

G. BDDS REST API Documentation

This document describe the way to use the Rest API for the BDDS application. Rest API exposed here will give the response in the Json .

On the API, if the list of product endpoints are called without the token, it will only return the list of public product and if the valid token is passed, it will return the aggregated list of public and subscribed products for the corresponding user

Below is the set up instructions which will help with all the details for the API

1. Authentication

Prior to any action to get the list of subscribed products and to download, it is required to authenticate via the UserId/Password credentials to get the valid authentication token which is valid for **60 mins**. Authentication is based on a standard OAuth 2 grant type: Password Credentials. End User will need to use his credential's and need to make a call to the OKTA URL to get the proper token. Bearer token that is to be returned by the OKTA authentication endpoint will be valid for 1 hours. After expiry time of such token , end user need to request a new token.

Type : oauth2

Flow : application

Token URL : <https://login.epo.org/oauth2/aus3up3nz0N133c0V417/v1/token>

Here user name would be the EPO Email ID (Example as akhtar1508@gmail.com) and password would be your login password

To get the OKTA token , it requires to have the valid client ID (**0oa3updn7an5pMI8O417**) in the of base 64 encoded format . Base 64 encoded value of given client ID is **MG9hM3VwZG43YW41cE1JOE80MTc=** and same has to be passed in Authorization header along as Basic authentication as mentioned as part of CURL command .

Corresponding CURL Command :

```
curl --location --request POST 'https://login.epo.org/oauth2/aus3up3nz0N133c0V417/v1/token' \
--header 'Authorization: Basic MG9hM3VwZG43YW41cE1JOE80MTc=' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=password' \
--data-urlencode 'username=*****' \
--data-urlencode 'password=*****' \
--data-urlencode 'scope=openid'
```

Another way of Doing in Postman

(In the Authorization header need to pass the value as **Basic MG9hM3VwZG43YW41cE1JOE80MTc=**)

POST <https://login.epo.org/oauth2/aus3up3nz0N133c0V417/v1/token...> Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/> grant_type	password			
<input checked="" type="checkbox"/> username	akhtar1508@gmail.com			
<input checked="" type="checkbox"/> password				
<input checked="" type="checkbox"/> scope	openid			
Key	Value	Description		

