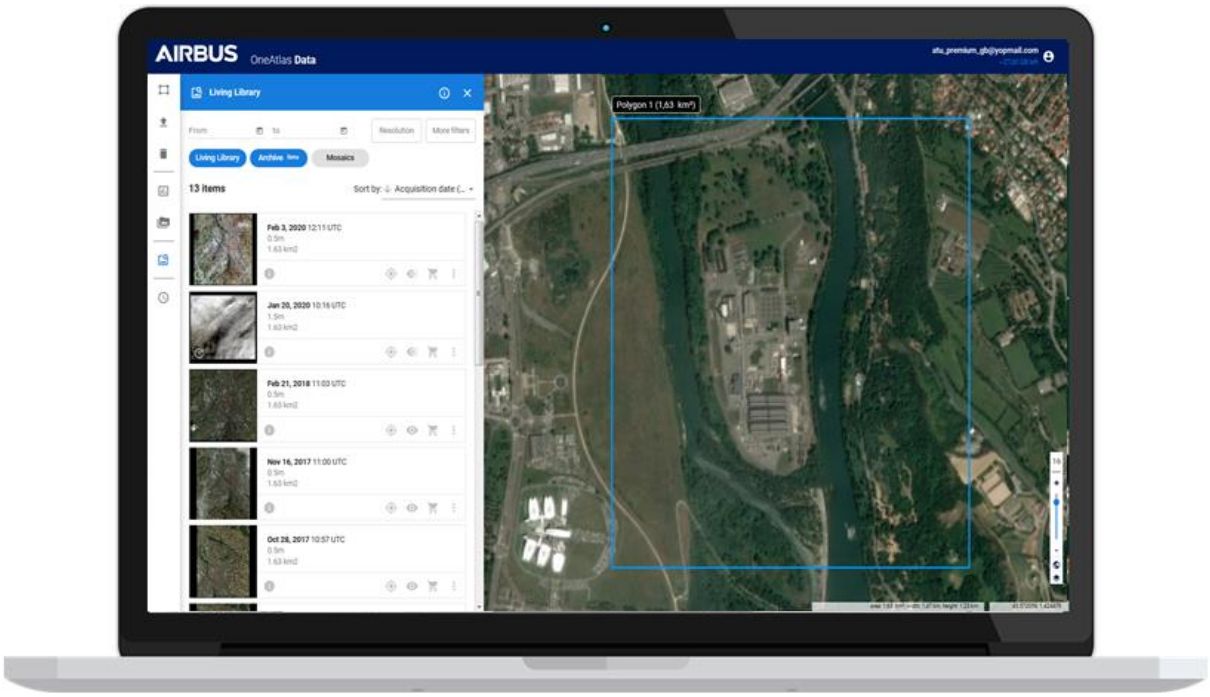


OneAtlas Basemap Streaming

Streaming API - Technical guide

This guide will provide you technical guidelines to migrate to our new OneAtlas Basemap streaming service. OneAtlas Basemap is a worldwide off-the-shelf imagery layer made from Airbus highest-grade satellite imagery. Sharp and constantly refreshed, it provides the most reliable context to you.

- ***Worldwide off-the-shelf layer:*** *Complete and consistent coverage of all earth landmasses*
- ***Highest-grade only:*** *Curated satellite imagery, with marginal cloud cover*
- ***Sharp:*** *1.5m over the world, 50cm over the most populated urban areas. 30cm over selected aeras. Precisely geolocated*
- ***Fresh:*** *New images are acquired every day, refreshed every year. A basemap is generated regularly from latest acquisitions*

1 How Do I Get Started?

To get started with OneAtlas Basemap, you will require an account. You should read our Authentication Section to discover how to create an account and authenticate using renewable tokens or your API KEY.
Once your account is created, you need to request access if you haven't been registered, so please contact Customer Care Support. We will be happy to provide you the access to the OneAtlas Basemap service.
Once your account is created and access provided, you should be ready to stream the basemap!
For additional details on the visualization capability and how to integrate our basemap directly in your GIS tools or your custom code, take a look at our View Service Guide.

2 Authenticate

If you do not have a OneAtlas account, please register here. You will receive login credentials for your account.
For authorization to use the OneAtlas services, an access token is required. Every OneAtlas account has an API key; this API Key is used to generate an access token. To get your API key, please visit the API Key Generator page.

> **Note:** *Please ensure you protect your API key. If anyone else gains access to it, they will be able to make requests and use your account.*

2.1 Get an Access Token from Your API Key

An API Key is your digital signature identifying you as a user of OneAtlas services. Using this key, you will need to get an access token that enables authorization. **Authorization** refers to the process of determining what permissions an authenticated client has for a set of resources.

