droidcon academy

Geocoding Service: Address Autocomplete with OpenStreetMap (OSM) in Flutter

John Osezele

@john_osezele

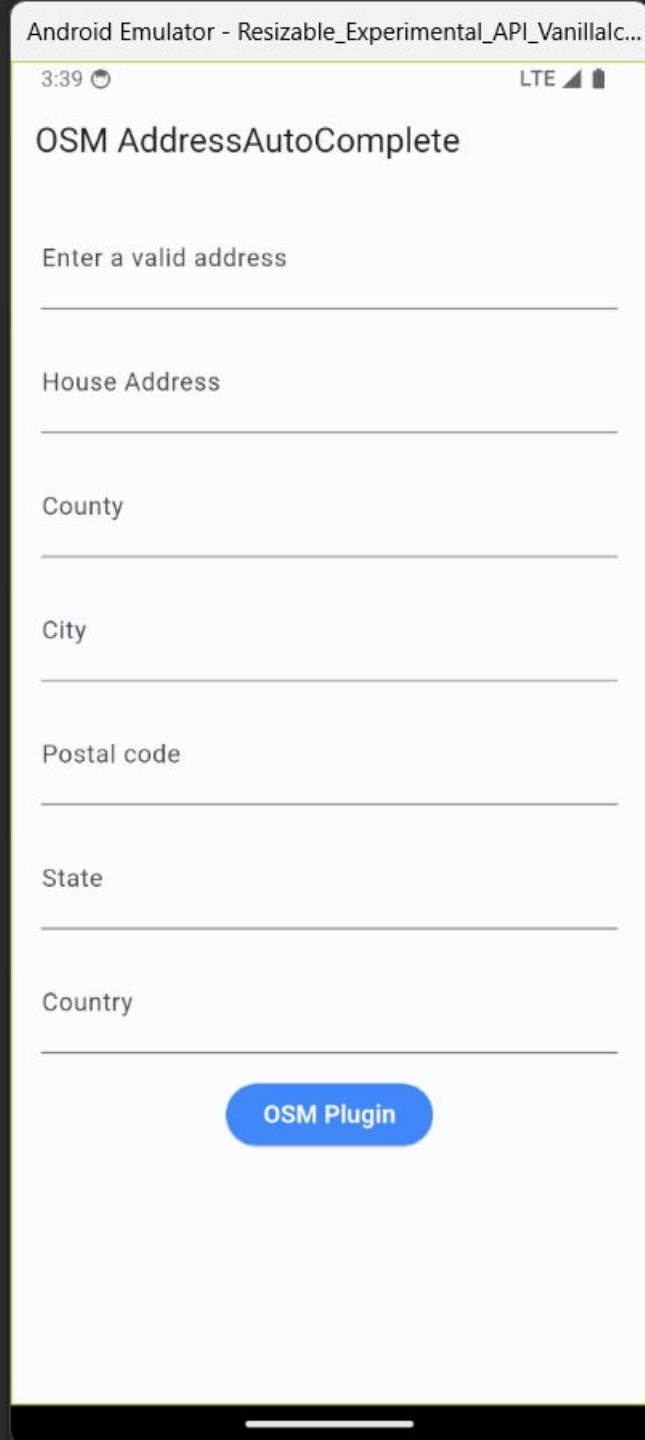
in/johnosezele

Overview

- Introduction to Geocoding in Flutter
- Explore the OpenStreetMap documentation
- Model the Address response
- Create the OpenStreetMap service class
- Populate the Address data into required address fields
- Code challenge
- Address search using the flutter_osm_plugin

Course Demo App

Demo App: Display address suggestions as user types, populate required address fields when user selects address. Get addresses using OSM plugin.



Course Demo App

Demo App: Display address suggestions as user types, populate required address fields when user selects address. Get addresses using OSM plugin.