Body Cookies (1) Headers (23) Test Results  200 OK 1099 ms 2.65 KB Save Response

Response from Token endpoint

```
code
{
  "token_type": "Bearer",
  "expires_in": 3600,
  "access_token": "eyJraWQioiJXS2ZQTjJnYUppRzVoUwlWZDNrc3oyY2E2dFB6T1FyZ2xLMVVLN2lTNDQ0IiwiYWxnIjoiUlMyNTYifQ.
eyJ2XXiOjEsImp0aSI6IkFULkFyc1JFV05HazdCT2ZhazKdmRBUVpfecnA2Umg5YWNN3cFJHd1dSOUTjOFEiLCJpc3MiOiJodHRwczovL2xvZ2lu
LmVbwy5vcmcvb2F1dGgyL2F1czN1cDNuejBOMTMzYzBNWDE3IiwiYXVkJoiYXBwOi8vYmRkcyIsImhdCI6MTY1NDY5MDYxNiwiZXhwIjoxNjU0
Njk0MjE2LCJjaWQiOiIwb2EzdXBkbjdhbjVwTUk4TzQxNyIsInVpZCI6IjAwdTN3YzhiMzM1ZGd4N29iNDE3Iiwick2NwIjpbIm9wZW5pZCjdLCJh
dXRox3RpBWUiOjE2NTQ20TA2MTYsInN1YiI6ImFraHRhcjE1MDhAZ21haWwuY29tIn0.hL79tT3ZoP_i54totyVStbaRIMhkuh588izocCVKDGe
-aCDsck6Wg7jUr0uptLBh3YIrtp03-yDSRqQY48PfJ9w1jikOeMRZ06LaOL0loG8nlrYCBKVySeQu5z96C8FK_PlCNXOvmDDm_KUEXJ-
LN0duoVzprKJMjx7aVw7g96HPlj5DmcUe-oZ9CcWpGf7p-
cSvKeLSVrlbePK8ZcWle4131WbIJTNDbfzQ1tUjhMRLK9B4JfoZPMvW4VNYowspa4_uVom1pIJCFFPp3765cYc6zkXX1Uhlj4qNdDmtdmnHpmBEf
SSWpolkQq5XB2Up48AiMPAFgfeo0vsuX38mTQ",
  "scope": "openid",
  "id_token": "eyJraWQioiJXS2ZQTjJnYUppRzVoUwlWZDNrc3oyY2E2dFB6T1FyZ2xLMVVLN2lTNDQ0IiwiYWxnIjoiUlMyNTYifQ.
eyJzdWIiOiIwMHUzd2M4YjMzNWReDdvYjQxNyIsInZlcIi6MSwiaXNzIjoiHR0cHM6Ly9sb2dpbi5lcG8ub3JnL29hdXR0Mi9hdXMzdXAbnow
TjEzM2MwVjQxNyIsImf1ZCI6IjBvYTN1cGRuN2FuNXBNSThPNDE3IiwiawFOIjoxnJU0NjkwnjE2LCJ1eHaiOjE2NTQ20TQyMTYsImp0aSI6Ik1E
LmdENz1lBMTA5cUM4TXE2QnQ4VmY2QXRpz1lyU19CUi1DckRuNkN5Wvoxd3ciLCJhbXiiolsicHdkI10sIm1kcCI6IjAwb2Vlb2Vyc1g0b2hGVEpz
NDE2IiwiYXV0aF90aW11IjoxnJU0NjkwnjE2LCJhdF9oYXNoIjoiEVJReF9Bcm510w1DbDVWXRcdGJNQSJ9.
2DuK1pbklQeIcXnCA_d19a7k1rzmSBE089BWenRCGc7FnysdFHts0_3ZGtoZ8VnCG--
m_10zbr6ecYvJy7GhH3o4aKMPEvZikZpkTW0ZPwkCb9WT2xeymooxlWibKdjbysaybHbbaWj4bC_bW41m5WeypJAPVx7igZThqrUJPx2uKiosobW
1jZF3swWUsdpbNzCfqSTw5Ij9MWG1i4XkhyhMkbxoGNwr_I2Ubzs5mwbxKp7dW604dqt0npFfBp-
aY0Hi41ICSpDccLoNyHXVgq3NHMPTW0XQuwgfvCql8jv_UzbtG1gl6fHBZ1if3go-hsK2U5sq9oXUjXrkKThDgA"
}
```

Sample JAVA Program to Get the authentication Token

code

```
public final static String USERNAME = "****"; // need to pass user own EPO email ID
public final static String PASSWORD = "*****"; // need to pass user own EPO password
public final static String TOKEN_URL = "https://login.epo.org/oauth2/aus3up3nz0N133c0V417/v1/token";
public final static String CLIENT_ID = "0oa3updn7an5pMI80417";

private static TokenResponse getAccessToken() {
    MultiValueMap<String, String> body = new LinkedMultiValueMap<> ();
    body.put("grant_type", Collections.singletonList("password"));
    body.put("username", Collections.singletonList(USERNAME)); // OKTA Email ID
    body.put("password", Collections.singletonList(PASSWORD)); // OKTA Password
    body.put("scope", Collections.singletonList("openid"));
    ResponseEntity<TokenResponse> responseEntity =
        new RestTemplate ().exchange(
            TOKEN_URL, // Token URL
            HttpMethod.POST,
            new HttpEntity<Object> (body, getTokenForPasswordFlow()),
            TokenResponse.class);
    TokenResponse res = responseEntity.getBody();
    return res;
}

private static HttpHeaders getTokenForPasswordFlow() {
    HttpHeaders httpHeaders = new HttpHeaders();
    httpHeaders.setContentType( MediaType.APPLICATION_FORM_URLENCODED );
    byte[] encodedAuth = Base64.getEncoder ().encode ( CLIENT_ID.getBytes () ); // Application Client ID
    String authHeader = "Basic " + new String(encodedAuth, Charset.defaultCharset());
    httpHeaders.set("Authorization", authHeader);
    return httpHeaders;
}
```