For security reasons this access token expires regularly, then it's necessary to renew the authentication process to get a new one.

The endpoint to use to generate access tokens is described in the following table:

API Endpoint	https://authenticate.foundation.api.oneatlas.airbus.com/auth/realms/IDP/protocol/openid-connect/token
REST verb	POST
Authentication	API Key
API Reference	Authentication API

The required parameters are listed in the table below:

Parameters	Required	Description
apikey	yes	The OneAtlas API key associated with the service account to authenticate.

Parameters	Required	Description
client_id	yes	The API service group accessed. This service group can be retrieved by consulting the service documentation or calling the unprotected /well_known/serviceGroup URI on the service itself.
grant_type	yes	OneAtlas Grant type.Must be the value api_key for API key authentication.

> **Note:** *The value of the parameter "client_id" depends on the services you want to access. If you want to use the OneAtlas services, then use the "IDP" value to generate an access token. However, if you want to use the API key management services (at https://authenticate.foundation.api.oneatlas.airbus.com), then use the "AAA" value to generate an access token.*

Below is an example to retrieve an access token with the API Key to use with OneAtlas services:

```
curl -X POST
https://authenticate.foundation.api.oneatlas.airbus.com/auth/realms/IDP/protocol/openid-
connect/token \
   -H 'Content-Type: application/x-www-form-urlencoded' \
   -d 'apikey=<api_key>&grant_type=api_key&client_id=IDP'
```

If the authentication information is valid, then the return JSON structured provides an access token and its validity duration.

```
{
      "access_token": "<access_token>",
      "expires_in": 3600,
      "token_type": "bearer"
}
```

However, if authentication information is invalid or omitted, an error message will be returned with status code 403:

```
{
   "error": "access_denied",
   "error_description": "Access denied"
}
```

> **Important:** *For security reason, providing an incorrect API key will automatically suspend the authorization to access the API for a limited period of time. During this suspension period, the user will receive a 403 error, even if the API key is valid.*

For more ease, let's define it as an environment variable for your own user or globally if necessary.

```
export MY_TOKEN=<api_key>
```

2.2 Manage the API Keys Associated to a User

A user can generate up to 10 API keys. This could be convenient if you need to access to the One Atlas services in different context, for example from different tools or validity periods.

> **Important:** *The access to these endpoints requires authentication with an access token. Please note that this token must be generated using an existing API key and the procedure described in the previous paragraph Key, but with the "client_id" parameter set to the value "AAA".*

3 OneAtlas Basemap Streaming

3.1 Streaming Service

The OneAtlas Basemap Streaming service allow users to easily access the basemap without having to download anything. This guide will walk you through using WMTS to stream.

> ***Prerequisites:*** *If you haven't already, see our Authentication Section for information on how to acquire an API key and eventually generate tokens to access to our endpoints.*

3.2 WMTS

The Web Map Tile Service (WMTS) service conforms to the WMTS Simple profile norm. According the reference, this service provides a pyramid of tiles in WebMercator (EPSG:3857).

> ***Note****: Access to the basemap can be worldwide or geofenced. If the access is geofenced, white tiles are returned over areas out of the allowed area.*

3.3 Access Basemap from QGIS

The Basemap streaming services follows OGC norm for its WMTS. Consequently, the integration of our URLs in GIS tools is straightforward. Below is the example of QGIS.

You can easily access the OneAtlas Basemap imagery with OSGeo's QGIS Desktop application. QGIS is an Open Source Geographic Information System (GIS) app licensed under the GNU General Public License. QGIS can act as a WMTS client. The procedure to view WMTS imagery is documented directly on the QGIS website.

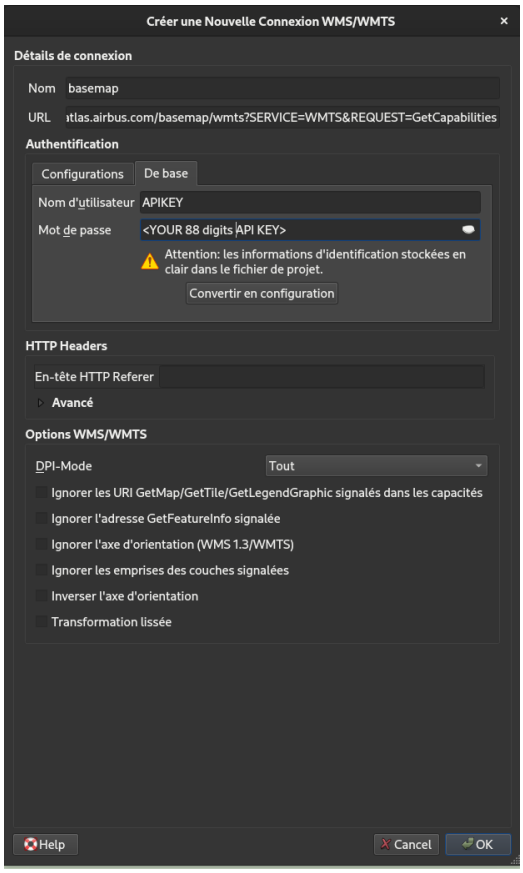
In the menu _"Layer" / "Add Layer" / "Add WMS/WMTS Layer…" / "New", you will need to provide your WMTS URL.

