBaseline(l0_names='L0_ProductName_1,%20L0_ProductName_2,L0_ProductName_3',mission='mission',unit='unit',product_type='product_Type'))
* With a sensing time interval (start and stop) + mission + unit + product\_type :
  + [https://reprocessing-auxiliary.copernicus.eu/rdb.svc/getReprocessingDataBaseline(start='sensing\_start\_datetime',stop='sensing\_stop\_datetime', ,mission='mission',unit='unit',product\_type='product\_Type')](https://reprocessing-auxiliary.copernicus.eu/rdb.svc/getReprocessingDataBaseline(start='sensing_start_datetime',stop='sensing_stop_datetime',%20,mission='mission',unit='unit',product_type='product_Type'))

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Example** |
| l0\_names | Comma separated list of level 0 products names | 'S3A\_OL\_0\_EFR\_\_\_\_20201108T205651\_20201108T205851\_20201108T220426\_0119\_065\_014\_\_\_\_\_\_LN1\_O\_NR\_002.SEN3' |
| mission | Mission to be considered while retrieving auxiliary data files | Can be one of the following :  'S2MSI', 'S1SAR', 'S3OLCI', 'S3MWR', 'S3SRAL', 'S3SYN', 'S3SLSTR' |
| unit | Platform Serial Identifier | 'A', 'B' |
| product\_type | L1/L2 type of the product to be reprocessed.  This type is mission dependent. | Depending of the mission  S1SAR : { 'L1SLC', 'L1GRD', 'L2OCN' }  S2MSI : { 'L1A', 'L1B', 'L1C', 'L2A' }  S3MWR : { 'L1CAL', 'L1MWR' }  S3OLCI : {'L1EFR', 'L1ERR' , 'L2LFR', 'L2LRR' }  S3SLSTR : { 'L1RBT', 'L2LST', 'L2FRP' }  S3SRAL : { 'L1CAL', 'L1SRA', 'L2LAN' }  S3SYN : { 'L1MISR', 'L2' } |
| start | Sensing start date time | 2019-10-06T18:00:00Z |
| stop | Sensing start date time | 2020-11-06T19:00:00Z |

Table 1 : getReprocessingDataBaseline parameters

## Requesting examples

The Query of applicable up to date reprocessing auxiliary data function is achieved through standard Odata API Function, the output of this function is list of auxiliary data files with AUXIP access cloud links.

Here after a calling list of the function ***getReprocessingDataBaseline*** showing the three use cases:

* With only one level 0 product name :

<https://reprocessing-auxiliary.copernicus.eu/rdb.svc/getReprocessingDataBaseline(l0_names='S1A_IW_RAW__0NDV_20201001T062556_20201001T063833_034599_040733_8B28.SAFE.zip',mission='S1SAR',unit='A',product_type='L1SLC')>

* With a list of level-0 product names :

<https://reprocessing-auxiliary.copernicus.eu/rdb.svc/getReprocessingDataBaseline(l0_names='S1A_IW_RAW__0NDV_20201001T062556_20201001T063833_034599_040733_8B28.SAFE.zip,S1A_IW_RAW__0SDV_20201124T101416_20201124T101448_035389_04229A_9816.SAFE.zip',mission='S1SAR',unit='A',product_type='L1SLC')>

* With a sensing time interval :

<https://reprocessing-auxiliary.copernicus.eu/rdb.svc/getReprocessingDataBaseline(start=2019-10-06T18:00:00Z,stop=2020-11-06T19:00:00Z,mission='S3OLCI',unit='A',product_type='L1EFR')>

**Example of the request response:**

{

"@odata.context": "$metadata#Collection(OData.CSC.DataBaseline)",

"value": [

{

"Level0": "S3B\_OL\_0\_EFR\_\_\_\_20201029T230119\_20201029T230319\_20201030T000536\_0119\_045\_115\_\_\_\_\_\_LN1\_O\_NR\_002.SEN3",

"AuxDataFiles": [

{

"Name": "S3B\_OL\_1\_RAC\_AX\_20180425T000000\_20991231T235959\_20190320T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_002.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(dd00d348-4b22-4676-bd36-9366f9635b0d)/$value"

},

{

"Name": "S3B\_OL\_1\_SPC\_AX\_20180425T000000\_20991231T235959\_20190320T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_002.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(8e540397-bb17-4e8d-885f-756bd8c54b9d)/$value"

},

{

"Name": "S3B\_OL\_1\_PCPBAX\_20180425T000000\_20991231T235959\_20180409T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_001.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(3a366ccb-1d48-46ba-90d4-d342d60139ee)/$value"

},

{

"Name": "S3B\_OL\_1\_EO\_\_AX\_20180618T000000\_20991231T235959\_20180621T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_003.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(bf7ee491-c85d-42e9-9e4b-ba48c8b75de5)/$value"

},

{

"Name": "S3B\_OL\_1\_PRG\_AX\_20180618T000000\_20991231T235959\_20180621T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_003.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(2cf1ed0f-b7e6-4fef-a213-26e869a5849e)/$value"

},

{

"Name": "S3B\_OL\_1\_CLUTAX\_20180425T000000\_20991231T235959\_20180409T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_001.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(c2fb3a58-75f0-4383-bbbd-69b00e86c5c9)/$value"

