ABH Automation Testing Task (API Testing Task)

Naida Hujić

Email: nhujic@gmail.com

Atlantbh d.o.o. Sarajevo

Email: jobs@atlantbh.com

Sarajevo, mart 2018. godine

TEST CASES

Geocoding

1. (Smoke Test)

- Description: Represents the set of geocoded places whose address best matches a given query address.
- Endpoint:

  https://maps.googleapis.com/maps/api/geocode/json?address=***ADDRESS***
  &key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json or XML
- Params:
  - Address - The street address that you want to geocode, in the format used by the national postal service of the country concerned. (required)
  - Key - Your application's API key. This key identifies your application for purposes of quota management. (required)
- Status code: OK
- Response:

```
{
  "results": [   {
    "address_components":       [
            {...},
            {...},
            {...},
            {...},
            {...},
            {...}
    ],
    "formatted_address": ... ,
    "geometry":       {
      "bounds":          {...},
      "location":          {...},
      "location_type": ...,
      "viewport":          {...}
    },
    "place_id":  ...,
    "types": [...]
  }],
  "status": "OK"
}
```

2.

- Description: Represents the set of geocoded places whose address best matches a given query address in the defined bounding box
- Endpoint:

  https://maps.googleapis.com/maps/api/geocode/json?address=***ADDRESS***&bounds=***SOUTHWEST_CORNER*|*NORTHEAST_CORNER***&key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Address – (required)
  - Bounds – defines the latitude/longitude coordinates of the southwest and northeast corners of this bounding box using a pipe (|) character to separate the coordinates. (required)
  - Key – (required)
- Status code: OK
- Response:

```
{
  "results": [   {
    "address_components":      [
              {...},
              {...},
              {...},
              {...},
              {...},
              {...}
    ],
    "formatted_address": ... ,
    "geometry":      {
      "bounds":         {...},
      "location":         {...},
      "location_type": ...,
      "viewport":         {...}
    },
    "place_id":  ...,
    "types": [...]
  }],
  "status": "OK"
}
```

3.

- Description: Return results biased to a particular region
- Endpoint:

  https://maps.googleapis.com/maps/api/geocode/json?address=***ADDRESS***&region=***REGION***&key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Address – (required)
  - Region – This parameter takes a ccTLD (country code top-level domain) argument specifying the region bias. (required)
  - Key – (required)
- Status code: OK
- Response:

```
{
  "results": [  {
    "address_components":      [
            {...},
            {...},
            {...},
            {...}
    ],
    "formatted_address": ...,
    "geometry":      {
      "bounds":        {
        "northeast":          {...},
        "southwest":          {...}
      },
      "location":        {...},
      "location_type": ...,
      "viewport":        {
        "northeast":          {...},
        "southwest":          {...}
      }
    },
    "place_id": ...,
    "types":      [...]
  }],
  "status": "OK"
}
```

4.

- Description: Filtering address using the components filter (country)
- Endpoint:

  https://maps.googleapis.com/maps/api/geocode/json?address=***ADDRESS***&components=country:***VALUE***&key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Address – (required)
  - Components (*required* if the request doesn't include an address)
    - Country - matches a country name or a two letter ISO 3166-1 country code.
  - Key – (required)
- Status code: OK
- Response:

```
{
  "results": [   {
    "address_components":       [
            {...},
            {...},
            {...},
            {...}
    ],
    "formatted_address": ...,
    "geometry":       {
      "bounds":          {
        "northeast":            {...},
        "southwest":             {...}
      },
      "location":          {...},
      "location_type": ...,
      "viewport":          {
        "northeast":            {...},
        "southwest":             {...}
      }
    },
    "place_id": ...,
    "types":       [...]
  }],
  "status": "OK"
}
```

5.

- Description: Filtering address using the components filter (postal_code)
- Endpoint:

  https://maps.googleapis.com/maps/api/geocode/json?address=***ADDRESS***&components=country:***VALUE***|postal_code:***VALUE***&key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Address – (required)
  - Components (*required* if the request doesn't include an address)
    - Postal_code - indicates a postal code as used to address postal mail within the country
  - Key – (required)

- Status code: OK
- Response:

```
{
  "results": [   {
    "address_components":       [
            {...},
            {...},
            {...},
            {...}
    ],
    "formatted_address": ...,
    "geometry":       {
      "bounds":          {
        "northeast":            {...},
        "southwest":            {...}
      },
      "location":          {...},
      "location_type": ...,
      "viewport":          {
        "northeast":            {...},
        "southwest":            {...}
      }
    },
    "place_id": ...,
    "types":       [...]
  }],
  "status": "OK"
}
```

6.