WMTS URL in QGIS	https://view-bm.api.oneatlas.airbus.com/basemap/wmts?SERVICE=WMTS&REQUEST=GetCapabilities

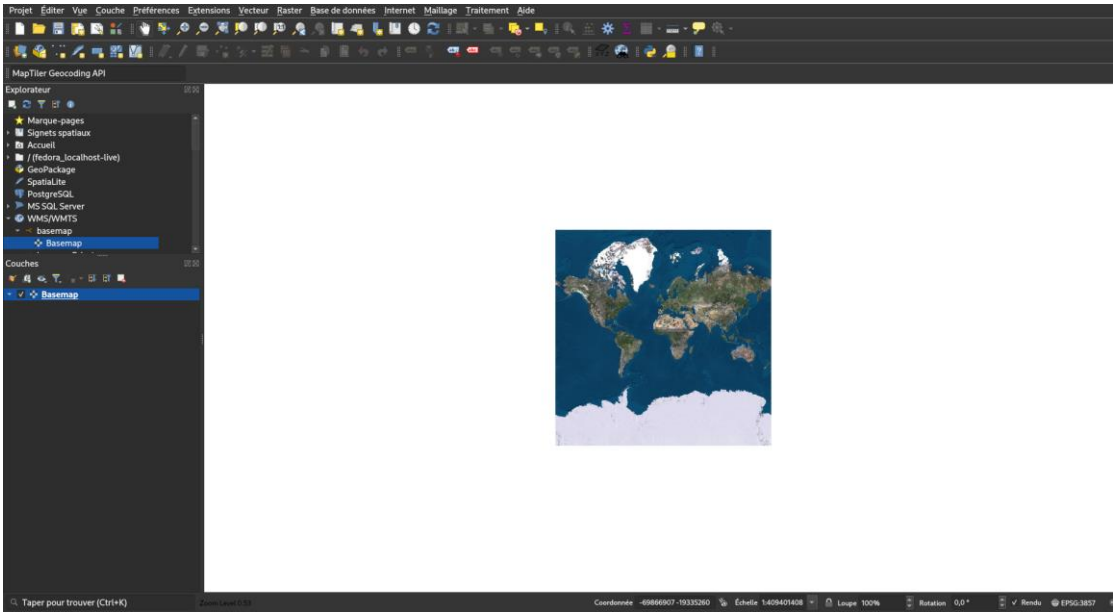
To ensure a secure experience, a username and password are **mandatory** to access to OneAtlas Basemap streaming service. The credentials to use are listed in the table below:

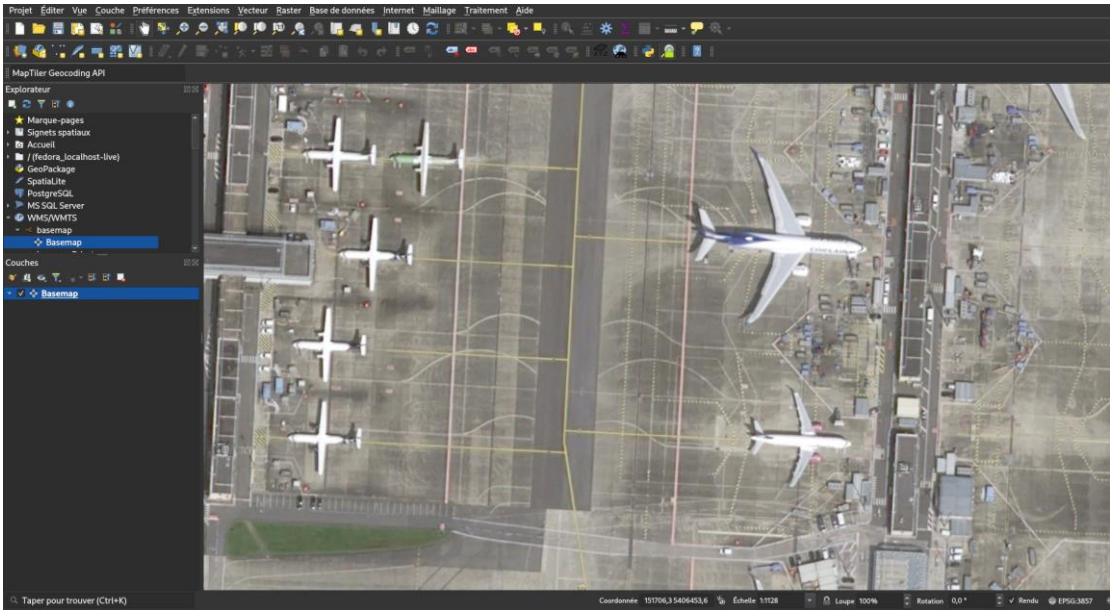
Username	APIKEY
Password	insert_your_88_digits_api_key

