

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

```
/*jshint esversion: 6*/
// functions created based on developers.google documentation
// https://developers.google.com/maps/documentation/places/web-service/supported_types
// and following tutorial Bobby did coding
// https://www.youtube.com/watch?v=wCn8WND-JpU&t=8s
// and refactored to the needs of the project

$.getScript( "https://maps.googleapis.com/maps/api/js?key=" + google_api_key + "&libraries=places"
.done(function( script, textStatus ) {
    google.maps.event.addDomListener(window, "load", initAutocomplete());
});

let autocomplete_user;
let autocomplete_employer;

/**
 * function that initializes autocomplete from google maps places API
 * user can search by postcode or full address
 * once API has been fetched successfully the div displays to the user that
 * the field passed validation successfully
 */
function initAutocomplete() {

    autocomplete_user = new google.maps.places.Autocomplete(
    document.getElementById("id_your_address"),
    {
        regions: ["postal_code"],
        componentRestrictions: {"country": ["uk"]}
    });

    autocomplete_user.addListener("place_changed", function(){
        onPlaceChanged("user");
    });

    autocomplete_employer = new google.maps.places.Autocomplete(
    document.getElementById("id_employer_address"),
    {
        regions: ["postal_code"],
        componentRestrictions: {"country": ["uk"]}
    });

    autocomplete_employer.addListener("place_changed", function(){
        onPlaceChanged("employer");
    });
}

/**
 * function that listens for input / place changed in the start and destination fields and
 * autocompletes the address and latitude and longitude fields
 */
function onPlaceChanged (addy){

    let auto;
    let el_id;
    let lat_id;
    let long_id;

    if ( addy === "user"){
        auto = autocomplete_user;
        el_id = "id_your_address";
        lat_id = "id_latitude";
        long_id = "id_longitude";
    }
    else{
        auto = autocomplete_employer;
        el_id = "id_employer_address";
        lat_id = "id_employer_latitude";
    }
}
```

CONFIGURE

JSHint

Metrics

version 2.13.3

There are 6 functions

(<https://github.com/jshint/jshint/>)

Function with the largest number of lines is 83

median is 0.5.

Largest function has 2.5.

Documentation (/docs)

Install (/install)

The most complex function has 3

Contribute (/contribute)

of 3 while the median is 1

Blog (/blog)

Two warnings

83 Functions declared but not used (google, \$, lat_id, long_id)

79 The body of a function is empty (to filter unwanted results)

Three undefined variables

9 \$

89 \$

90 \$

9 google_api_key

11 google

25 google

37 google

74 google

85 google

```
71     long_id = "id_employer_longitude";
72 }
73
74 var geocoder = new google.maps.Geocoder();
75 var address_user = document.getElementById("id_your_address").value;
76 var address_employer = document.getElementById("id_employer_address").value;
77 var address_list = [address_user, address_employer];
78
79 for (var address in address_list) {
80     address = document.getElementById(el_id).value;
81
82
83     geocoder.geocode( { "address": address}, function(results, status) {
84
85         if (status == google.maps.GeocoderStatus.OK) {
86             var latitude = results[0].geometry.location.lat();
87             var longitude = results[0].geometry.location.lng();
88
89             $("#"+ lat_id).val(latitude);
90             $("#"+ long_id).val(longitude);
91
92         } else {
93
94             alert("Directions request failed due to " + status);
95             window.location.assign("/users/");
96
97         }
98     });
99 }
100 }
101 }
102 }
103 }
```



version 2.13.3
There are 6 functions
(<https://github.com/jshint/jshint/>)
Function with the large
median is 0.5.
About (about)
Largest function has
Documentation (/docs)
2.5.
Install (/install)
The most complex fu
Contribute ((contribute)
of 3 while the median
Blog (/blog)

Two warnings

- 83 Functions declar
scoped variable
(google, \$, lat_ic
- 79 The body of a fo
to filter unwanted

Three undefined v

- 9 \$
- 89 \$
- 90 \$
- 9 google_api_key
- 11 google
- 25 google
- 37 google
- 74 google
- 85 google