Once the token is granted, these token must be used in the subsequent request as **Bearer token** and should be passed in as Authorization header

2. To get the list of products

GET <https://publication-bdds.apps.epo.org/bdds/bff-service/prod/api/products/>

Note: Authorization header must be passed with valid token in the request header, so that list of subscribed product can be assessed. If the token is not passed, above endpoint will just return the public products

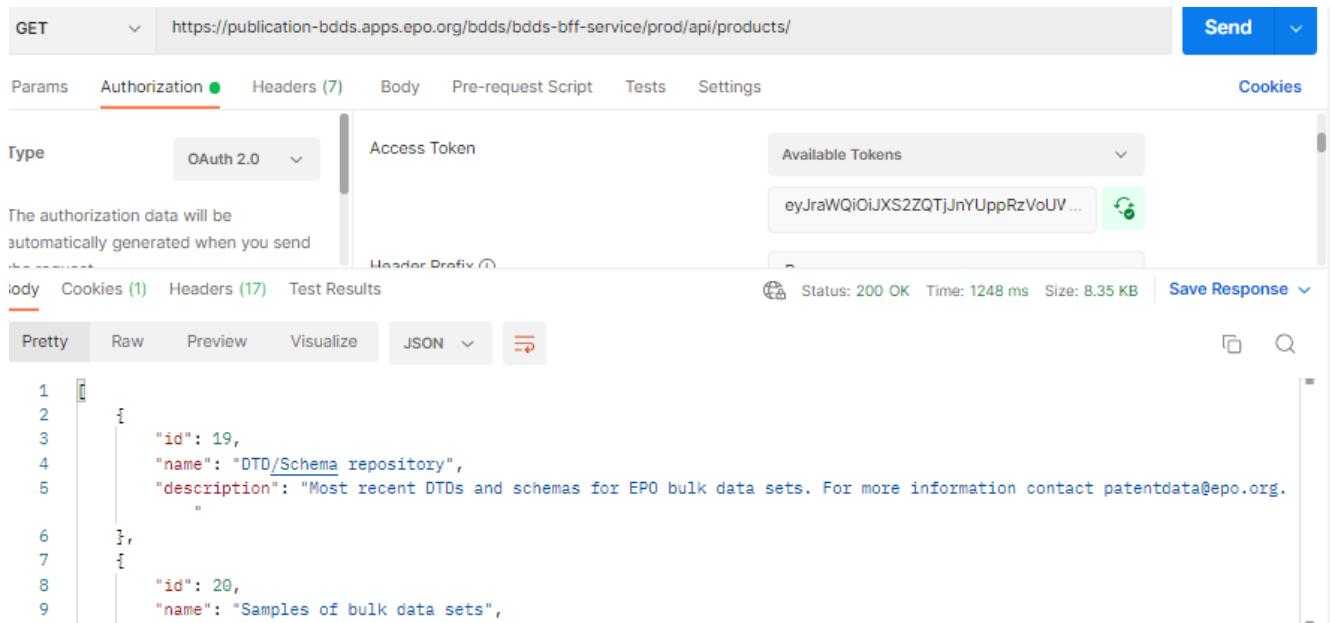
Ex: Authorization = **Bearer** eyJraWQiOiJXS2ZQTjNjYUppRzVoUWlWZDNrc3oyY2E2dFB6T1FyZ2xLMVVNLn2iTNDQ0liwiYWxnljoiUIMyNTYifQ.eyJ2ZXIiOjEslmp0aSl6lkFULkpJQUUp2aG1Sb2JrbTlvV292clFDN1hNeWpCV0gtN1d2U0lzSm1mNnRhbfkILCJpc3MiOiJodHRwczovL2xvZ2luLmVwby5vcmcvb2F1dGgyL2F1czN1cDNuejBOMTMzYzBWNDE3liwiYXVkljo1YXBwOi8vYmRkcylsmihdCi6MTY1NDY5NjQ2NywiZXhwlijoxNjU0NzAwMDY3LCJiaWQiOilwb2EzdXBkbjdhbjVwTUk4TzQxNyIslnVpZCI6ljAwdTN3YzhiMzM1ZGd4N29iNDE3liwic2Nwljpblm9wZW5pZCJdLCJhdXRoX3RpBWUjOjE2NTQ2OTY0NjcslnN1Yi6lmFraHRhcjE1MDHAZ21haWwuY29tn0.GCCbwYZNuNqibg4kiAA6HaGxzOcXQRNEU-RYZkJL8IYTo9xDpfXK-vLsJwlqV96kMfkOQuCuDC4aM4xR6Tilx5gxDrIq1D1DBfwga7OKIOD9RAq85C9Miq9LJKqmvVgWuNbSlnTl6nI91x3A411MN1JPbpDdVngch9p49uM4izUNJwPBo7bkQucbsVw35-J9-o1u6IJS74Hf_UTWubLxe3ngyKOiFW8wKnyQfGzSN43qHw-hkW7fvU2s376VOMfnvylMz_oIYhoRCNyIK2U8skJwcWQeJInE0X8mtoJ6zfA3UraN5Klhgx1Ua4rmVQDrI6EWM17eKJe7fp3-A

