STEPHEN KENT

BACKEND WEB DEVELOPER

Github
stevejameskent1@gmail.com
(240) 441-4057

Profile

Backend web developer with experience using Python and Javascript (Node.js) with MongoDB in an agile environment.

Education

University of Maryland, College Park
B.S. Computer Engineering

Skills

JavaScript	Node.js	Express
Python	Java	Git
Github	Mapbox	MongoDB

Experience

CFPB

Backend Dev (Contract)

HMDA Pilot **Nov. 2014 - July 2015**

Helped develop a next-gen linting tool for HMDA (Home Mortgage Disclosure Act) filers. The tool allows users to submit a proposed HMDA file and have it checked for common errors (defined here). Issues with the file are summarized in the UI grouped by edit id with field names and their submitted values. The user can then edit the file to fix issues and continue to test until all errors are resolved before going through the final submission process.

Responsibilities

- Helped create a business rule engine which uses flexible JSON specifications for easy versioning of rulesets by year.
- Rule engine written as a node.js module and used in front end code via Browserify.
- Expanded the engine using mixins for versioning of engine logic by year in addition to ruleset specs.
- Wrote unit tests for all code I developed using mocha.js and must.js.
- Helped develop a RESTful lookup API for querying against public data (Census, etc) using kraken.js.
- Used Github issues to track stories and coordinate tasks.
- Git pull requests used to coordinate between team members and review code and relevant unit tests being integrated.
- Persistence layer built using Mongoose (MongoDB ORM).
- Database of choice was MongoDB for performance, easy scaling, and convenient storage format (JSON).

Environment

Node.js 0.10, kraken.js, express, mongoose, bluebird, lodash, superagent, moment, coveralls, istanbul, mocha.js, must.js, grunt, angular, MongoDB 2.6, Sublime Text 3.

FCC

Backend Dev (Contract)

Vizmo **Mar. 2014 - Nov. 2014**

Developed backend Python code for aggregating and displaying cellular internet performance testing results. Results were taken from the official FCC mobile speedtest app, binned into geographic areas (defined by various resolutions of hexagons), and then tabulated into aggregate statistics. These statistics were then used to generate a map using Tilemill and hosted by Mapbox. In the UI users could then click on a hexagon to see the statistics for that area, including the min, max, median, and average for download speed, upload speed, latency, and packet loss for each carrier.

Responsibilities

- Created backend Python code for aggregating cellular internet performance testing results.
- Results came in JSON format and were geo binned to hexagons to be displayed as aggregates.
- Python script ran and produced daily statistics using custom map-reduce code.
- Aggregation code was run in parallel using a worker-producer model via the Python multiprocessing module.
- Created maps in Tilemill to display the results of each run.
- Also wrote a second Python script to run after the first to automatically create updated maps and push to the Mapbox servers.
- Both scripts set to run and produce updated statistics and maps daily.
- Database of choice was MongoDB for performance, easy scaling, and convenient storage format (JSON)

Environment

Python 3, Curl, Tilemill, Mapbox, MongoDB 2.4, Pycharm / Sublime Text 3, node.js, restify, forever, mongojs.