**Postman tool**

**Automated Smoke Tests**

Postman is complete toolchain for API development. It is designed to support all types of API development. It is built on a singled underlying layer. Postman is ensures consistent performance and user experience. It contains next features: Run test, Document & Monitor features and they are designed to work together. It is used to make our work faster and better.

We automated our smoke tests using Postman:

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

**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**: 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:

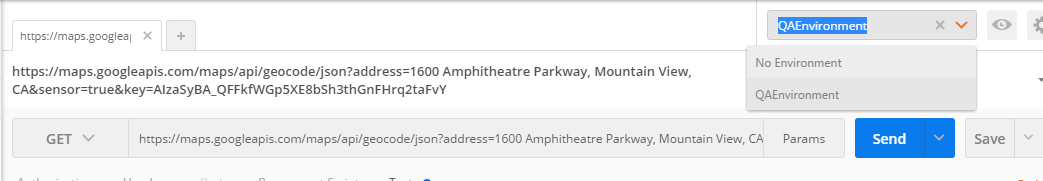
**Address**: 1600 Amphitheatre Parkway, Mountain View, CA,

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

T**ype:** street\_address

**Automated Test Case steps and results:**

1. Create QA environment in Postman:



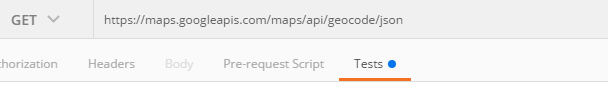
1. Navigate on Google Maps API website: <https://developers.google.com/maps/documentation/geocoding/intro?place_id=ChIJJ9Gx3C_KWEcRM1_dURuabrk>

- click on Get a KEY button

- create new project

- generate key

1. In Postman request tab insert Google Maps API url: <https://maps.googleapis.com/maps/api/geocode/>
2. Define output file type /json or /xml



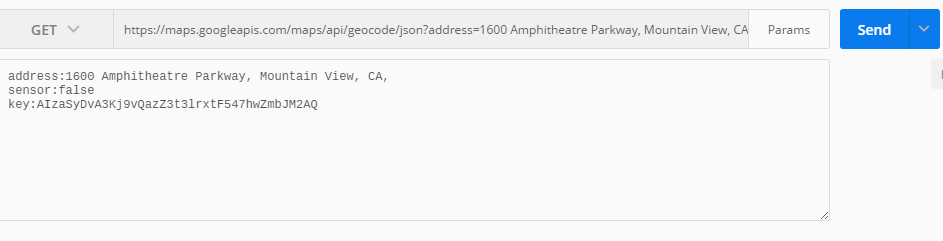
1. Click on Params tab and choose bulk edit

Insert in edited field address you are searching for, sensor and key from your API:

**address**: 1600 Amphitheatre Parkway, Mountain View

**sensor**: false

**key**: AIzaSyDvA3Kj9vQazZ3t3lrxtF547hwZmbJM2AQ



1. In Test field type code for checking status\_code:

tests["Status code is 200"] = responseCode.code === 200;

Write tests:

tests["Body matches street\_number"] = responseBody.has("street\_number");

tests["Body matches address components route"] = responseBody.has("route");

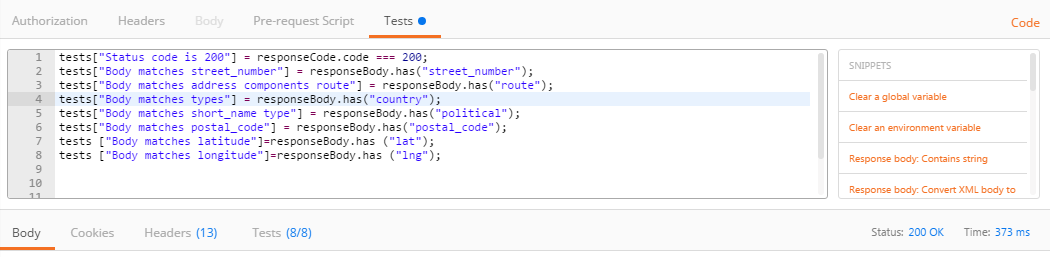
tests["Body matches types"] = responseBody.has("country");

tests["Body matches short\_name type"] = responseBody.has("political");

tests["Body matches postal\_code"] = responseBody.has("postal\_code");

