ENG 100L Cruz Roja Notebook

Oliver Reyes
A10680065

Table of Contents

Individual Documentation

Leaflet Documentation	3
Django Template Documentation	5
Mac Environment Setup	6
Notes/TODO 11/9	7
Notes/TODO 11/16	8
Notes/TODO 11/25	9

Meeting Notes

10/4	10
10/11	11
10/18	13
10/25	14
11/1	15
11/8	16

Contact Information

Individual Documentation: Leaflet

Leaflet Doc

STARTING:

Need -

- Leaflet CSS file - in header: <link rel="stylesheet" href="https://unpkg.com/leaflet@1.0.1/dist/leaflet.css" />
- Leaflet JS file - <script src="https://unpkg.com/leaflet@1.0.1/dist/leaflet.js"></script>
- div for map: <div id="mapidname"></div>
- map container height in CSS: #mapidname { height: 180px; }

Setup complete

- making map: var mymap = L.map('mapid').setView([51.505, -0.09], 13); //long and lat
- add tile layer: L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token={accessToken}', {
- attribution: 'Map data &copy; <a href="http://openstreetmap.org">OpenStreetMap</a> contributors, <a href="http://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, Imagery © <a href="http://mapbox.com">Mapbox</a>',
- maxZoom: 18,
- id: 'your.mapbox.project.id',
- accessToken: 'your.mapbox.public.access.token'
- }).addTo(mymap);
-
-
- **Events:**
- event methods (set of methods shared between event-powered classes like Map. Events allow you to execute a function when something happens with an object. If user clicks map, map 'click' event executes.
- -big ones:
- addEventListener(<String> type, <Func> fn, <Object> context?) returns this
- - adds listener FN to particular event TYPE of the OBJECT. *Alias on(…)
- **Event Properties**
- -Event (base event object. All event objects contain properties)
- - type (type String) - event type, (click, hover)
- - target (type Object) - object that fired event (click?)
- Possibly useful properties
- - LocationEvent: latlng - geographic loc of user, speed, timestamp
-
-
- **Class**
- L.Class

- -initialize method is class’s constructor func (it gets called when you do new MyClass(…)).
-
-
- Class Factories (leaflet objects created w/o NEW keyword. This is bc of lowercase factory method).
- new L.Map(‘map’); //becomes
- L.map(‘map’); // make Map lowercase, drop NEW
- // L.map = function (id, options) { return new L.Map(id, options); };
-
-
- **Markers**
- L.marker([latlng, latlng], options?).addTo(map); // add marker
- Methods:
- -getLatLng() // return current coordinates
- -update() //updates marker position
- -bindPopup(<String> html/ html elmt|<Popup> popup, <Popup options> options?)
- -openPopup()
- -setPopupContent(<String>html | HTML elmt);
- **Popup**
- marker.bindPopup(popupContent).openPopup(); // bind and open popup when clicked
-
-
- leaflets.com/reference.html#events

Individual Documentation: Django Templates

Django Doc

Template extending: Use same parts of HTML for different pages of site

- create a base template (base.html)
    - most basic template to extend
    - blog/templates/base.html, post_list.html
    - {% block content %} //starts the area that will have HTML inserted in it
    - {% endblock %}
    - HTML will come from another template that extends base.html
    - put those tags around where you want the added content and the actual block content, so at least 2 files
- Connect templates:
    - {% extends 'blog/base.html' %}

Class Based Views

- *TemplateView and RedirectView*: most basic views
- can just call from urls file and add key arguments if basic
    - if you want to do more, you can inherit off that view
- //url file: from django.views.generic import TemplateView, HAVE TO IMPORT CLASSES
- // url patterns func — TemplateView.as_view(template_name=''),
- RedirectView: redirects to right place everytime
- *ListView and FormView*

