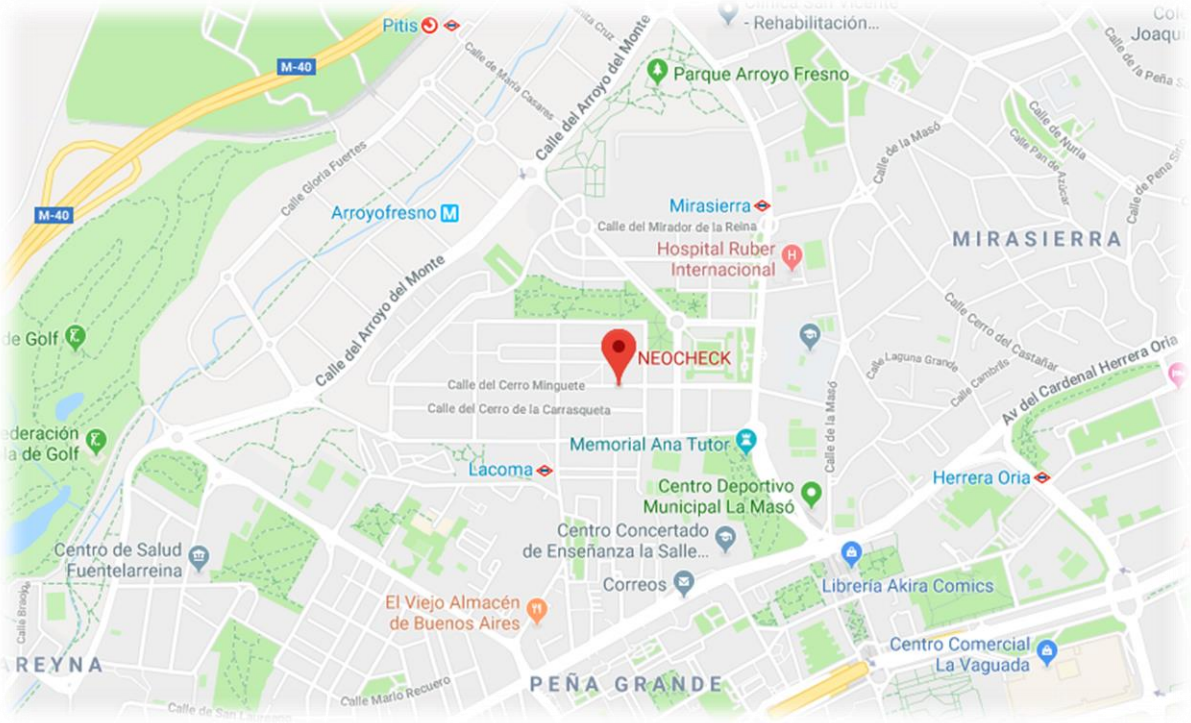

ADDRESS CHECK API

Developers Guide

Content description

This is a reference manual and configuration guide for the NeoCheck Address Check API product. It shows how to interact with the JSON Web API from an external client to know if a given address exists and if it can be correlated with any bill or document.

Madrid, February 27th 2019

Disclaimer

The information contained within this document is and shall remain the property of NeoCheck. It must not be produced in whole or in part, or given or communicated to any third party without the prior consent of NeoCheck.

Whilst careful attention has been paid to ensure the appropriate referencing of information and sources, it is possible that a document of this nature might contain some errors. In such cases, upon notification NeoCheck will immediately analyze and make any required corrections or amendments.

Release Notes

Version	Date	Description
1.0	28/01/2019	Initial Version
1.1	27/02/2019	Changes in API models

Index

1. NeoCheck API Endpoint

NeoCheck provides an API for address verification checks. This API allows users to check if a given address exists and correlate it against a given document, such as bills, contracts, etc.

Endpoint	Environment
`https://neocheck.net/api`	PRO

2. Authentication

The Address Check API access is protected by using basic authentication and OAuth 2.0 authorization tokens.

2.1. Obtain JWT Authentication Token

HTTPS POST method to authenticate. In order to request access to Address Check API, you must call the authentication method with your API `client_id` and `client_secret`, along with a valid `username` and `password`. Once validated, you will receive an authorization token, which must be used to call Address Check API methods.

Endpoint	HTTP Method
connect/token	POST

Request Parameters:

Name	Type	Description
content_type	string	`application/x-www-form-urlencoded` indicates that parameters are passing as key/value pairs in the body of the HTTPS message
grant_type	string	`password` indicates that it's an authentication request including basic authentication
client_id	string	your Client Id
client_secret	string	Your client secret
username	string	valid username already created in the system for the Client Id
password	string	valid password for username provided

Response:

If request is correct, you will receive a Success response with the access token generated. This token is only valid for a short period, exposed in the `expires_in` property. You will also receive a `refresh_token`, which allows you to refresh your access token without having to send your credentials again. The API call to refresh token is explained in the next point.

