

Backend Developer Test

Problem:

Build a Django application that will connect with 3 external weather services and provide an average temperature for a given zip lat/lon.

The Django application should have a single url route that takes in a latitude, longitude, and a list of external services to filter on. The acceptable filters will be 'noaa', 'weather.com', and 'accuweather'. For example: if the user sends in 'noaa' and 'accuweather' in the filter list, then only those two services will be used to calculate the average temperature for the given lat/lon.

In order to connect with the 3 external APIs, we have created a simple Flask application that you will run and connect to. This will prevent you from having to actually integrate with three external providers. Please access this application and view the readme here:

<https://github.com/otterlogic/mock-weather-api>

Although this is a simple application, please use architecture and design patterns as you would for a larger and more complex project.

Guidelines:

- Use Django
- Create a url route that accepts: latitude, longitude, and filters
- Filter the external providers depending on the user input filters
- The response to the request will be a json response with the average current temperature
- Create a repository on GitHub for your code and share the link with us. (if the repo is private share with 'nfortenberry')

Bonus (not required):

- Validate the lat/lon with the Google Maps API

- Also accept zip codes and convert to lat/lon using the Google Maps API

This test will take approximately 3 hours but feel free to spend more or less time on it. Please let us know when you will be able to deliver your implementation.