
BACKEND DEVELOPER TEST

Summary

You will be tasked with creating a minimal inventory app along with a sync mechanism.

The information stored by the app is a stock reading. In a shop an employee can for any reference, enter the current shortest expiry date present on the shelves.

Example: a store sells a soda. On display, there are bottles expiring on the 10th of October and others expiring on the 13th of October. The shortest date is 10th of October and we call Stock Reading the information that at this moment, the soda is available in this shop with 10th of October as its closest expiry date.

The app should be able to display the current most up to date stock reading for a particular reference and also a history off all StockReadings.

The app should also expose a sync mechanism which would be used by a mobile app to save data created when offline and retrieve data created on the web app.

Requirements

Create a Django application backed by a PostgreSQL database.

We consider that the Django app handles a single shop for a single user. So, no login is required.

The app can be open simultaneously on multiple navigators and display the most up to date info after a refresh.

The following features are expected:

- Create a new StockReading with at least a reference id and an expiry date. A reference id means the 13 digits below the barcode on any product.
- Display a list of all current StockReadings.
- Display a list of all past StockReadings when clicking on a current StockReading

The design is up to you and is not the main concern.

The sync mechanism can use any technology or protocol you choose. The end game is that a mobile app can at any time synchronize its local database with the backend to make sure that both the web app and the mobile app have the same data. The latest entered data should prevail, whether entered on the mobile app or web app.

If you don't have enough time to code the sync mechanism, you can provide a document detailing how such feature can be implemented.

Expected Result

1. Estimate the time needed to develop the backend. Split this estimate in small tasks as needed. Send us this estimate as soon as possible.
2. Deploy the Django app on a serveur of your choice.
3. Send us:
 - a. The URL of the app
 - b. A link to your remote git repository (needs to be public so we can clone it)
 - c. An explanation of the process to build the app locally
 - d. A short text explaining your technical decisions and possible areas of improvement.