The resulting dialog box filled out should look like:



And the result in QGIS is:





3.4 Access Basemap Using Your Own Code

This section details how to use the WMTS protocol from your own code.

3.4.1 GetCapabilities

To retrieve the basemap WMTS information and the tiles detailed schema used, you should use the GetCapabilities service. The characteristics of the endpoint to access to the GetCapabilities are:

API Endpoint	https://view-bm.api.oneatlas.airbus.com/basemap/wmts?SERVICE=WMTS&REQUEST=GetCapabilities
REST verb	GET
Authentication	Bearer <access_token>

> ***Prerequisites:*** *If you haven't already, see our Authentication Section for information on how to acquire an API key and generate access tokens to access to our endpoints.*

Here is an example of a GetCapabilities request:

```
curl --location --request GET 'https://view-bm.api.oneatlas.airbus.com/basemap/wmts?SERVICE=WMTS&REQUEST=GetCapabilities' --header 'Authorization: Bearer <ACCESS TOKEN≥'
```

This will return an XML file with tiles. Below you can see an extract the XML file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Capabilities xmlns="http://www.opengis.net/wmts/1.0" xmlns:ows="http://www.opengis.net/ows/1.1" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:gml="http://www.opengis.net/gml"
xsi:schemaLocation="http://www.opengis.net/wmts/1.0 http://schemas.opengis.net/wmts/1.0/wmtsGetCapabilities_response.xsd" version="1.0.0">
  <!-- Service Identification -->
  <ows:ServiceIdentification>
    <ows:Title>One Atlas Basemap Streamer</ows:Title>
    <ows:Abstract>One Atlas - The world's freshest basemap at your fingertips</ows:Abstract>
    <ows:ServiceType>OGC WMTS</ows:ServiceType>
    <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
  </ows:ServiceIdentification>
  <ows:ServiceProvider>
    …
  </ows:ServiceProvider>
  <!-- Operations Metadata -->
  <ows:OperationsMetadata>
    <ows:Operation name="GetCapabilities">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="https://view-bm.api.oneatlas.airbus.com/basemap/wmts?">
            <ows:Constraint name="GetEncoding">
              <ows:AllowedValues>
                <ows:Value>KVP</ows:Value>
              </ows:AllowedValues>
            </ows:Constraint>
          </ows:Get>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
    <ows:Operation name="GetTile">
      <ows:DCP>
        <ows:HTTP>
          <ows:Get xlink:href="https://view-bm.api.oneatlas.airbus.com/basemap/wmts?">
            <ows:Constraint name="GetEncoding">
              <ows:AllowedValues>
                <ows:Value>KVP</ows:Value>
              </ows:AllowedValues>
            </ows:Constraint>
          </ows:Get>
        </ows:HTTP>
      </ows:DCP>
    </ows:Operation>
  </ows:OperationsMetadata>
  <Contents>
    <Layer xmlns:ows="http://www.opengis.net/ows/1.1">
      <ows:Title>Basemap</ows:Title>
      <ows:Abstract>Top quality, never ageing basemap.</ows:Abstract>
      <ows:Identifier>advancedBasemap</ows:Identifier>
      <ows:WGS84BoundingBox>
        <ows:LowerCorner>-180 -90</ows:LowerCorner>
        <ows:UpperCorner>180 90</ows:UpperCorner>
      </ows:WGS84BoundingBox>
      <Style isDefault="true">
        <ows:Title>Default Style</ows:Title>
        <ows:Identifier>default</ows:Identifier>
      </Style>
      <Format>image/unknown</Format>
      <TileMatrixSetLink>
        <TileMatrixSet>3857</TileMatrixSet>
      </TileMatrixSetLink>
    </Layer>
    <!--TileMatrixSet-->
    <TileMatrixSet xmlns:ows="http://www.opengis.net/ows/1.1">
      <ows:Identifier>3857</ows:Identifier>
      <ows:SupportedCRS>urn:ogc:def:crs:EPSG:6.3:3857</ows:SupportedCRS>
      <WellKnownScaleSet>urn:ogc:def:wkss:OGC:1.0:GoogleMapsCompatible</WellKnownScaleSet>
      <TileMatrix>
        <ows:Identifier>0</ows:Identifier>
        <ScaleDenominator>559082264.02871788</ScaleDenominator>
        <TopLeftCorner>-20037508.3427892 20037508.3427892</TopLeftCorner>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1</MatrixWidth>
        <MatrixHeight>1</MatrixHeight>
      </TileMatrix>
      ...
      <TileMatrix>
        <ows:Identifier>19</ows:Identifier>
        <ScaleDenominator>1066.3647919248922</ScaleDenominator>
        <TopLeftCorner>-20037508.3427892 20037508.3427892</TopLeftCorner>
```

