Name: Casey Boettcher
ONID: boettchc
Date: 2018-11-18
Assignment: week 8 activity

1. Make sure you can create sessions, add and update different kinds of data to sessions and end sessions.

   I had difficulty understanding how to set a cookie initially, as this seems to be glossed over in the lessons regarding sessions. Then I read some of the `express-session` documentation and realized that it's handled automatically, for the most part. But I still had to let go of the idea of the cookie as a store of data and accept that it's nothing more than a key into the persistent store on the server (within the limited context of this class and our use of `express-session`).

2. Make sure you can make HTTP requests, including POSTs, from the server and deal with the data you get back.

   It was initially puzzling to deal with three APIs that have some notion of an HTTP request (`node`, `express`, and the uniquely-named `request`). But I soon realized that we were creating yet another dependency to more easily allow for server-side calls to web-based APIs. After looking at the odd way that `node` expects a request to be created and then handled (create the request, but don't give it a callback, then call `fetch`...) I was grateful that Wolford had introduced the library.

3. Make sure that you can set up pages to accept data from forms.

   Not really a problem after the initial hurdle of finding and configuring `body-parser`.

4. Make sure you can handle nested asynchronous requests and process errors if they crop up (force errors to happen by doing things like entering in the wrong URL).

   Again, not an issue given that we were provided the necessary boilerplate and were instructed to call `next` from within the various route handlers.

What is most frustrating about the `express-handlebars` package is that things you'd expect to be available to you at view render time are not, or they're nested inside an object or an array. The documentation is not exactly lucid when it comes to distinguishing between app.locals, res.locals, objects that are passed in as a context, and the various intrinsic contexts that arise as a result of using conditional or iterative logic in a view template. String interpolation shouldn't be this difficult and time-consuming.