

Citizens API

Overview

The Citizens API uses the [REST](#) architectural style. The API is organized into resources and each resource should be reachable using an unique [URI](#). Different actions occurs depending on the HTTP verb you use to access the resource. In general, the http verbs works this way on our API:

HTTP Verb	Description
POST	When the target is a collection resource it will create a new item for the collection. It is also used for performing special actions.
GET	If the target is a collection it should return a subset of the collection that could be filtered using URL query parameters. If the target is a document resource identified by ID it will return the specific information for that resource.
PUT	This HTTP method will be used to update an already existing resource.
DELETE	This method lets you to delete a resource. It is only allowed for certain document resources.

All the API responses will return an HTTP [Status Code](#) and all data is sent and received as [JSON](#).

Security

For authorization you must provide a valid API key in the request by using the "**X-API-KEY**" header. The valid API Key is customized per installation and will be set on the server using an environment variable name: **CZ_API_KEY**. For instance if you set the variable like this:

```
export CZ_API_KEY="1QOna4hcXzKZYa3owkPUw2UzZqFSRfQnmhWcm2uE"
```

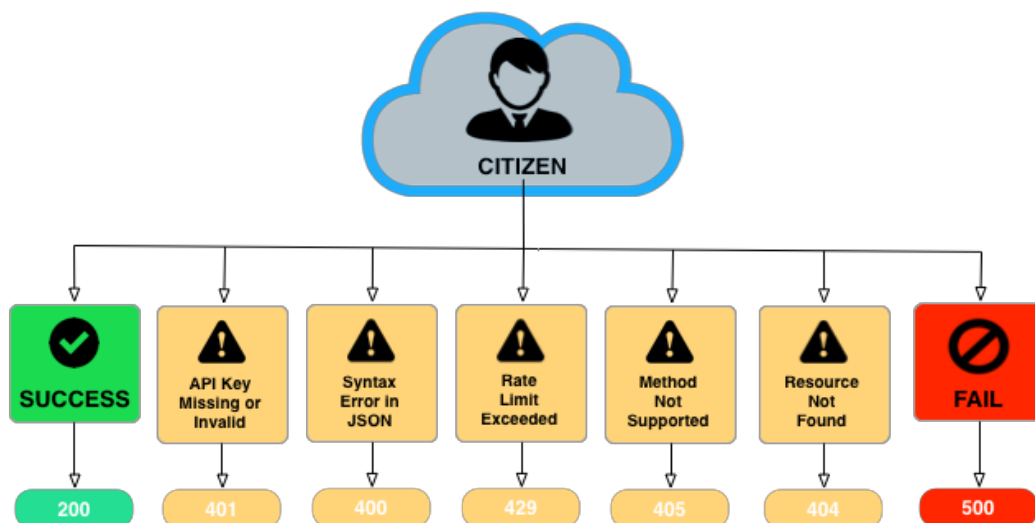
Clients must set the header value for every request to the API using that key:

X-API-KEY: 1QOna4hcXzKZYa3owkPUw2UzZqFSRfQnmhWcm2uE

Status Codes

We will use standard HTTP response codes to indicate failures or success on the API responses. Responses with 2xx represent a successful request and responses with 4xx codes means there was an error on the request. When you received a 5xx error that means something unexpected happen on the server.

Code	Description
200	OK - Successful request. Everything works as expected.
201	Created - This code is returned when the client did a POST to a collection resource and the new resource was successfully created.
400	Bad Request - Something failed. Most likely on the client side. Could be a non-expected parameter or a bad formatted value.
401	Unauthorized - Either the API key was not provided or was invalid or the basic HTTP authentication failed because the client didn't send the valid credentials.
404	Not Found - The requested resource could not be found. Probably the URI was wrong or the intended item id is not present in the request.
405	Method not allowed - A request was made using an HTTP verb the resource not support.
406	Not acceptable - We don't support the requested content type. We only support JSON. Try using <code>"*/*"</code> or <code>"application/json"</code> on your <code>Accept</code> header.
429	Too many requests - The client exceeded the rate limit of requests per second.



Resources

Citizen



Represents a Costa Rican citizen.

Field	Type	Description
identification	string	Identification number of the person.
firstName	string	The complete first name of the person.
lastName	string	Family name. The last name of the person's father.
surName	string	Family name. The last name of the person's mother.
gender	string	Person's gender: M = male, F = female.
address	Address	Registered voting district of the person.

Field	Type	Description
province	string	One of the seven provinces of Costa Rica
city	string	One of the 81 cantons of Costa Rica
district	string	One of the 473 districts of Costa Rica

READ

To retrieve the information of a person perform a **GET** request over the URI: `/citizens/{id}`

Example:

GET `/citizens/205480941`

X-API-KEY: *Secret Key*

Content-Type: *application/json*

Response:

Status: **200 OK**

Content-Type: *application/json*

Body:

```
{
  "Identification": "205480941",
  "FirstName": "Greivin",
  "LastName": "Lopez",
  "SurName": "Paniagua",
  "Gender": "M",
  "Address": {
    "Province": "Alajuela",
    "City": "Central",
    "District": "Villa Bonita (Parte Sur)"
  }
}
```