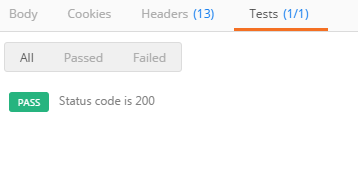
tests ["Body matches latitude"]=responseBody.has ("lat");

tests ["Body matches longitude"]=responseBody.has ("lng");



1. Run Test
2. Verify Geocodes returns status code OK:

"status": "OK"



1. **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

"formatted\_address": "1600 Amphitheatre Pkwy, Mountain View, CA 94043, Sjedinjene Države",

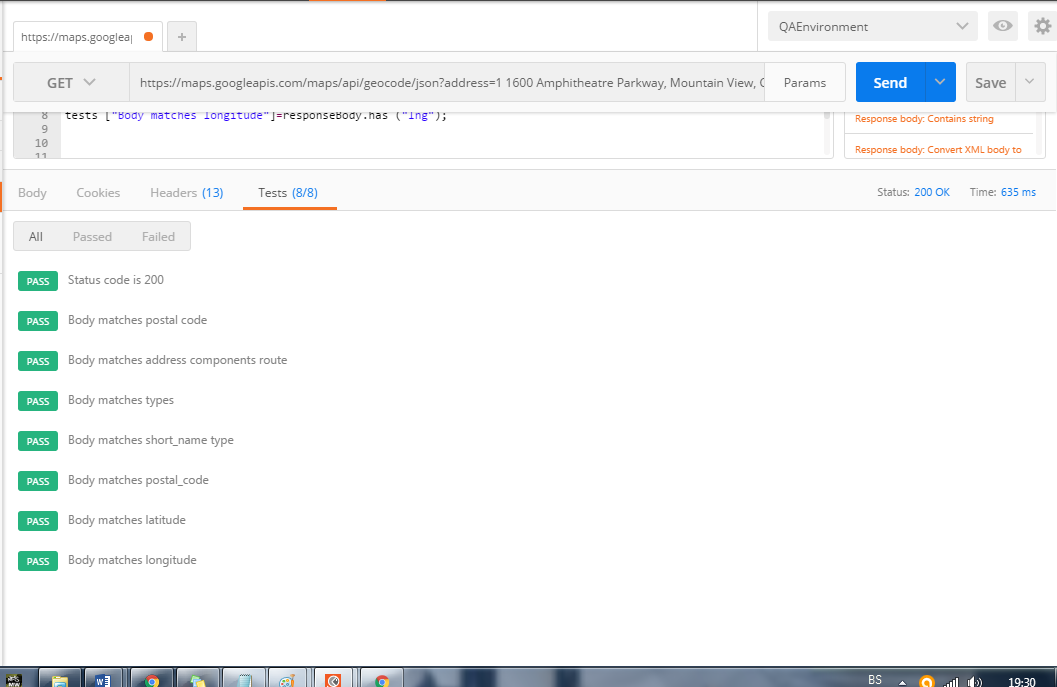
"geometry": {

"location": {

"lat": 37.4220303,

"lng": -122.0841273

},



### 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 exp**: 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:

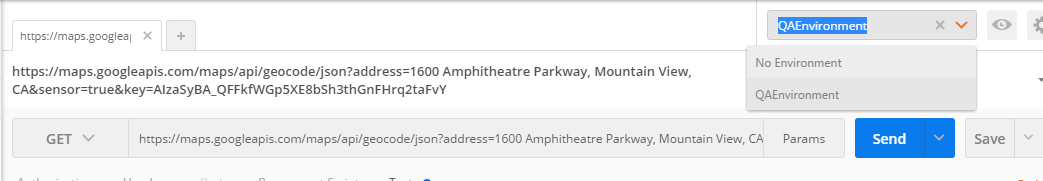
**Address**: 1600 Amphitheatre Parkway, Mountain View, CA,

