

# How to use Product Digital Asset API – V1.0

## Contents

1. Preamble.....	2
2. Overview.....	2
3. Getting Started .....	2
4. API Endpoint (Base URL) .....	3
5. Apikey Access .....	3
6. OAuth2 Access.....	3
7. GET View Operation .....	4
8. GET Viewer-URL Operation .....	5
9. Cached Responses .....	5
10. Status Codes .....	6

## 1. Preamble

This document describes how to use Schneider Electric Product Digital Asset v1.0 API which enables callers to request a 360° or 3D visualization of a product, identified by its product code.

## 2. Overview

Product Digital Asset is a RESTful API service by Schneider Electric that provides operations to request a 360° or 3D visualization of a product, identified by its CR (Commercial Reference).

- A 360° visualisation is using a set of images of the product. It is shown as an interactive image that can be rotated.
- A 3D visualisation is using a 3D model of the product. It is shown as an interactive object that can be manipulated and rotated by the user in any direction.

There are two possible operations to integrate those visuals in your product pages:

- One operation redirects to a media server which returns HTML that displays the visualization, suitable for managing an hyperlink in a web page.
- The other operation returns a link to the viewer image location, allowing the calling system to use this information independently.

In a future release, we plan also to provide you the way to embed directly on your product page a widget showing the 360° or 3D. Let us know if you are interested by this d type of integration.

## 3. Getting Started

### What you will need

The service is oriented to our partners. No special requirements, apart from requesting credentials.

Two options are supported to access this API.

- OAuth2 access, where you submit your credentials to retrieve a short-lived token, you then provide the access token in the API call via the Authorization Bearer header.
- Apikey access, where you provide the long-lived apikey as a query parameter

OAuth2 is supported by all the APIs in our catalog and is the preferred solution. Please use the “Contact Us” link on <https://developer.se.com> to request an new apikey or OAuth2 credentials. Your request will be referred to the API owner for validation.

## 4. API Endpoint (Base URL)

### Token URL

Environment	Token URL
Production	<a href="https://api.se.com/token">https://api.se.com/token</a>

### API URL

Environment	API URL
Production	<a href="https://api.se.com/v1/reference-data/product/digital-asset">https://api.se.com/v1/reference-data/product/digital-asset</a>

## 5. Apikey Access

Once you have an apikey, you can exercise both GET operations directly in a web browser.

Here are examples of both Get operations. Replace “your-apikey” with the actual value of your apikey.

<https://api.se.com/v1/reference-data/product/digital-asset/360/view?cr=BMXP342020&apikey=your-apikey>

<https://api.se.com/v1/reference-data/product/digital-asset/360/viewer-url?cr=BMXP342020&apikey=your-apikey>

## 6. OAuth2 Access

Once you have a Client ID and Client Secret, follow these steps to retrieve an access token. The sequence needs to be repeated before the token expires in 1 hour.

### Token Generation Sample using Postman:

1. Select POST and enter the Token URL.
2. From the Body tab, click on radio button x-www-form-urlencoded.
3. Manually enter Key and Value, “grant\_type” and “client\_credentials”.
4. From the Authorization tab, select Type “Basic Auth”.
5. Set Username to the **client key** (provided to you by the APIFIRST team offline).
6. Set Password to the **client secret** (provided to you by the APIFIRST team offline).
7. Send request and retrieve value of **access\_token** from the response message.

### Sample curl for Token

```
curl -X POST \
  https://api.se.com/token \
  -H 'Accept: */*' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -H 'Authorization: Basic
OFNkwlFqTVVGBwThNG1tVVhvTGUxajcxY2VHQUxBczY6YzQ0Mj10a1VMS3FRQkJDMA=='
\
  -d grant_type=client_credentials
```

## 7. GET View Operation

### Request

The request should match the following structure:

**URL:** <https://api.se.com/v1/reference-data/product/digital-asset/{asset-type}/view>

Where {asset-type} is one of 360, 3D, 360-or-3D, 3D-or-360

**HTTP Method:** GET

**HTTP Query Param:** cr=<commercial-reference>

**HTTP Query Param:** Optionally, apikey=<your-apikey>

**HTTP Header:** Optionally, Authorization Bearer <access-token>

### Successful Response

A successful response returns.

**HTTP Status Code:** 307

**HTTP Header:** Location=<visual-url-on-media-server>

Automatically redirects to the media server which returns an HTML page like this one:

```
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-
scale=1.0, maximum-scale=5.0">
  <title>Schneider Electric 360&deg; View</title>
  <script src="/scripts/360-widget/se-360-widget.js"></script>
</head>

<body style="width: 100vw; height: 100vh; margin: 0">
  <se-360-widget uri="/BMXP342020_360_NAM/20200128T1013/360"></se-360-widget>
</body>

</html>
```

## 8. GET Viewer-URL Operation

### Request

The request structure is the same as the other operation:

**URL:** <https://api.se.com/v1/reference-data/product/digital-asset/{asset-type}/viewer-url>

Where {asset-type} is one of 360, 3D, 360-or-3D, 3D-or-360

**HTTP Method:** GET

**HTTP Query Param:** cr=<commercial-reference>

**HTTP Query Param:** Optionally, apikey=<your-apikey>

**HTTP Header:** Optionally, Authorization Bearer <access-token>

### Successful Response

A successful response returns.

**HTTP Status Code:** 200

**HTTP Header:** Content-Type=application/json

**HTTP Body:**

Example:

```
{
  "rel": "related",
  "href": "https://visuals.se.com/BMXP342020_360_NAM/20200128T1013/360",
  "title": "Schneider Electric 360° view"
}
```

## 9. Cached Responses

The first query for a new product may take some seconds to return a result, but subsequent queries for the same product will be returned within milliseconds, due to the implementation of a response cache.

## 10. Status Codes

The API generates the following response status codes:

STATUS CODE	DESCRIPTION
307	OK for GET view operation. The request has succeeded. Automatically redirect to the url found in the Location header, which will return a HTML response.
200	OK for GET viewer-url operation. The request has succeeded. Please proceed with parsing the JSON response.
400	<p>Bad Request.</p> <p>The API request is invalid. One of the following error messages will be included in the error response body to help diagnose the problem. Variables within curly braces will be replaced with actual values.</p> <p>“Please specify product via query parameter cr (commercial reference) or ean”            “No commercial reference found for EAN {get.request.ean}”            “Invalid asset type {get.request.asset-type}”</p> <p>Please check the API specification for correct usage.</p>
401	<p>Unauthorized.</p> <p>You may not have provided an apikey, or if you are using the OAuth2 method, the access token may have expired; please request a new token via the /token endpoint and try once again.</p>
403	<p>Forbidden.</p> <p>You may have used a verb other than GET or the resource endpoint was incorrect. This error message will be included in the error response body.</p> <p>“Http verb {request.verb} is not supported for path suffix {proxy.pathsuffix}”</p>
404	<p>Not found.</p> <p>No image was found for the requested product. One of the following error messages will be included in the error response body.</p> <p>“No visual found for commercial reference {get.request.cr}”            “Invalid image link found {get.response.header.location}”            “Product not found for {get.request.cr}, server error: {developerMessage}”</p>
429	<p>Too many requests</p> <p>If you hit a spike arrest or quota limit, you will see this status code. There is a generous limits setup which can be customized for you. When requesting access, please share information about your expected call rate per day.</p>
500	<p>Service Unavailable.</p> <p>We are experiencing a temporary technical problem contacting the Product service in the backend, either due to schedule maintenance or a connectivity problem. Please try again later.</p>