Response from this would be like below

code

```
[  
  {  
    "id": 19,  
    "name": "DTD/Schema repository",  
    "description": "Most recent DTDs and schemas for EPO bulk data sets. For more information contact patentdata@epo.org."  
  },  
  {  
    "id": 20,  
    "name": "Samples of bulk data sets",  
    "description": "This section contains EPO bulk data product samples. For more information contact patentdata@epo.org."  
  },  
  {  
    "id": 12,  
    "name": "Statistical mapping: CPC to FI, FI to CPC and CPC to IPC",  
    "description": "The tables are produced by using a statistical analysis of symbols allocated to families of documents present in DOCDB and classified simultaneously in the CPC, IPC and FI schemes. \nSame content is also available also here: https://www.epo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc/cpc-fi.html\nhttps://www.epo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc/fi-cpc.html\nepo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc/ipccpc.html\nFor more information contact patentdata@epo.org. \n"  
  },  
  {  
    "id": 28,  
    "name": "Unitary Patent (UP) sample packages",  
    "description": "See README file in each package for more details.\nWe will be adding packages from all bulk data products as we receive them. \nFor any questions contact support@epo.org\n"  
  }  
]
```

Similar Example in Postman to get all the list of subscribed products



The authorization data will be automatically generated when you send the request.

Access Token

Available Tokens

eyJraWQiOiJSZQTjJnYUppRzVoUV...

Status: 200 OK Time: 1248 ms Size: 8.35 KB Save Response

```
Pretty Raw Preview Visualize JSON
```

```
1 [  
2   {  
3     "id": 19,  
4     "name": "DTD/Schema repository",  
5     "description": "Most recent DTDs and schemas for EPO bulk data sets. For more information contact patentdata@epo.org."  
6   },  
7   {  
8     "id": 20,  
9     "name": "Samples of bulk data sets",  
10    "description": "This section contains EPO bulk data product samples. For more information contact patentdata@epo.org."  
11  },  
12  {  
13    "id": 12,  
14    "name": "Statistical mapping: CPC to FI, FI to CPC and CPC to IPC",  
15    "description": "The tables are produced by using a statistical analysis of symbols allocated to families of documents present in DOCDB and classified simultaneously in the CPC, IPC and FI schemes. \nSame content is also available also here: https://www.epo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc/cpc-fi.html\nhttps://www.epo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc/fi-cpc.html\nepo.org/searching-for-patents/helpful-resources/first-time-here/classification/cpc/ipccpc.html\nFor more information contact patentdata@epo.org. \n"  
16  },  
17  {  
18    "id": 28,  
19    "name": "Unitary Patent (UP) sample packages",  
20    "description": "See README file in each package for more details.\nWe will be adding packages from all bulk data products as we receive them. \nFor any questions contact support@epo.org\n"  
21  }  
]
```