```
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>524288</MatrixWidth>
        <MatrixHeight>524288</MatrixHeight>
      </TileMatrix>
    </TileMatrixSet>
  </Contents>
</Capabilities>
```

3.4.2 Get Tiles for a Web Mercator Tile Matrix Set

To get a single tile you should call the GetTile API as defined in the WMTS standard.
Example:

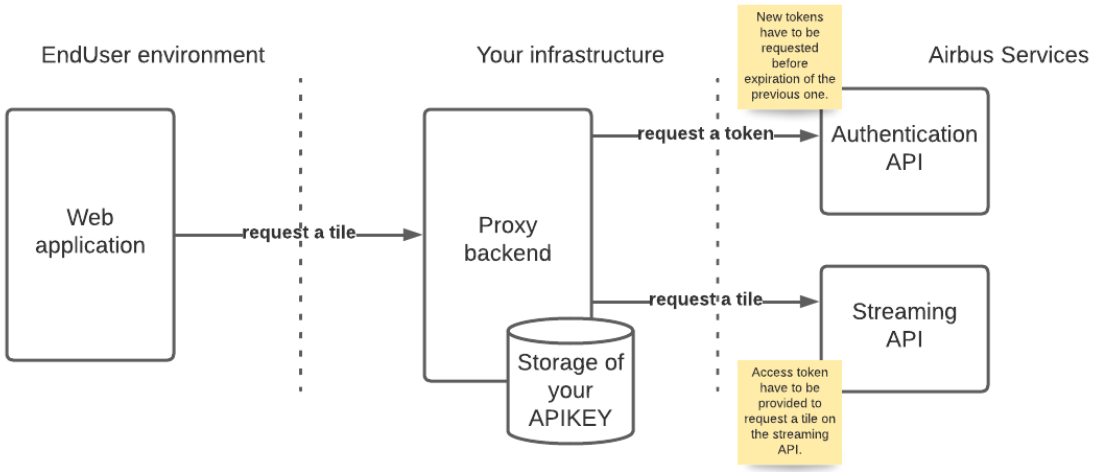
```
curl --location --request GET 'https://view-bm.api.oneatlas.airbus.com/basemap/wmts?style=default&tilematrixset=3857&Service=WMTS&Request=GetTile&Version=1.0.0&Format=image/unknown&layer=advancedBasemap&TileMatrix=19&TileCol=264129&TileRow=191412' --header 'Authorization: Bearer <Access Token>'
```

Here you can see the result:

4 Recommendations and advices

4.1 Webapplication integration

If you plan to distribute the basemap content thru a webapplication, it's important to hide the mechanism of APIKEY and access token from this webapplication and rely to a proxy backend for that. This backend has to be deployed in your infrastructure and be responsible for the storage of your APIKEY.



4.2 Access token lifecycles

Access token can be created by requesting the Authentication API with your APIKEY. Those tokens have a lifetime fixed by Airbus Authentication Service. Actually, the lifetime of a OneAtlas Access token is one hour.
It is recommended:

- To manage a cache of access token in your application
- To avoid inappropriate volume of Authentication API requests
- To anticipate the request of a new access token before the expiration limit.

4.3 APIKEY lifecycles

APIKEY are defined with an expiration date at creation. It is important to manage this constraint and to anticipate rotations. A good practice is to define regular rotations of APIKEY.
If your APIKEY is expired, you won't be able to generate new Access Tokens.