Google Maps APIs

# Developer's Guide

Test Plan Approach

|  |
| --- |
| Table of Contents  Contents  [Developer's Guide 1](C:\\Users\\Mirzeta\\Desktop\\GoogleMap_API_TestPlan.docx" \l "_Toc470893620)  [Feature Testing Approach 3](#_Toc470893621)  [Audience 3](#_Toc470893622)  [Review Expectations 3](#_Toc470893623)  [Supporting Documentation 3](#_Toc470893624)  [Developer's Guide 3](#_Toc470893625)  [High Level Feature Information 3](#_Toc470893626)  [High Level Testing Information 4](#_Toc470893627)  [Requirements 4](#_Toc470893628)  [Coverage Review 5](#_Toc470893629)  [Scenario 1: Verify latitude and longitude positions using location address 6](#_Toc470893630)  [Scenario 2: Verify geocode returns location address based on inserted latitude and longitude 6](#_Toc470893631)  [Scenario 3: Verify generated JSON file content 6](#_Toc470893632)  [Scenario 4: Verify content of generated XML file 7](#_Toc470893633)  [Scenario 5: Compare JSON and XML files content 7](#_Toc470893634)  [Scenario 6: Verify google map API key 7](#_Toc470893635)  [Scenario 7: Verify developers guide content 8](#_Toc470893636)  [Scenario 8: Verify all guides are presented on the page 8](#_Toc470893637)  [Scenario 9: Verify google maps API status 8](#_Toc470893638)  [Scenario 10: Verify geocode returns zero\_results if you insert invalid address in search field 9](#_Toc470893639)  [Scenario 11:Verify that geocode returns zero\_results when you insert invalid longitude and latitude- reverse geocoding 9](#_Toc470893640) |

Revision & Sign-off Sheet

|  |  |  |  |
| --- | --- | --- | --- |
| Change Record | | | |
| Date | Author | Version | Change Reference |
| 12/28/2016 | Mirzeta Hankic | V1.0 | Requirements, Test scenarios and other details created |

# 

# Feature Testing Approach

This document outlines the high level test plan with all scenarios included.It includes test scenarios about Google App geocoding. It is designed to ensure that QA understands all website functionalities about geocoding and reverse geocoding. It does not have detailed test steps.

## Audience

The audience for this document includes:

* Assigned people from company

## 

## Review Expectations

Dedicated company members are expected to review this document to ensure:

* Test scope and steps are understood by team and ready for testing
* Test scenarios meets expectation and all required fields are covered
* QA specified configuration and all required tools for testing
* Smoke test is indentified and test cases are executed based on priority

## Supporting Documentation

The documentation below should be used to analyse test case scenarios:

| Ref # | Reference Name | Links |
| --- | --- | --- |
|  | Developer's Guide | <https://developers.google.com/maps/documentation/geocoding/intro#Geocoding> |

## High Level Feature Information

**Geocoding** is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers on a map, or position the map.We will use addresses and place on map and look for longitude and latitude and test does google returns right results. We will also use latitude and longitude and place them on map and test positions, does google API returns right position and any results based on our latitude and longitude. This is called reverse geocoding. We will also test negative scenarios too.

[**Reverse geocoding**](https://developers.google.com/maps/documentation/geocoding/intro#ReverseGeocoding) is the process of converting geographic coordinates into a human-readable address. The Google Maps Geocoding API's reverse geocoding service also lets you find the address for a given [place ID](https://developers.google.com/places/place-id).The Google Maps Geocoding API provides a direct way to access these services via an HTTP request. The following example uses the Geocoding service through the Google Maps JavaScript API to demonstrate the basic functionality.