Individual Documentation: Cruz Roja Macbook Environment Setup

* Install Python 3+
* Install Django
* Install Postgres App

* pip3 install psycopg2

* if error correct the path:
PATH="/Applications/Postgres.app/Contents/Versions/latest/bin:$PATH"

* brew install gdal

* python3 manage.py (make sure you see command list)

* pip3 install django-braces

* sudo pip3 install Pillow

* python3 manage.py makemigrations

* python3 manage.py migrate

* python3 manage.py createsuperuser (fill in username, email, password)

* python3 manage.py loaddata doc/data.json

* python3 manage.py runserver

* In browser address box put: http://localhost:8000/craed/ and you should see the application

Individual Documentation: Notes/TODO 11/9

aed_detail = where does aed.description come from?
aed_list = data.___ from data.json file?

use .getJSON to retrieve data from json object

***views.py: class AEDListView
// how we can get aed.name etc
model = AED
context_object_name = ‘aed_list’
getJSON is used for ajax call to pull from webpage

*** LeafletWidget.js = where to implement map via JS

- make popups only exist on EVENT (click). so it doesnt waste memory

*** Use templates
*** aed_list : get points for markers:

```
 // add marker using templated data from views
  var marker = L.marker([{{ aed.latitude }},
                        {{ aed.longitude }}]).addTo(mymap);
```

Individual Documentation: Notes/TODO 11/16

GOT base connected to views
-NEED: populate markers according to data from tables

- template for loops!!! {{% for ambulance in ambulance_list %}}
- NEED TO ADD DATA, no data in ambulance database DOH

DONE

Individual Documentation: Notes/TODO 11/26

Need to :
Use AJAX request to pull from database every few milliseconds //DONE
Change icon to ambulance or car //DONE
Make clicking on list center map on ambulance marker //DONE

Need to:
Make clicking on list center on ambulance marker after it has moved

Meeting Notes: 10/4

Getting Started

Q: How do you run a Django project from the terminal?

Q: Are there any existing tracking devices that have been implemented? Other hardware components available to us?

Q: Who is the primary user for this app? What is the role of dispatcher? How do we contact the ambulance driver?

Notes:

-prioritize the ambulances right now
-create smaller roles within team after gaining better understanding of existing projects and files we have
-talk to ambulance mobile app team, is our framework scalable across different devices?

HW:
-go over documents
-familiarize ourselves with the tools

Meeting times in the future:

-Sunday, Tuesday, Friday (mb)

Meeting Notes: 10/11

Check-ins:

- Were you able to get the project setup and working on your machine?
- Were you able to read the documentation on our tools?

Functional Components

- Databases
  - Users
  - Ambulances
- Mobile Component
  - Login
  - Choose an ambulance you’re tracking for
  - Update status as you do yo thang
- Website Component
  - Display map with real-time location of ambulances (phones of folks in ambulances)

Updates:

- New mobile component required
- Everyone has the project locally
- Setting up environment
  - Issues with postgres/running “python manage.py migrate”
- Established timeline
  - Mobile component mockups/Environments ready (10/15)
  - Working leaflet/GeoDjango supported map modal on a webpage (10/18)
  - Track location of individual phones (10/25)
  - Implement leaflet-realtime functionality (11/1)
  - Accounts/State information displayed on webpage (11/15)
  - Styling/Troubleshooting/QA (11/29)
  - Galvanization of ambulance data (12/6)
- Established Schema
  - Database entries for users and ambulances
  - Users
    - Name
    - Password
    - Email
    - Status
  - Ambulances
    - License plate number
    - Status
    - Location
    - Current user bound to
    - Last location dispatched to*

- Make
- Color
- Number of trips

Action Items (Due 10/19)

- Create webpage with Leaflet functionality
    - Map displayed
    - Map shows user’s current position
- Environments fully set up
- Explore leaflet-realtime (https://github.com/perliedman/leaflet-realtime)

- Testing database
    - Discretely update location (every x seconds)
- Considerations
    - Limited connectivity (who has data lol)