- Description: Finding address without param address
- Endpoint:

  https://maps.googleapis.com/maps/api/geocode/json?components=locality:***VALUE***|administrative_area:***VALUE***|country:***VALUE***&key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params
  - Components – (required)
    - Locality- indicates an incorporated city or town political entity
    - Administrative_area
    - Country
  - Key – (required)
- Status code: OK
- Response

```
{
  "results": [   {
    "address_components":       [
            {...},
            {...},
            {...},
            {...}
    ],
    "formatted_address": ...,
    "geometry":       {
      "bounds":          {
        "northeast":            {...},
        "southwest":            {...}
      },
      "location":          {...},
      "location_type": ...,
      "viewport":          {
        "northeast":            {...},
        "southwest":            {...}
      }
    },
    "place_id": ...,
    "types":       [...]
  }],
  "status": "OK"
}
```

7.

- Description: Finding location that is not in country that we entered (example Sarajevo in Serbia)
- Endpoint

    https://maps.googleapis.com/maps/api/geocode/json?components=locality:***VALUE***
    |country:***VALUE***&key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
    - Components – (required)
        - Locality
        - Country
- Status code: ZERO_RESULTS
- Response:

```
{
  "results": [],
  "status": "ZERO_RESULTS"
}
```

8.

- Description: Finding address without enter API key
- Endpoint

    https://maps.googleapis.com/maps/api/geocode/json?address=***ADDRESS***

- Request: **GET**
- Formats supported: json
- Params:
    - Address – (required)
- Status code: OVER_QUERY_LIMIT
- Response

```
{
  "error_message": "You have exceeded your daily request quota for this API. We
recommend registering for a key at the Google Developers Console:
https://console.developers.google.com/apis/credentials?project=_",
  "results": [],
  "status": "OVER_QUERY_LIMIT"
}
```

9.

- Description: Do not enter any parameter
- Endpoint

  https://maps.googleapis.com/maps/api/geocode/json

- Request: **GET**
- Formats supported: json
- Params:
- Status code: INVALID_REQUEST
- Response

```
{
  "error_message": "Invalid request. Missing the 'address', 'bounds', 'components', 'latlng' or 'place_id' parameter.",
  "results": [],
  "status": "INVALID_REQUEST"
}
```

Reverse Geocoding

1.

- Description: Finding addresses by the latitude and longitude values which are specifying the location
- Endpoint

  https://maps.googleapis.com/maps/api/geocode/json?latlng=***VALUE***?key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Latlng - The latitude and longitude values specifying the location for which you wish to obtain the closest, human-readable address. (required)
- Status code: OK
- Response:

```
{
  "results":   [
        {
     "address_components":        [
                {...},
                {...},
                {...},
                {...}
        ],
        "formatted_address": ...,
        "geometry":          {...},
        "place_id": ...,
        "types": [...]
      },
          {
        "address_components":          [
                {...},
                {...}
        ],
        "formatted_address": ...,
        "geometry":          {...},
        "place_id": ...,
        "types":          [...]
      } , ...
   ],
   "status": "OK"
}
```

2.

- Description: Filter address by result type
- Endpoint

  https://maps.googleapis.com/maps/api/geocode/json?
  latlng=***VALUE***?result_type=***VALUE***?key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Latlng - The latitude and longitude values specifying the location for which you wish to obtain the closest, human-readable address. (required)
  - Result_type – the API fetches all results for the specified latlng, then discards those results that do not match the specified address type
- Status code: OK
- Response:

```
{
  "results": [   {
    "address_components":       [
              {...},
              {...},
              {...},
              {...}
    ],
    "formatted_address": ...,
    "geometry":     {
      "bounds":        {
        "northeast":          {...},
        "southwest":          {...}
      },
      "location":        {...},
      "location_type": ...,
      "viewport":        {
        "northeast":          {...},
        "southwest":          {...}
      }
    },
    "place_id": ...,
    "types":     [...]
  }],
  "status": "OK"
}
```

3.

- Description: Filter address by location type
- Endpoint

  https://maps.googleapis.com/maps/api/geocode/json?latlng=***VALUE***?location_type=***VALUE***?key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Latlng - The latitude and longitude values specifying the location for which you wish to obtain the closest, human-readable address. (required)
  - Location_type – the API fetches all results for the specified latlng, then discards those results that do not match the specified location type
- Status code: OK
- Response:

```
{
  "results": [  {
    "address_components":      [
            {...},
            {...},
            {...},
            {...}
    ],
    "formatted_address": ...,
    "geometry":       {
      "bounds":          {
        "northeast":             {...},
        "southwest":             {...}
      },
      "location":          {...},
      "location_type": ...,
      "viewport":          {
        "northeast":             {...},
        "southwest":             {...}
      }
    },
    "place_id": ...,
    "types":       [...]
  }],
  "status": "OK"
}
```

4.

