Developing the Patient- and Provider- Facing Applications, and the Recommendations Engine

Team Grade-A-FHIR
Josh Woo, Patrick Barringer, Krishna Nadimetla, & Sunitha Dasari
https://github.gatech.edu/gt-hit-fall2016/Childhood-Obesity
OMSCS6440 Fall 2016

Setting up your development environment

Instructions for running the applications locally are provided in the README.md files for each application. Please follow the instruction there to set up your development environment.

FHIR

Our applications use the FHIR test server operated by the Michigan Health Information Network (MiHIN):

http://52.72.172.54:8080/fhir/home

This server will be regularly purged and reloaded with newly generated test data. When this happens, it would be necessary to reload our test data onto the server. Unfortunately, this would generate new IDs for resources so it would also be necessary to update the code.

The following sections describe how to do this.

Reloading test data required to run the patient- and provider- facing applications

A script for loading our test data to the MiHIN operated FHIR test server is available.

To run the script:

1. Go to the root of the application folder, e.g.

```
/git-repos/gt-hit-fall2016/Childhood-Obesity/patient-facing-app/
```

2. If you have not already created your Python Virtual Environment:

```
$ virtualenv fghenv
```

3. Activate your Python Virtual Environment:

```
$ source fghenv/bin/activate
```

4. If you have not already done so, install Django and other dependencies:

```
$ pip install -r requirements.txt
```

5. Run the following command:

```
$ python manage.py load_resources MiHIN --include-patients --include-questionnaires --include-organizations
```

6. Go to http://52.72.172.54:8080/fhir/home to locate the newly created resources and record their IDs. Look for the following:

    - Patient “Clark Kent”
    - Patient “Kara Kent”
    - Patient “Diana Prince”
    - Patient “Beatrice Prior”
    - Patient “Tobias Eaton”
    - Questionnaire “Healthy Habits Tracker (adolescent)”
    - Questionnaire “Healthy Habits Questionnaire”
    - Questionnaire “Food Insecurity Questionnaire”
    - Questionnaire “WIC Child Nutrition Questionnaire”
    - Questionnaire “Healthy Habits Goal Questionnaire”
    - Questionnaire “Healthy Habits Goal Status”
    - Organization “MD”
    - Organization “Patient Care Coordinator”
    - Organization “WIC”

Updating the code

Fortunately, the only application that needs to be updated is the patient-facing app. The file to update is db_references.py in the /json_data/tools/ directory, e.g.:

```
/git-repos/gt-hit-fall2016/Childhood-Obesity/patient-facing-app/json_data/tools/db_references.py
```

Example:

```
REF_CLARK = '2019276'
REF_KARA = '2019280'
REF_DIANA_PRINCE = '2019277'
REF_BEATRICE_PRIOR = '2019278'
REF_TOBIAS_EATON = '2019279'

REF_HHA_TEEN = '2019281'
REF_HHA_CHILD = '2019282'
REF_WIC = '2019284'
```

```
REF_GOALS = '2019285'
REF_STATUS = '2019286'
REF_INSECURITY = '2019283'

REF_WIC_ORG = '2019289'
REF_MD_ORG = '2019287'
REF_PATCO_ORG = '2019288'
```

Google Places API

In addition to FHIR, our Recommendations Engine also uses the Google Places API:

https://developers.google.com/places/

The code in GitHub contains our API key which has a limit of 1,000 queries per day. The next team that continues with this work may want to provision their own API key, enable the required APIs in Google's API Manager and update the API key in the Recommendations Engine's configuration file located at:

```
/git-repos/gt-hit-fall2016/Childhood-Obesity/recs-engine/app/config.json
```

The required APIs are:

- Google Maps Geocoding API
- Google Places API Web Service

Deploying the applications

This semester (Fall 2016), we chose to deploy the applications to Ubuntu 14.04 virtual servers running in the cloud. We used one server per application.

Patient-facing Application

Instead of running:

```
python manage.py runserver
```

as you would when running the application locally for development, we strongly recommend serving the application using Apache + mod_wsgi in production. To do this, follow the instructions here:

https://www.digitalocean.com/community/tutorials/how-to-serve-django-applications-with-apache-and-mod_wsgi-on-ubuntu-14-04

Provider-facing Application

The provider-facing application is a Single-Page Application with no server-side logic. As such, it can be hosted on any cloud storage (e.g. Amazon S3) but we chose to serve the files from Apache.

Recommendations Engine

Instead of running:

```
node app/server.js
```

as you would when running the application locally for development, we strongly recommend using "forever":

https://www.npmjs.com/package/forever

To install "forever":

```
$ [sudo] npm install forever -g
```

Once "forever" is installed, to start the Recommendations Engine:

```
$ forever app/server.js
```

We also strongly recommend that you use Apache as a reverse proxy to Node.js in production. To set this up, see:

http://shinysparkly.com/blog/2015/05/31/node-in-apache/