



READAR
real estate radar

API documentation



VERSION 2020



API reference

The Readar API is organised around REST. This architecture allows developers to access data from the Readar servers directly via HTTP protocol. Our API has predictable resource-oriented URLs and returns data in various encoded responses; such as JSON, XML, PNG and tiff.

Base URL

<https://api.readar.com>

By default, the Readar API Documentation demonstrate using cURL to interact with the API over HTTP, which can be done from the Linux terminal or a shell script. However, in the section Code Snippets you can find some Python examples.

Authentication

The Readar API uses Bearer tokens to authenticate requests. The bearer token is a cryptic string, generated by the server in response to a login request. Readar will provide you a user/password combination which enables you to retrieve this token.

Here's an example of how the curl request should look with your user/password entered:

```
curl --location --request POST 'https://api.readar.com//users/login' \  
  --header 'Content-Type: application/json' \  
  --data-raw '{  
    "email": "user@domain.com",  
    "password": "password"  
  }'
```

Here is what the response would look like:

```
{"token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlbnR5cCI6IkpXVCJ9eyJlbnR5cCI6IkpXVCJ9eyJlbnR5cCI6IkpXVCJ9"}
```

To authenticate a request with this token, you will need to include this token in a authorisation header.

With cURL, use:

```
--header "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlbnR5cCI6IkpXVCJ9eyJlbnR5cCI6IkpXVCJ9eyJlbnR5cCI6IkpXVCJ9"
```

It's not necessary to request the token every time you use it, to increase performance you can reuse the token a couple of days. Only until the servers returns a **401** status code, you will have to request a new token.



Endpoints and data types

We currently provide three types of data through the Readar API; heightdata, imagery and propertydata, each have their own endpoint.

Heightdata

Heightdata created by stereo matching or via other methods, can be accessed at the endpoint:

<https://api.readar.com/heightdata/{dsm}?>

Identifier

- dsm (string): Identifier of the digital surface model

Parameters

- boundingBox (string): Boundingbox of the desired area
 - or geoJson (string): Polygon of the desired area
- srid (integer): SRID of the boundingbox/polygon
- format (string): Format of the result; json or xml.

The request will return multiple arrays with height values. Values that are missing in the data or have been clipped in case you sent a polygon, are returned as null-values.

Examples

Example to retrieve heightdata clipped by a bounding box:

```
curl --location --request GET 'https://api.readar.com/heightdata/dsm-2019-25cm-nederland?
  boundingBox=132300,410800,132400,411000
  &srid=28992
  &format=json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlbWFpbiCI6ImVyaWt'
```

Example to retrieve heightdata clipped by a polygon:

```
curl --location --request GET 'https://api.readar.com/heightdata/dsm-2019-25cm-nederland?
  geojson={%22type%22:%22Polygon%22,%22coordinates%22:[[132579.367,410883.826,0],[132579.072,
  410887.521,0],[132570.073
  &srid=28992
  &format=json' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlbWFpbiCI6ImVyaWt'
```




Imagery

Retrieving imagery, such as true-orthophoto's, is similar to retrieving heightdata. The only difference is that it's only possible to retrieve an area by boundingbox and you can set a imagetype. The endpoint for the imagery is: <https://api.readar.com/imagery/{orthophoto}?>

Identifier

- orthophoto (string): Identifier of the image data

Parameters

- boundingBox (string): Boundingbox of the desired area
- srid (integer): SRID of the boundingbox
- imageType (string): Desired image format (jpeg/png/tif)

The API returns the image in de desired image format. A request with imageType=jpeg returns a compressed image, other imageTypes return uncompressed images.

Example

The following request returns a jpeg image using a boundingbox:

```
curl --location --request GET 'https://api.readar.com/imagery/trueortho-2019-10cm-waalwijk?
  boundingBox=132100,410800,132200,410900
  &srid=28992
  &imageType=jpeg' \
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlbWFpbiCI6ImVyaWt'
```

Propertydata

Information about an address can be retrieved using the propertydata endpoint. The only parameters that is needed for this data is a postal code/house number combination and the desired output format.

The location of the endpoint is: <https://api.readar.com/propertydata?>

Parameters

- postcode (string): Postcal code, without a white space
- huisnummer (integer): House number
- format (string): Format of the result; json or xml



The API returns json or xml data containing the following attributes:

- `postcode` (string): Postcal code
- `huisnummer` (integer): House number
- `huisletter` (string): House letter
- `huisnummertoevoeging` (string): House number addition
- `woonplaatsnaam` (string): City
- `identificatie` (string): Identification conform the BAG
- `gebouwooppervlak` (float): Building surface area in m2
- `dakoppervlakPlat` (float): Flat roof area in m2
- `dakoppervlakSchuin` (float): Sloped roof area in m2
- `goothoogte` (float): Gutter height in m
- `nokhoogte` (float): Ridge height in m
- `gebouwinhoud` (float): Building volume in m3
- `helling` (float): Slope of the roof in degrees (main roof)
- `richting` (float): Direction of the roof in degrees (main roof)
- `bouwJaar` (integer): Year of construction
- `typeWoning` (string): Type of building
- `typeDak` (string): Type of roof

The structure of the shell script is:

```
curl --location --request GET + ' + API-address + /propertydata? + option1 + & option2 + & option3 + etc. + ' + \  
--header 'Authorization: Bearer + personal token + '
```

Example

The following request will return all available data for one particular address:

```
curl --location --request GET 'https://api.readar.com/propertydata?  
  postcode=9901BE  
  &huisnummer=32  
  &format=json' \  
--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlbWFpbiCI6ImVyaWt'
```



Code Snippets

Example how to request the Bearer token using python:

```
import requests
import json

def get_auth_token(login):
    url = 'https://api.readar.com/users/login'

    response = requests.post(url, json=login).content
    json_obj = json.loads(response)
    token_string=json_obj["token"].encode("ascii","ignore")

    return token_string

login = {"email": "user@domain.com", "password": "password"}
auth_token = get_auth_token(login)
```

Example of how to retrieve propertydata using python:

```
import requests
import json

def get_data(auth_token, postalcode, number, dataformat):
    url = 'https://api.readar.com/propertydata?postcode={}&huisnummer={}&format={}'

    request_url = url.format(postalcode, number, dataformat)
    header = {'Authorization': 'Bearer ' + auth_token}
    result = requests.get(request_url, headers=header)

    return result.content

auth_token = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlbnVyaWt'
result = get_data(auth_token, '9901BE', 32, 'json')
```