

API Overview

12/19/2023 11:38 am EST

Overview

These documents describe the Autoaddress 3.0 API, including authentication, HTTP resources and recommended best practices. This content is intended for individuals with web development experience and knowledge of HTTP.

The Autoaddress 3.0 API uses:

- HTTP GET
- RESTful endpoint structure
- Response payloads formatted as JSON

API Endpoint

All API calls use the following base URL:

```
https://api.autoaddress.com/3.0/
```

Authentication

Each call to the Autoaddress 3.0 REST API must be authenticated. This is done by passing either an API Key or a token to a request. A token can be created using the *createtoken* endpoint, and can be used in multiple subsequent API calls for 24 hours.

Security

The Autoaddress 3.0 REST API is accessible only using HTTPS secure protocol.

Responses

Returned as JSON only. We reserve the right to expand all JSON responses in the future by adding new properties.

Each response returns the appropriate HTTP Status Code based on the incoming request (See below for full list of HTTP Status Code returned).

Compliance with HTTP, URL and JSON Specifications

We reserve the right to deny HTTP requests which are not fully compliant to our specifications. Use properly formed URLs and query strings, which includes URL-encoded query string parameters. Interpret all HTTP response payloads according to their Content-Type.

For example, application/json payloads should always be parsed as JSON. Our API produces valid JSON according to ECMA-404 standard, including whitespace between tokens. We are not responsible for problems resulting from handling JSON data improperly.

HATEOAS Links

API response usually includes an array of [HATEOAS](#) links. This allows interaction with the API solely through the hyperlinks we provide you. You do not need to hardcode logic into your client in order to use our API.

```
"links": [ { "rel": "next", "href": "https://api.autoaddress.com/3.0/lookup?token=..." } ]
```

The array of links can have many links, the link of interest must be identified using the value of the “rel” property (“next” in above example). When the client identifies the link, it should just follow the hyperlink in the value of “href” property. That value can have some internal parameters, so the client should not parse it or modify it in any way.

Common values used for “rel” are:

- next – follow this URL to comply with the recommended API flow for real-time capture
- self – indicating the URL that created the current document

Response HTTP Status Codes

Code	Summary	Description
200	Success	Everything worked as expected, search result is returned.
400	Bad Request	Invalid or missing input parameter.
401	Unauthorized	The API key or token is invalid or missing from the request.
403	Forbidden	The API key or token is not allowed to use the service. Reasons for this include: <ul style="list-style-type: none">• The API key is disabled• The token is expired• The client has reached the limit for API calls on their plan
404	Not Found	Request URL does not exist.
429	Too Many Requests	The client has been rate limited, please wait a short while and retry the request again.
500	Error	An error has occurred with the service.