JS Blog

Design
Katie Broida
12/6/15

For my final project I created a blog app using full-stack JavaScript. What I've created is the admin view of the blog app where you can create, view, update and delete posts. There isn't a users account functionality added yet, so this project is meant to run locally and demonstrate what the administrative features of this blog app. Learning an entirely new stack was challenging, especially when connecting the different components together. Below I've walked through the more interesting technical elements.

The Stack
This app is built using Node.js, Express, Angular and Postgres. Below are the contributions each part of this stack brought to the project:
- Node creates and runs the server
- Express is framework that allows the app to easily be organized into MVC and create an API for accessing the database
- Angular provides more powerful ways to display data in the HTML and client-side routing
- Postgres is the database I used. I mostly stuck with using SQL commands with Postgres, but is has some database specific syntax as well. Using parameters in my query allowed me to provide some protection against SQL injection attacks ([read more here](#)).

Client vs. server-side routes

At first I didn't understand the need for server-side and client-side routes, but once I did I could see some of the power and appeal of using JavaScript on the front and back ends. In the routes folder I have two separate files for the front (client.js) and the back end (server.js) routing. Server.js creates and API that the front end can then query. Client.js allows the app to use different templates on the front end. In public/javascripts/app.js I have create the routes that the user sees using Angular's routing functionality.

Views with Jade

Learning the Jade templating engine was tricky at first, but once I got the hang of it I found its sparse syntax easy to write and quick to parse visually. The views/layout.jade template is used with each view. In layout, there is a div for "ng-view", which allows the partial template to be injected. In views/partials, I have all of the Jade templates that get called depending on

the URL accessed in the browser. This relationship is set in the Angular routes in public/javascripts/app.js.

Future plans

Learning the JavaScript stack was tough at first, but now that I have seen some of its capabilities I am excited to build this project out further. Below are some features I have planned for my JS Blog:

- Add the ability to create user accounts and hide the admin pages from non-logged in users
- Add allow posts to have tags
- Create a special kind of tag called a "series," that would allow several posts to be organized together thematically. Each series could then have it's own page with a description of the series, and some series could be highlighted and promoted on the front page.

Final project walk through
https://youtu.be/3vMctRRnUcI