},

{

"Name": "S3B\_OL\_1\_CAL\_AX\_20200321T171527\_20991231T235959\_20200331T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_010.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(eb3de5bd-b4bb-49c5-9812-0a353f4c9c86)/$value"

},

{

"Name": "S3B\_OL\_1\_INS\_AX\_20200301T000000\_20991231T235959\_20200331T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_006.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(f28871bd-78e0-45fe-ac31-788ed76ffd21)/$value"

}

]

},

{

"Level0": "S3B\_OL\_0\_EFR\_\_\_\_20201104T001454\_20201104T001654\_20201104T011621\_0119\_045\_187\_\_\_\_\_\_LN1\_O\_NR\_002.SEN3",

"AuxDataFiles": [

{

"Name": "S3B\_OL\_1\_RAC\_AX\_20180425T000000\_20991231T235959\_20190320T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_002.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(dd00d348-4b22-4676-bd36-9366f9635b0d)/$value"

},

{

"Name": "S3B\_OL\_1\_SPC\_AX\_20180425T000000\_20991231T235959\_20190320T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_002.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(8e540397-bb17-4e8d-885f-756bd8c54b9d)/$value"

},

{

"Name": "S3B\_OL\_1\_PCPBAX\_20180425T000000\_20991231T235959\_20180409T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_001.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(3a366ccb-1d48-46ba-90d4-d342d60139ee)/$value"

},

{

"Name": "S3B\_OL\_1\_EO\_\_AX\_20180618T000000\_20991231T235959\_20180621T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_003.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(bf7ee491-c85d-42e9-9e4b-ba48c8b75de5)/$value"

},

{

"Name": "S3B\_OL\_1\_PRG\_AX\_20180618T000000\_20991231T235959\_20180621T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_003.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(2cf1ed0f-b7e6-4fef-a213-26e869a5849e)/$value"

},

{

"Name": "S3B\_OL\_1\_CLUTAX\_20180425T000000\_20991231T235959\_20180409T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_001.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(c2fb3a58-75f0-4383-bbbd-69b00e86c5c9)/$value"

},

{

"Name": "S3B\_OL\_1\_CAL\_AX\_20200321T171527\_20991231T235959\_20200331T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_010.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(eb3de5bd-b4bb-49c5-9812-0a353f4c9c86)/$value"

},

{

"Name": "S3B\_OL\_1\_INS\_AX\_20200301T000000\_20991231T235959\_20200331T120000\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MPC\_O\_AL\_006.SEN3.zip",

"AuxipLink": "https://reprocessing-auxiliary.copernicus.eu/auxip.svc/Products(f28871bd-78e0-45fe-ac31-788ed76ffd21)/$value"

}

]

},

# Limitations

For all services, a GET request should not exceed 2000 characters and the result is limited to 200 filenames (the “skip” parameter can be used to get the following results).

# APPENDICES

## Appendix A – HTTP Status and Error Messages

The GET response of applicable products list is an HTTP response to the GET reprocessing data baseline request. As recommended by the OData specification, the Reprocessing Data Baseline OData service supports responses in JSON.

The following overall HTTP status codes may be returned with the response:

* 200 OK: if the request is accepted and a response can be returned
* 400 Bad Request
* 401 Unauthorized: if the requesting client is unauthorised
* 404 Not Found
* 500 Internal Server Error

## Appendix B – $metadata

<?xml version="1.0" encoding="UTF-8"?>

<edmx:Edmx Version="4.0" xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">

<edmx:DataServices>

<Schema xmlns="http://docs.oasis-open.org/odata/ns/edm" Namespace="OData.CSC">

<EntityType Name="DataBaseline">

<Property Name="Level0" Type="Edm.String"></Property>

<Property Name="AuxDataFiles" Type="Collection(OData.CSC.Product)"></Property>

</EntityType>

<ComplexType Name="Product">

<Property Name="Name" Type="Edm.String" Nullable="false"></Property>

<Property Name="AuxipLink" Type="Edm.String" Nullable="false"></Property>

</ComplexType>

<Function Name="getReprocessingDataBaseline">

<Parameter Name="l0\_names" Type="Edm.String" Nullable="false"></Parameter>

<Parameter Name="mission" Type="Edm.String" Nullable="false"></Parameter>

<Parameter Name="unit" Type="Edm.String" Nullable="false"></Parameter>

<Parameter Name="product\_type" Type="Edm.String" Nullable="false"></Parameter>

<ReturnType Type="Collection(OData.CSC.DataBaseline)"/>

</Function>

<Function Name="getReprocessingDataBaseline">

<Parameter Name="start" Type="Edm.DateTimeOffset" Nullable="false"></Parameter>

<Parameter Name="stop" Type="Edm.DateTimeOffset" Nullable="false"></Parameter>

<Parameter Name="mission" Type="Edm.String" Nullable="false"></Parameter>

<Parameter Name="unit" Type="Edm.String" Nullable="false"></Parameter>

<Parameter Name="product\_type" Type="Edm.String" Nullable="false"></Parameter>

<ReturnType Type="Collection(OData.CSC.DataBaseline)"/>

</Function>

<EntityContainer Name="Container">

<FunctionImport Name="getReprocessingDataBaseline" Function="OData.CSC.getReprocessingDataBaseline" IncludeInServiceDocument="true"></FunctionImport>

</EntityContainer>

</Schema>

</edmx:DataServices>

</edmx:Edmx>