- Description: Filter address by location type
- Endpoint

  https://maps.googleapis.com/maps/api/geocode/json?latlng=***VALUE***?location_type=***VALUE***?key=***YOUR_API_KEY***

- Request: **GET**
- Formats supported: json
- Params:
  - Latlng - The latitude and longitude values specifying the location for which you wish to obtain the closest, human-readable address. (required)
  - Location_type – the API fetches all results for the specified latlng, then discards those results that do not match the specified location type ("ROOFTOP", "RANGE_INTERPOLATED", "GEOMETRIC_CENTER", "APPROXIMATE")
- Status code: OK
- Response:

```
{
  "results": [   {
    "address_components":       [
            {...},
            {...},
            {...},
            {...}
        ],
        "formatted_address": ...,
        "geometry":      {
          "bounds":        {
            "northeast":          {...},
            "southwest":          {...}
          },
          "location":        {...},
          "location_type": ...,
          "viewport":        {
            "northeast":          {...},
            "southwest":          {...}
          }
        },
        "place_id": ...,
        "types":      [...]
      }],
      "status": "OK"
    }
```

5.

- Description: Do not enter any parameters
- Endpoint

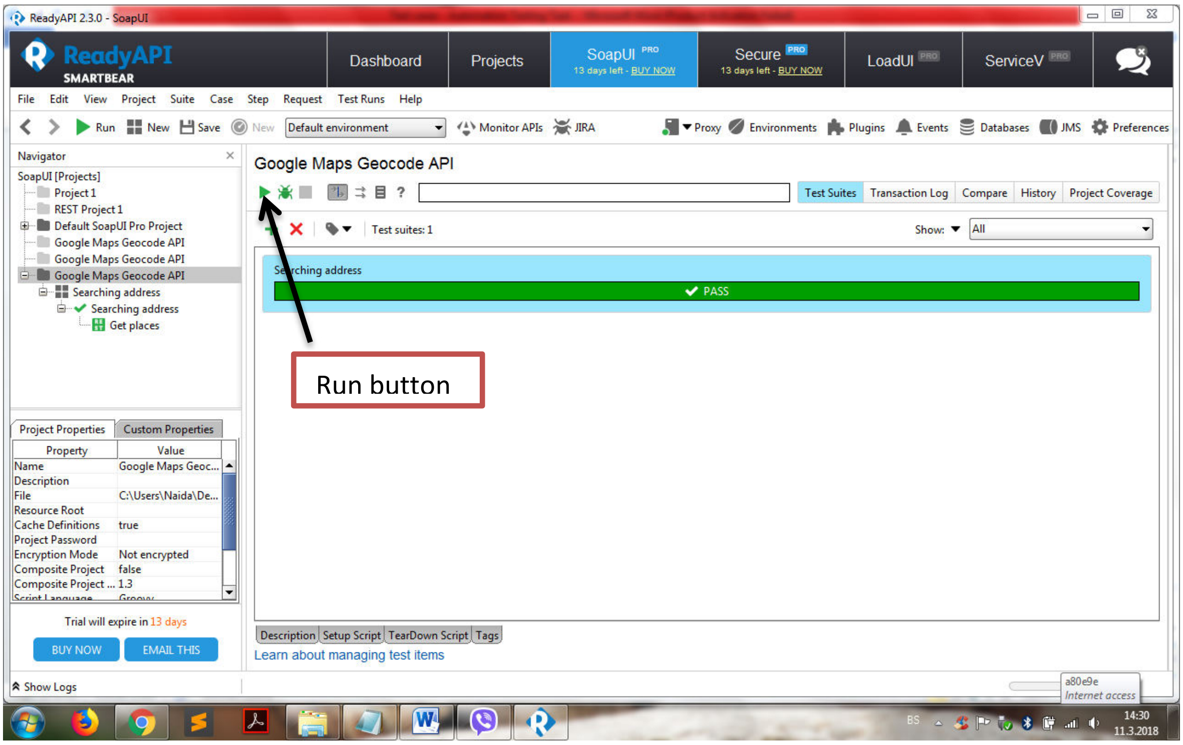
    https://maps.googleapis.com/maps/api/geocode/json