Similar Example with Curl Command

```

curl --location --request GET 'https://publication-bdds.apps.epo.org/bdds/bdds-bff-service/prod/api/products/' \
--header 'Authorization: Bearer
eyJraWQiOijXS2ZQTjJnYUppRzVoUWlWZDNrc3oyY2E2dFB6T1FyZ2xLMVVLN21tTNDQ0IiwiYWxnIjoiUlMyNTYifQ.
eyJ2ZXIiOjEsImp0aSI6IkFULkpJQUp2aG1Sb2JrbTlv292c1FDN1hNeWpCV0gtN1d2U0IzSmlmNnRhbFkiLCJpc3MiOiJodHRwczovL2xvZ2lu
LmVwby5vcmcvb2F1dGgyL2F1czN1cDNuejBOMTMzYzBWNDE3IiwiYXVkJoiYXBwOi8vYmRkcyIsImlhdCI6MTY1NDY5NjQ2NywiZXhwIjoxNjU0
NzAwMDY3LCJjaWQiOiwB2EzdXBkbjdhbjVwTUk4TzQxNyIsInVpZC16IjAwdTN3YzhMzM1ZGd4N29iNDE3Iiwc2NwIjpbIm9wZW5pZCjdLCJh
dXRoX3RpBWUoje2NTQ2OTY0NjcsInN1YiI6ImFraHRhcjE1MDhAZ21haWwuY29tIn0.GCCbwYZNuNqibg4kiAA6HaGxzOcXQRNEU-
RYZkjl8IYT09xDpfXK-vLsJwIqv96kMfkQuCuD-
C4aM4xR6T1lx5gxDr1q1D1DBfwgja7OKIOD9RAq85C9Miq9LJKqmVgWuNbSlnTI6nl91x3A411MN1JPbpDdVnghc9p49uM4izUNJwPBo7bkQucb
sVw35-J9-olu6IJSt74Hf_UTWubLxe3ngyKOilFW8wKnyQfGzSN43qHw-
hkW7fvU2s376VOMfnvyliMz_oILYhoRCNy1k2U8skJwcWQeJInE0X8mtoJ6zfa3UraN5Klhgx1ua4rmVQDrI6EWM17eKJe7fp3-A'

```

3. To get the details of product along with delivery and files

3. To Get the details of all the products along with all the delivery and Items (here Product id will be product id retrieve in the list of product calls). It will return all the available delivery and items for the corresponding product to download

Note: Delivery Date and item publication data is returned by the **API is UTC format**

GET <https://publication-bdds.apps.epo.org/bdds/bdds-bff-service/prod/api/products/{productID}>

Sample response for product id **19**

<https://publication-bdds.apps.epo.org/bdds/bdds-bff-service/prod/api/products/19>

code
<pre>{ "id": 19, "name": "DTD/Schema repository", "description": "Most recent DTDs and schemas for EPO bulk data sets. For more information contact patentdata@epo.org.", "deliveries": [{ "deliveryId": 106, "deliveryName": "DOCDB in XML - DTDs - exchange-documents-v2.5.8", "deliveryPublicationDatetime": "2022-06-07T19:11:00.000+02:00", // // UTC format "deliveryExpiryDatetime": null, "files": [{ "fileId": 199, "fileName": "exchange-documents-v2.5.8.xsd", "fileSize": "544.6 kB", "fileChecksum": "0387391B23B683309C58EDDA791F4483D207ABDC", "filePublicationDatetime": "2022-06-07T19:11:00.000+02:00" // UTC format }] }] }</pre>

4. To Download the item for corresponding delivery and product

GET <https://publication-bdds.apps.epo.org/bdds/bdds-bff-service/prod/api/products/{productID}/delivery/{deliveryID}/file/{fileID}/download>

<https://publication-bdds.apps.epo.org/bdds/bdds-bff-service/prod/api/products/19/delivery/106/file/199/download>

It will generate the Response and will be sent as streaming and it needs to be save on the client end.

Note:

If you pass the invalid Bearer token or expired token, all the API's will return the HTTP response code as **401** and in this scenario, you need to generate the new token programmatically and pass it in the API.

If you call the API's without these token, API's will return the Response only for the public product.

If you try to get the product details for the subscribed products without the proper Bearer token, API will return the 401 response code with message as "Access Denied"