## High Level Testing Information

* QA will be testing scenarios where google map API returns location based on longitude and latitude
* QA Will be testing scenarios where google map API returns latitude and longitude based on address
* QA Will be testing negative scenarios of google map API geocoding
* QA will be testing negative scenarios of reverse google map API geocoding

## Requirements

The following requirements are addressed :

| Ref # | Requirements Description | TFS ID |
| --- | --- | --- |
|  | Access Google Map API: |  |
| 1. 2. | Inserting Longitude and Latitude: exp:  latitude 37.423021 and longitude -122.083739 |  |
|  | Inserting addresss : exp  "1600 Amphitheatre Parkway, Mountain View, CA |  |
|  | Automating test case scenarios using Postman tool |  |

## 

## Coverage Review

Test scenarios, environments and priority are listed below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scenario** | **Description** | **Method** | **Priority** | **Neg/Pos** | **Suite** |
|  | Scenario 1: Verify longitude and latitude positions using location address | Manual | 1 | Pos | Smoke |
|  | Scenario 2: Verify that geocode returns location address based on inserted longitude and latitude- Reverse geocoding | Manual | 1 | Pos | Smoke |
|  | Scenario 3: Verify generated JSON file content | Manual | 1 | Pos | Regression |
|  | Scenario 4: Verify generated XML file content | Manual | 1 | Pos | Regression |
|  | Scenario 5: Compare JSON and XML files content | Manual | 1 | Pos | Regression |
|  | Scenario 6: Verify google maps API key | Manual | 2 | Pos | Regression |
| 7. | Scenario 7: Verify Developers Guide content | Manul | 2 | Pos | Regression |
| 8. | Scenario 8: Verify all guides are presented on webpage | Manual | 2 | Pos | Regression |
| 9. | Scenario 9: Verify Googla Maps API Geocode Status | Manual | 2 | Pos | Regression |
| 10. | Scenario 10: Verify that geocode returns zero results if you insert wrong address in search field | Manual | 2 | Neg | Regression |
| 11. | Scenario 11: Verify that geocode returns zero results if you insert wrong lat and lng in search field | Manual | 2 | Neg | Regression |

**NOTE**: Testing below will be performed in order to check location details returned by geocode. For testing we will use address,longitude and latitude information and insert into search field. Below is list of items which will be tested.

* 1. Longitude and Latitude:

Checking does geocode returns right longitude and latitude based on inserted address.

* 1. Address:

Checking does geocode returns address based on inserted latitude and longitude.

* 1. JSON file :

Verify does JSON file contains all required information and compare them with generated information on website.

* 1. XML file:
  2. Verify that XML file contains all required data and compare it with generated data on website.
  3. Developer Guide Content:

Verify Developer’s guide content.

* 1. Compare XML and JSON files :

Compare and verify do XML and JSON files contain same data.

### Scenario 1: Verify latitude and longitude positions using location address

**SMOKE TEST** –We will need to verify does geocode returns latitude and longitude based on inserted address in search field.

1. **User navigates to:** https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. **User Verifies:** Google Maps API is successfully opened
3. **Insert address in search field exp**: 1600 Amphitheatre Parkway, Mountain View, CA
4. **Click**: Geocode button
5. **User Verifies**: geocode returns status : **„OK“**
6. **User should verify that:** Geocode returns Country, Postal Code, Street Address, Route, Latitude and Longitude, Postal code, exp:

**Address**: 1600 Amphitheatre Parkway, Mountain View, CA,

**Location longitude and latitude**: latitude 37.4220303, lng: -122.0841273

T**ype:** street\_address

### Scenario 2: Verify geocode returns location address based on inserted latitude and longitude

**SMOKE TEST** – We need to verify does geocode returns location address based on inserted latitude and longitude.

1. **User navigates to:** https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. **User Verifies:** Google Maps API is successfully opened
3. **Insert latitude and longitude in search field**: latitude 37.4223632 and longitude : -122.0843854
4. **Click**: Geocode button
5. **Verify**: geocode returns status : **„OK“**
6. **User should verify that:** Geocode returns Country, Postal Code, Street Address, Route, Latitude and Longitude, Postal code, exp:

**Address**: 1600 Amphitheatre Parkway, Mountain View, CA,

**Location longitude and latitude**: latitude 37.422366 and longitude -122.083945,

T**ype:** street\_address

### Scenario 3: Verify generated JSON file content

We will need to verify generated content in JSON file

1. **User Navigates to:** https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. **User Verifies:** Google Maps API is successfully opened
3. **Insert latitude and longitude in search field exp**: latitude 37.423021 and longitude -122.083739
4. **Click**: Geocode button
5. **Verify**: geocode returns status : **„OK“**
6. **User should Click on JSON link and open file**
7. Verify JSON file contains **address\_components node with long\_name, short\_name and type inside it**
8. Verify inside brackets below are placed **long\_name** and **short\_name** (exp **Amphitheatre Parkway value) with route type**
9. Verify that below (**Amphitheatre Parkway) are placed long\_name and short\_name** with value **(Mountain\_View) and** types **[„locality“,“political“]**
10. **User Verifies that postal (94043) code** is displayed in JSON file
11. **User Verifies** that **country name** is displayed in JSON file
12. **User Verifies same longitude and latitude** dimensions are displayed in **JSON file as** on generated PopUP window on site
13. **User Verifies** on the bottom **of JSON file pace\_id and status code** are displayed

### Scenario 4: Verify content of generated XML file

We will need to verify that generated XML file has all required data.

1. **User navigates to:** https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. **User verifies:** Google Maps API is successfully opened
3. **Insert latitude and longitude in search field exp**: latitude 37.423021 and longitude -122.083739
4. **Click**: Geocode button
5. **Verify**: geocode returns status : **„OK“**
6. **User should Click on generated XML link and open file**
7. Verify XML file contains **formatted\_address**, **address\_components node with long\_name, short\_name and type inside it**
8. Verify that XML file contains **long\_name** and **short\_name** (exp **Amphitheatre Parkway value) with route type**
9. Verify that below (**Amphitheatre Parkway) are placed long\_name and short\_name** with value( **Mountain\_View) and** types **[„locality“,“political“]**
10. **Verify that postal (94043) code** is displayed in XML file
11. **Verify** that **country name** is displayed in XML file
12. **Verify same longitude and latitute** dimensions are displayed in **XML file as** on generated PopUP window on site
13. **Verify that place\_id** is displayedon the bottom **of XML file** and on the top of XML file is displayed **status code**

### Scenario 5: Compare JSON and XML files content

We will need to compare content of generated XML and JSON files and analyze do they contain same data.

1. **User navigates to:** https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. **User Verifies:** Google Maps API is successfully opened
3. **Insert latitude and longitude in search field exp**: latitude 37.423021 and longitude -122.083739
4. **Click**: Geocode button
5. **Verify**: geocode returns status : **„OK“**
6. **User should Click on generated JSON and XML links and open files**
7. Verify that JSON and XML files contain same **formatted\_address** with full address, **address\_components nodes with long\_name, short\_name and types inside it**
8. Verify that both JSON and XML files contain same address\_component : **long\_name** and **short\_name** (exp with **Amphitheatre Parkway value) with route type**
9. Verify that both files contain **long\_name and short\_name** with value **(Mountain\_View) and** types **[„locality“,“political“]**
10. **Verify that same postal (94043) code** is displayed in both files
11. **Verify** that same **country name** is displayed in XML file and JSON files
12. **Verify same longitude and latitute** dimensions are displayed in **XML and JSON files**
13. **Verify** that both files have same **place\_id** and **status\_code** generated

### Scenario 6: Verify google map API key

We will need to verify is google map API key generated successfully

1. **User navigates to**: <https://developers.google.com/maps/documentation/geocoding/intro#Geocoding>
2. **User Verifies:** Google Maps API is successfully opened
3. **Verify**- right side on the top of the page : **Get a Key and View Pricing and Plans buttons**
4. **Click**: Get a Key button
5. **Verify**: PopUp window with **Select or Create New** project is displayed
6. User enters : **New Project Title** and Clicks: **Create and Enable API button**
7. **Use Verifies**: API key is generated and Resources are displayed
8. **User Clicks**: API Console and **Verifies**: That project dashboard is displayed and ready for developing

### Scenario 7: Verify developers guide content

We will need to verify does developer guide contains all required data

.