- Request: **GET**
- Formats supported: json
- Params
- Status code: INVALID_REQUEST
- Response:

```
{
  "error_message": "Invalid request. Missing the 'address', 'bounds', 'components',
'latlng' or 'place_id' parameter.",
  "results": [],
  "status": "INVALID_REQUEST"
}
```

Documentation for automated test in SoapUI Pro

With a tool SoapUI Pro I wrote automated test for Smoke Test – Searching address. In the Github repository You can find a XML – file of the project which You can import in the ReadyAPI application. After importing the project, you can run the test by clicking on the run button

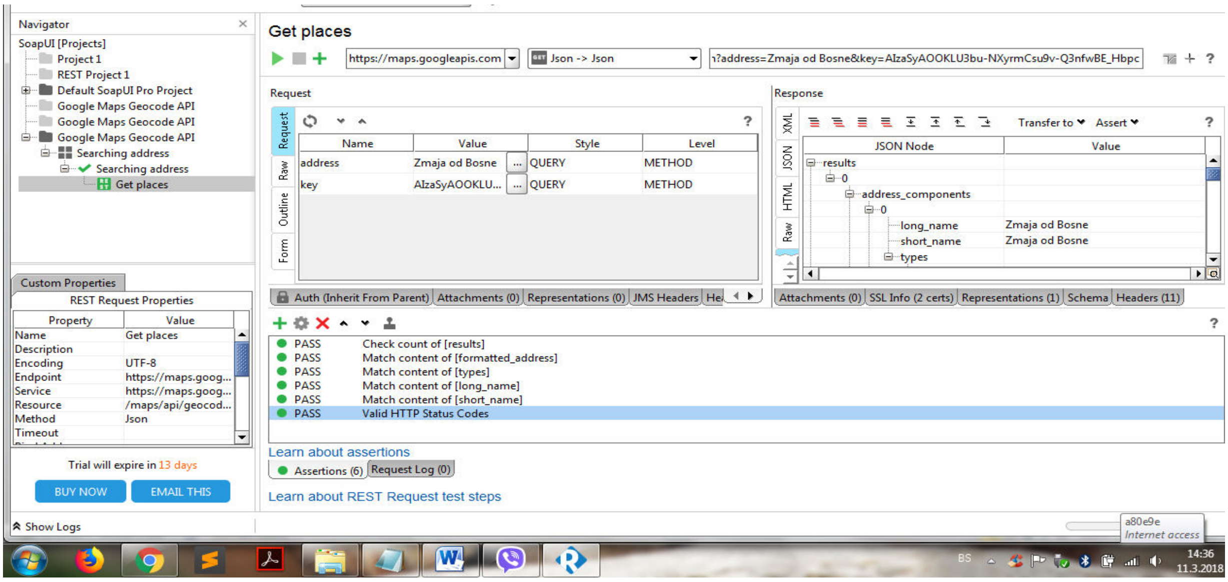


Property	Value
Name	Google Maps Geoc...
Description	
File	C:\Users\Naida\De...
Resource Root	
Cache Definitions	true
Project Password	
Encryption Mode	Not encrypted
Composite Project	false
Composite Project ...	1.3
Script Language	Groovy

Searching address represents a test case, and Get place present a test step.

The parameters in this test case are address and key. The address represents the street address that you want to geocode, in the format used by the national postal service of the country concerned. The key represents the API key that you can get on the web address https://developers.google.com/maps/documentation/geocoding/get-api-key.

In the test step Get place I make some assertions for this request, for example Valid HTTP Status codes, and these assertions will pass if the status code of the request is 200. The other assertions you can see on the next picture:



REST Request Properties

Property	Value
Name	Get places
Description	
Encoding	UTF-8
Endpoint	https://maps.goog...
Service	https://maps.goog...
Resource	/maps/api/geocod...
Method	Json
Timeout	

Get places

https://maps.googleapis.com   Json -> Json   ı?address=Zmaja od Bosne&key=AIzaSyAOOKLU3bu-NXyrmCsu9v-Q3nfwBE_Hbpc

Request

Name	Value	Style	Level
address	Zmaja od Bosne	QUERY	METHOD
key	AIzaSyAOOKLU...	QUERY	METHOD

Response

JSON Node	Value
results	
0	
address_components	
0	
long_name	Zmaja od Bosne
short_name	Zmaja od Bosne
types	

```
PASS        Check count of [results]
PASS        Match content of [formatted_address]
PASS        Match content of [types]
PASS        Match content of [long_name]
PASS        Match content of [short_name]
PASS        Valid HTTP Status Codes
```

If all assertions pass the test, the whole step will pass and that will be successful test case. Otherwise, the whole test case will be unsuccessful.