Meeting Notes: 10/18

- Progress since last meeting:
    - Environments are now set up on everyone’s personal computer
    - Basic Leaflet map now set up on HTML file by Khalid

- Action Items due 10/19:
- Create webpage with Leaflet functionality
    - Map displayed - DONE
    - Map shows user’s current position - TODO
- Environments fully set up - DONE
- Explore leaflet-realtime (https://github.com/perliedman/leaflet-realtime)

- Map Details
    - Mapbox account under Khalid’s name
    - Need to make a Git Branch so we can start pulling/pushing to same project
    - Map as background div with menu overlay vs map taking ¾ of page + ½ menus
    - Mapbox GUI details (tons of options)

- TODO:
    - Find out how to check out branch by team?
    - Powerpoint Slides due tomorrow
    - Show user’s current position
    - Find out where AED database is, create ambulance table
    - Roles:
        - Work on Database (add table, define schema, add ambulances) - Khalid
        - Adding notifications and items on the map, (Leaflet, JS files) - Oliver
        - Overlay Menu (HTML)

Meeting Notes: 10/25

Meeting Notes

-We figured out how to create tables in the database and we are wondering how to populate it

-Manually updating data.json or if it is generated during runtime (apparently the tables will survive after closing the project, you probably only use data.json if you've deliberately emptied out your tables or something and wanna repopulate again)

-Ask Mauricio how to use leaflet widget.js and if he made it himself or if he got it from the Internet

In-class:

- Django library supports creating users already
- Populate databases with "python manage.py load data <<filename>>"
- Forms → views → models → templates
- Data.json is dump from database
- Prof says DON'T USE JSON (just transfer raw data?)
- ^^yea lol what then what's that getJSON() method all about? Not sure. Maybe he did it that way and he wants us to do it differently?
- **TODO: Email Gabri notebook stuff/ Post goal on FB by tomorrow**

Aye the notes look nice and full now :)

Meeting Notes: 11/1

How we do the talking:

1) Provide a URL for information to be posted to (http://www.wecantaffordadomain.com/receive) or something
2) Let mobile devices craft POST requests (using some httprequest library android has) and send the data to the location from (1)
3) We use AJAX (that's u Oliver ayyyy) to parse the data on the page and turn it into Postgres info
4) Update the relevant postgres tables (some javascript running on (1) or something I dunno)

Template files (HTML):

- Learned about the templates feature in Django
- Starting with base.html to form the base HTML page for the ambulance site
- Using base.html from AED site as a starting point (will change styling of page i.e. have map as full background with menu items overlayed)
- Need to put all Leaflet Javascript in its own file (LeafletWidget.js in AED site)

Auto-Update Table Data (models/forms)

- You gotta know what django models and django forms are
- Added new URL destination "update_ambulance/(?P<pk>[0-9]+)"
- Page takes you to a form where you can enter new status for ambulance
- URL number is ambulance license plate, the primary key of the Ambulances table
- A phone will send a POST request and structure its URL based on license plate of ambulance it is reporting for
    - Ex. "http://localhost:8000/craed/update_ambulance/<license_plate>"
- As soon as paragraph element is updated, post is sent to database

Meeting Notes: 11/8

Deadlines:

1. Notebook: (12/4)
2. Continuity report: next sunday (11/27)
3. Final Presentation: (11/30)

Notes:

- Tell us what the project is, from scratch.
- Presentation isn't on a sub-team basis, one overall presentation (you can probably still split by subteam)
- Presentation is minimum 10 minutes
- Inventory is literally everything people can look at, touch, read
- Reflection includes how well we did as a team? Why? What can we do different.

Contact Information

Ambulance Web App Subteam Members:

Khalid: kalmande@ucsd.edu
Ashley: a9zhao@ucsd.edu
Shaheryar: sajmal@ucsd.edu

Team Leader:
Kelvin: kssarabi@ucsd.edu, number: 619 362 0136