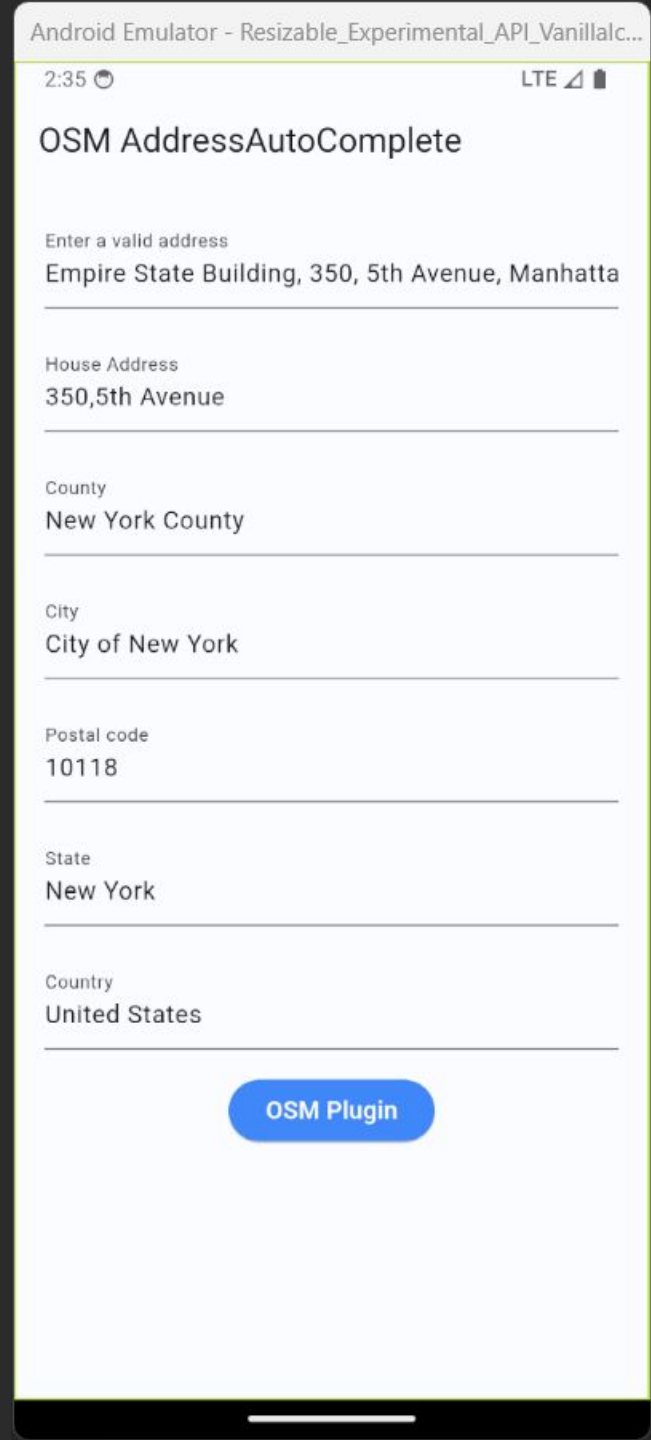
Course Demo App

Demo App: Display address suggestions as user types, populate required address fields when user selects address. Get addresses using OSM plugin.

Course Demo App

Demo App: Display address suggestions as user types, populate required address fields when user selects address. Get addresses using OSM plugin.

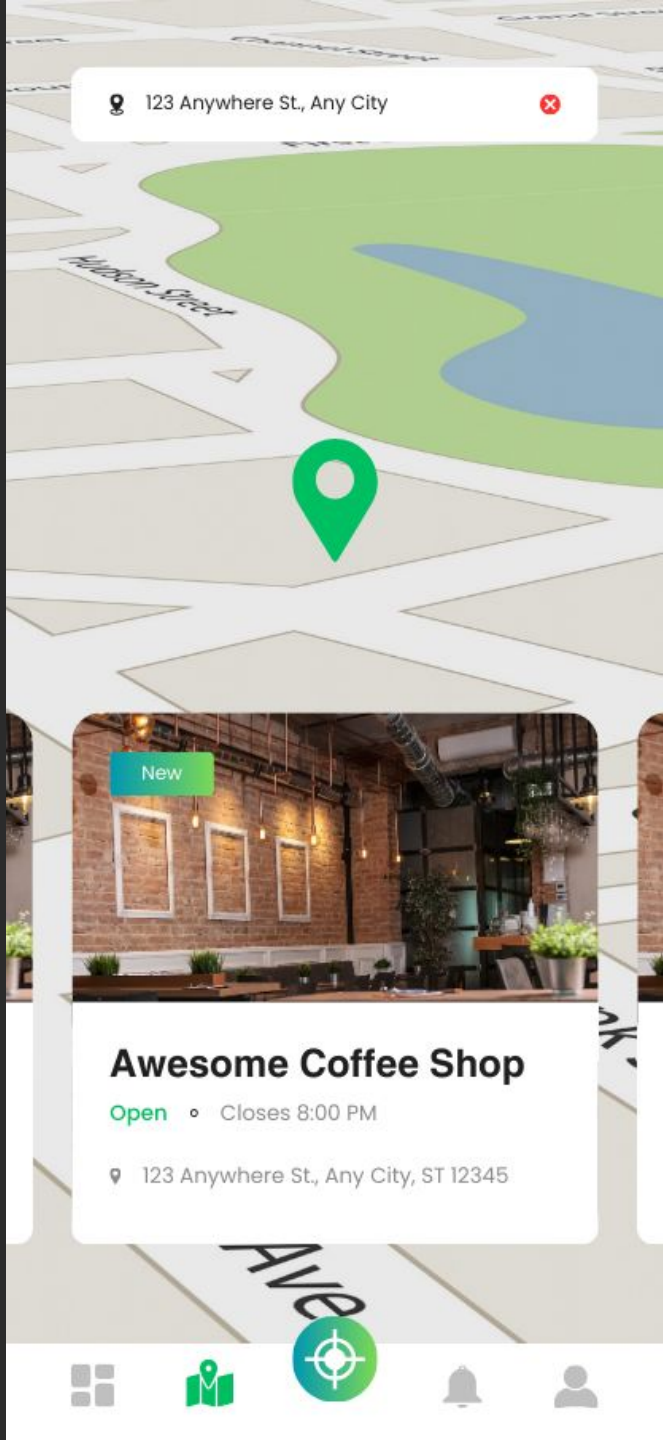
- From our demo app, we have been able to:
  - Search for an address.
  - Tap on our preferred result.
  - Populate the text fields with the corresponding address data.
  - Perform an address search using the Flutter OSM Plugin.



