The statdns service is ferschluggener, so I adopted the tack suggested in the challenge outline, FreeGeoIP. But when I try to make a jQuery.AJAX call to the service, it succeeds, returning data and jqXHR objects with all of the properties one would expect. BUT, all of the properties are undefined.

However, if you enter a domain name (or an IP address) into a browser thus: - <http://freegeoip.net/json/altenbernd.com> - it works, showing in the UI something like this:  
{"ip":"64.7.69.242","country\_code":"US","country\_name":"United States","region\_code":"CA","region\_name":"California","city":"Novato","zip\_code":"94947","time\_zone":"America/Los\_Angeles","latitude":38.1032,"longitude":-122.6299,"metro\_code":807}

So why am I getting undefined for all the properties? Dinnoe. Tried changing the $.getJSON() call to simply $.get(), thinking that perhaps getJSON() might be somehow in conflict with the /json/ portion of the URL. But no; the results were exactly the aame.

But this is just trying to solve an existing bug and is not getting me closer to meeting the challenge. So I will push ahead with saving unique searches locally and building tests for them, then perhaps come back and try to resolve this problem. But it’s important that I now have an example of the object he FreeGeoIP is returning.

So at 3:30 I have solved this problem shile pushing ahead. The problem is that I was trying this:

For (var prop in data) {

console.log(console.log('\t' + prop + ': ' + data.prop);

}

And all of the properties were coming back as undefined. Of course what I should have said was

console.log(console.log('\t' + prop + ': ' + data.prop);

Said this way, it works just fine.

Sunday, May 07, 2017

From my experimenting in the TestBad, it seems as though having multiple calls to different services is unnecessary. A call to ip-api.com/json/ is sufficient and can be made with an IP address, a domain name, or no argument, and it always returns the desired info. But I am not going to try to refactor the entire application. Well, probably not.

Now it’s on to trying to capture and store new searches. Let’s take half-an-hour to see if an AngularJS service or directive is feasible.

Monday, May 08, 2017

All day on **Avenue Code** challenge.

At about 10:00, **Sam McCosker** sent e-mail asking for how I was doing. I signaled back that I have made good progress but want to take another day or so to make sure I do it right. He replied that he was cool with tat but stressed that they are moving quickly and wanted to verify that I will be ready to start within I week. I responded that I was.

Got the visited links list working fairly well by EOD.

Tuesday, May 09, 2017

Here is where I am:

* Visit history now the first thing in UI, above the Locate Yourself button
* When the Show History button is clicked, the Sites Visited list appears, and the rest of the UI moves down to accommodate the list
* When the button is clicked again, the list disappears, and the rest of UI moves back up
* Sites are listed in the order visited
* A given site is listed but once, based on IP address, no matter how many times it’s been visited
* While an IP address is unique within the list, a given domain name can appear multiple times (e.g. I had 5 items for Amazon, each with a different IP address)
* When the list is visible, I try to make it invisible again by setting the ctrl.showHistory variable, which is the subject of an ng-show directive on the list, to false, but that doesn’t work. The variable works fine for toggling list visibility when the Show History button is clicked but not when the value is set programmatically within the addSite() function.
* When the list is visible and a new site is added to the list, the Show History button does not toggle appearance of list but rather causes the content of the list to be updated to show the newly added site
* When you click the Revisit button next to an entry in the history list, data for that site is displayed in the Estimated Location table and the map
* I have corrected the problem with clicking the help icon in the Estimated Location table. Previously clicking the button caused a message to pop up for the corresponding item (e.g. IP) for all sites that have been visited during the session, requiring a separate click to discharge each of the messages. Now a message for only the currently displayed item appears.
* Similar to the immediately previous bullet, pins in the map accumulate for all sites that have been visited during the session. It probably makes more sense to display a single pin for the currently displayed site. There probably is a GoogleMaps API call to remove pins, but I haven’t had a chance to look into it yet.
* I have no prior experience with Jasmine and am climbing its learning curve. Thus I do not yet have any testing in place, although I am working on it.

Sent the preceding list of bullets to Sam, asking if he would like me to check in my work so that they could get started with the evaluation. He responded that he would like me to do that.   
Which I did, and I placed the bullet points above, as well as several others, into a README.md file. Sam responded to acknowledge that his evaluators have access to my repository and that he will respond ASAP with feedback. Ooooh! The anxiety!

Wednesday, May 10, 2017

Will take an hour or two to work on a couple of bugs that I have identified. Then it will be on to unit testing with Jasmine.

Thursday, May 11, 2017

Beginning at about 6:30, working on Jasmine testing.

HA! By about 1:00, I had a successful Jasmine unit test of the controller and it’s addSite() function. Pushed it up to the Bitbucket site and sent e-mail to **Sam McCosker** asking if there were more I should be doing. HE responded that we are now just waiting for the evaluators to do their thing and he will let me know the outcome. Ooooh, the anxiety!

Friday, May 12, 2017

AHA! As I got out of bed at 6:15, I had a sudden insight. What if, instead of saying showHistory = false in the addSite() function (which does not hide the Visited Sites list), what if I just fired btnShowHistory.click()? I tried it, and it worked: when a new site is added to the list, the list disappears. When the Show History button is clicked, the list reappears with the newly added site included. I Gitted it and uploaded it to Bitbucket, sending **Sam McCosker** e-mail to inform him.

Sam responded that he had spoken with the evaluator who looked over my code, and the reaction was generally positive. Then at 10:00, Sam called and said that the evaluator had nothing but good things to say about my code. He set up a technical phone screen with **Mark Carlson** at 2:45 on Monday. It will be mostly a conversation about the coding challenge and how I went about solving it, and also to make sure that it was I who did it. Pending success with that call, they will bring me in for an on-site. The anxiety has diminished somewhat.

Monday, May 15, 2017

Temporizing as I build up to the phone call this afternoon. Actually, I am fairly confidant and am looking forward to it. Modified the code challenge app so that if the history list is visible when a new site is added to the list, it remains visible rather than disappearing and the newly added site pops up at the end of the list. Updated the README.md file and push the changes to it and avenueControllers.js to the Bitbucket site.