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

T**ype:** street\_address

**Automated Test Case 2 steps and results:**

1. Use QA environment already created in Postman:



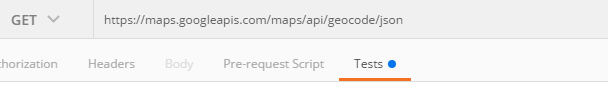
1. Navigate on Google Maps API website: <https://developers.google.com/maps/documentation/geocoding/intro?place_id=ChIJJ9Gx3C_KWEcRM1_dURuabrk>

- click on Get a KEY button

- Use already created project

- generate key

1. In Postman request tab insert Google Maps API url: <https://maps.googleapis.com/maps/api/geocode/>
2. Define output file type /json or /xml



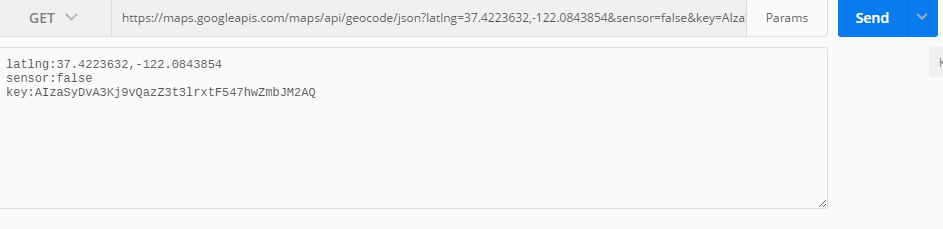
1. Click on Params tab and choose bulk edit

Insert in edited field latitude and longitude you are searching for, sensor and key from your API:

**latlng:** 37.4223632,-122.0843854

**sensor**: false

**key**: AIzaSyDvA3Kj9vQazZ3t3lrxtF547hwZmbJM2AQ



1. In Test field type code for checking status\_code:

tests["Status code is 200"] = responseCode.code === 200;

Write tests:

tests["Body matches street\_number"] = responseBody.has("street\_number");

tests["Body matches address components route"] = responseBody.has("route");

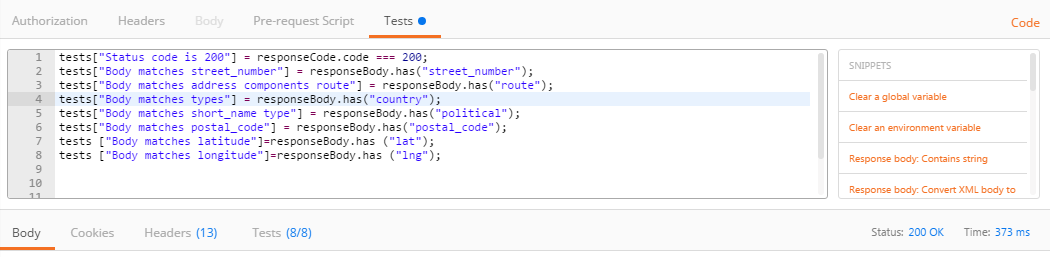
tests["Body matches types"] = responseBody.has("country");

tests["Body matches short\_name type"] = responseBody.has("political");

tests["Body matches postal\_code"] = responseBody.has("postal\_code");

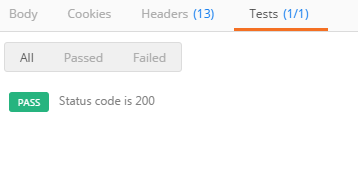
tests ["Body matches latitude"]=responseBody.has ("lat");

tests ["Body matches longitude"]=responseBody.has ("lng");



1. Run Test
2. Verify Geocodes returns status code OK:

"status": "OK"



1. **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

"formatted\_address": "1600 Amphitheatre Pkwy, Mountain View, CA 94043, Sjedinjene Države",

"geometry": {

"location": {

"lat": 37.4220303,

"lng": -122.0841273

},