1. User navigates to: https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. User Verifies: Google Maps API is successfully opened
3. **Verif**y: that Google Maps API Geocode is below Documentation tab on the website
4. **Verify**: Home, Documentation, Pricing and Plans tabs on the top of the page and that we can open them
5. **Verify**: Before You Begin Topic is displayed on the page with all required links
6. **Verify** : Geocoding (Latitude/Longitude Lookup) description is displayed on the page
7. **Verify**: Geocoding response examples in JSON and XML are displayed on the page
8. **Verify**: list with Status Code types is displayed together with error messages on the page
9. **User Verifies**: Results, Address Types, View Port Biasing, Region Biasing, Component Filtering, Required parameters in reverse geocoding, optional parameters in reverse geocoding examples are displayed

### Scenario 8: Verify all guides are presented on the page

We will need to verify do all guides exist on the page

1. Navigate to: https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. Verify: Google Maps API is successfully opened
3. Verify: Developers Guide and What is Geocoding description are displayed on the page
4. **Verify**: Google Map for Geocoding is displayed on the page
5. **Verify**: on Top Left side of website GUIDES and SUPPORT tabs and that we can access them
6. **Verify**: on left panel : Get Started, Developer’s Guide, Best Practices, Geocoder Faq, Get a Key, Usage limits, Optimizing Quota Usage, Policies, Terms of Services, Google Maps Web Services, Other APIs and that we can access them
7. **Verify**: Resources, Platforms, Product Info and Google Developers options are placed on the bottom of the page and that they are accessible.

### Scenario 9: Verify google maps API status

We will need to verify status returned by geocode

1. **Navigate to:** <https://developers.google.com/maps/documentation/geocoding/intro#Geocoding>
2. **Verify:** Google Maps API is successfully opened
3. **Insert addresss in search field** : exp: "Stupska 1a“
4. **Click**: Geocode button
5. **Verify**: geocode returns status : **„OK“ (The request did not ecounter any errors)**
6. **User should verify that:** Geocode returns Country, Postal Code, Street Address, Route, Latitude and Longitude, Postal code, exp:

**Address**: Stupska 1D, Sarajevo 71000, Bosna i Hercegovina,

**Location longitude and latitude**: latitude 43.841426,longitude 18.325002,

**Type**: street\_address

1. **Verify**: that results are represented in JSON| XML
2. **Click** on **JSON and XML** links and verify that they are not empty
3. **Verify** that your searched location is selected and inside square on the map
4. **Click** anywhere inside selected square and verify that locations are added with **adresses, longitude, latitude and types in alphabetical order A,B,C,D,E...**

### Scenario 10: Verify geocode returns zero\_results if you insert invalid address in search field

We will need totest negative scenario of geocode when it returns zero\_results.

1. **Navigate to:** https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. **Verify:** Google Maps API is successfully opened
3. **Insert invalid address in search field** : exp "Latička 1a“
4. **Click**: Geocode button
5. **Verify**: geocode returns status : **„ZERO\_RESULTS“**
6. **Verify:** that request did not encounter any errors but returns zero results

### Scenario 11:Verify that geocode returns zero\_results when you insert invalid longitude and latitude- reverse geocoding

We will need to test does geocode returns zero\_results when we insert invalid longitude and latitude values.

1. **Navigate to:** https://developers.google.com/maps/documentation/geocoding/intro#Geocoding
2. **Verify:** Google Maps API is successfully opened
3. **Insert invalid longitude and latitude in search field** : exp

lat**itude 37.423021 and longitude -**222.083739

1. **Click**: Geocode button
2. **Verify**: geocode returns status : **„ZERO\_RESULTS“**
3. **Verify:** that request did not encounter errors but returns zero results