HTTP Code	Body	Description
200	JSON `Authorization` object	Valid OAuth token for request authorization
400	JSON `ErrorMessage` object	Invalid ClientId
400	JSON `ErrorMessage` object	Unsupported grant type
400	JSON `ErrorMessage` object	Invalid credentials
500	JSON `ErrorMessage` object	Internal Error

Example of Successful Authorization Response:

```
ResultCode: HTTP/1.1 200 OK
Content-Type: application/json
Body:
{
  "token_type": "Bearer",
  "access_token": "yourAccessToken",
  "expires_in": 3600,
  "id_token": "youIdToken",
  "refresh_token": "yourRefreshToken"
}
```

Example of Invalid Client Response:

```
ResultCode: HTTP/1.1 400 BadRequest
Content-Type: application/json
Body:
{
  "error": "Invalid ClientId",
}
```

2.2. Token Refresh

If you already have authenticated, you can use the `refresh_token` property provided to renew your `access_token` authorization token, which must be used to call Address Check API methods.

Endpoint	HTTP Method
connect/token	POST

Request Parameters:

Name	Type	Description
content_type	string	`application/x-www-form-urlencoded` indicates that parameters are passing as key/value pairs in the body of the HTTPS message
grant_type	string	`refresh_token` indicates that it's an authentication request including basic authentication
refresh_token	string	the refresh token
client_id	string	your Client Id
client_secret	string	Your client secret

Response:

If request is correct, you will receive a Success response (StatusCode 200), with a new access token, and the same refreh_token. If request is not correct, you will receive a Bad Request response (StatusCode 400), with error description

HTTP Code	Body	Description
200	JSON `Authorization` object	Valid OAuth token for request authorization
400	JSON `ErrorMessage` object	Invalid ClientId
400	JSON `ErrorMessage` object	Unsupported grant type
400	JSON `ErrorMessage` object	Invalid ticket (`refresh_token` not valid)
500	JSON `ErrorMessage` object	Internal Error

2.3. Abusive usage countermeasures

In order to prevent a bad or abusive usage and assure the best performance, there is a maximum number of requests per hour that a user can perform. This maximum number is set to 3600 calls per hour (an average of 1 call per second). After this limit is reached, the Address Check API will respond with a BadRequest (StatusCode 400) and an error message stating that the maximum number of calls per hour has been reached.

3. Address Check

3.1. Validity Check

HTTPS POST method to check the existence of an address using Google Maps API.

Endpoint	HTTP Method
/v1/Address/Check	POST

Request Parameters:

Name	Type	Description
authorization	query parameter	`Bearer access_token` authorization header with a valid access token
addressCheckRequest	Body: JSON object	JSON object, with properties: - **fullAddress**: entire address including all data. If this field is set, it will be used and the rest will be ignored - **streetType**: (optional) type of via. (st, av, etc.) - **streetName**: (optional) name of the street - **streetNumber**: (optional) house number - **postalCode**: (optional) postal number - **city**: (optional) city - **state**: (optional) State or region - **country**: (optional) country

Response:

If request is correct, you will receive a Success response along with an object containing the result of the operation, including a map image of the address surrounding area if it was found.

HTTP Code	Body	Description
200	AddressCheck result	**- foundInMap**: yes, no - **mapBase64**: (optional) picture in Base64String format of the surrounding area with a marker
400		Invalid content
401		Unauthorized. Invalid token
500	JSON `ErrorMessage` object	Internal Error

Example of Successful Validity Check Response:

```
ResultCode: HTTP/1.1 200 OK
Content-Type: application/json
Body:
{
  "foundInMap": true,
  "mapBase64": "IMAGE_BASE64"
}
```

3.2. Document Validity Check

HTTPS POST method to find the existence of an address in a given document.

Endpoint	HTTP Method
/v1/Address/DocumentCheck	POST

Request Parameters:

Name	Type	Description
authorization	query parameter	`Bearer access_token` authorization header with a valid access token
addressDocumentRequest	Body: JSON object	JSON object, with properties: - **fullAddress**: entire address including all data. If this field is set, it will be used and the rest will be ignored - **streetType**: (optional) type of via. (st, av, etc.) - **streetName**: (optional) name of the street - **streetNumber**: (optional) house number - **postalCode**: (optional) postal number - **city**: (optional) city - **state**: (optional) State or region - **country**: (optional) country - **documentImagesBase64**: collection of document page images in Base64String format where the address is supposed to be located

Response:

If request is correct, you will receive a Success along with an object containing the result of the operation, including a list of hits found in the document for the address provided, with

HTTP Code	Body	Description
200	addressDocument result	**- foundInDocument**: yes, no, not executed. - **hits**: (optional) List of possible locations, including page index, coordinates x and y, extracted text and accuracy respect to the request address
400		Invalid content
401		Unauthorized. Invalid token
500	JSON `ErrorMessage` object	Internal Error

Example of Successful Document Validity Check Response:

```
ResultCode: HTTP/1.1 200 OK
Content-Type: application/json
Body:
{
  "hits": [
    { "pageIndex": 0,
      "locationX": "25",
      "locationY": "94",
      "extractedText": "445 Mount Eden Road, Mount Eden, Auckland",
      "accuracy": 100},
    { "pageIndex": 1,
      "locationX": "78",
      "locationY": "225",
      "extractedText": "445 Mount Eden Road",
      "accuracy": 80}
  ],
  "foundInDocument": true
}
```