What is Geocoding?

droidcon academy

- Imagine you're building an app to help users discover new coffee shops.
- Geocoding enables address autocomplete.
- Users can quickly retrieve coffee shop suggestions based on their input.
- Reduces typing errors and ensures accurate locations.



Geocoding Services Available in Flutter:

- Google Maps Geocoding API
- Mapbox Geocoding API
- Photon API
- TomTom Geocoding API
- Bing Maps Geocoding API
- LocationIQ Geocoding API
- HERE Geocoding & Search API
- Yandex Geocoding API
- OpenCage Geocoding API

Search URL

- The baseUrl is our OSM endpoint.
- Our 'format' is set to 'json' which works best for our scenario.
- Our query param, 'q' is a free form query which is used to query our address.
- 'countrycodes' parameter limits search result to country.
- 'addressdetails' is set to 1, to include a breakdown of the address into elements.
- 'limit' is used to limit the maximum number of returned results.
- 'namedetails' set to 1, include a full list of names for the result.
- 'extratags' is set to 1, to include any additional information in the result that is available in the database.

```
String baseUrl =
 "https://nominatim.openstreetmap.org";

 final url = "$baseUrl/search?"
  +"format=json"
  +"&q=$address"
  +"&countrycodes=$countryCode"
  +"&addressdetails=1"
  +"&limit=10"
  +"&namedetails=1"
  +"&extratags=1";
```

GET requests using HTTP Library

- ‘Uri.parse()’: Parses the URL we want to send request to.
- ‘http.get()’: Initiates ‘http’ get request
- ‘await’: The code waits for the response from the server to be ready before going further in the execution
- This code snippet is sending a GET request to a specified URL with certain query parameters and awaits the response.

```
final response = await http.get(
Uri.parse(
 "$baseUrl/search?"
  +"format=json"
  +"&q=$address"
  +"&countrycodes=$countryCode"
  +"&addressdetails=1"
  +"&limit=10"
  +"&namedetails=1"
  +"&extratags=1";
    )
 );
```

droidcon academy

GET requests using DIO Library

- ‘final url = …’ : This block constructs the URL string where the GET request will be sent.
- ‘Dio()’: Creates a new instance of the Dio client, which is used to send HTTP requests
- ‘.get(url)’: Sends a GET requests to the specified URL
- This code snippet constructs a URL with query parameters and then uses the DIO library to send a GET request to that URL.

```
final url =
 "$baseUrl/search?"
  +"format=json"
  +"&q=$address"
  +"&countrycodes=$countryCode"
  +"&addressdetails=1"
  +"&limit=10"
  +"&namedetails=1"
  +"&extratags=1";

 final response = await Dio().get(url);
```

Clone Starter Project

Up next

droidcon academy

Section Summary

- An introduction to Geocoding in Flutter
- We explored the OpenStreetMap documentation
- We explored how to make network calls using the HTTP & DIO libraries
- Then we modeled our address response

droidcon academy

Entering Addresses and Displaying the Search Suggestions

John Osezele

@john_osezele

in/johnosezele

Section Overview

- Create the OpenStreetMap Service class
- Populate our address fields with the required address data
- Perform the coding challenge
- Perform an address search using the flutter osm plugin

Code Challenge ⚙️

droidcon academy

Code Challenge Solution 💡

droidcon academy

Summary

- Using Dio and HTTP library for OpenStreetMap API requests
- Fetching address suggestions dynamically
- Displaying address suggestions in a dropdown as the user types
- Implementing Flutter OSM plugin for geocoding and address search

Section Summary

- Understood the concept of Geocoding in Flutter
- Understood how to query the OpenStreetMap URL for address search results
- Learned how to create a model for the address response
- Learned how to create the OpenStreetMap Service class
- Populated the address data into required address fields
- Performed a code challenge
- Performed address search using the OpenStreetMap Plugin

droidcon academy

THANK YOU

John Osezele

@john_osezele

in/johnosezele