Unit Capstone: Instagram Searcher

Estimated Time: 5 hours

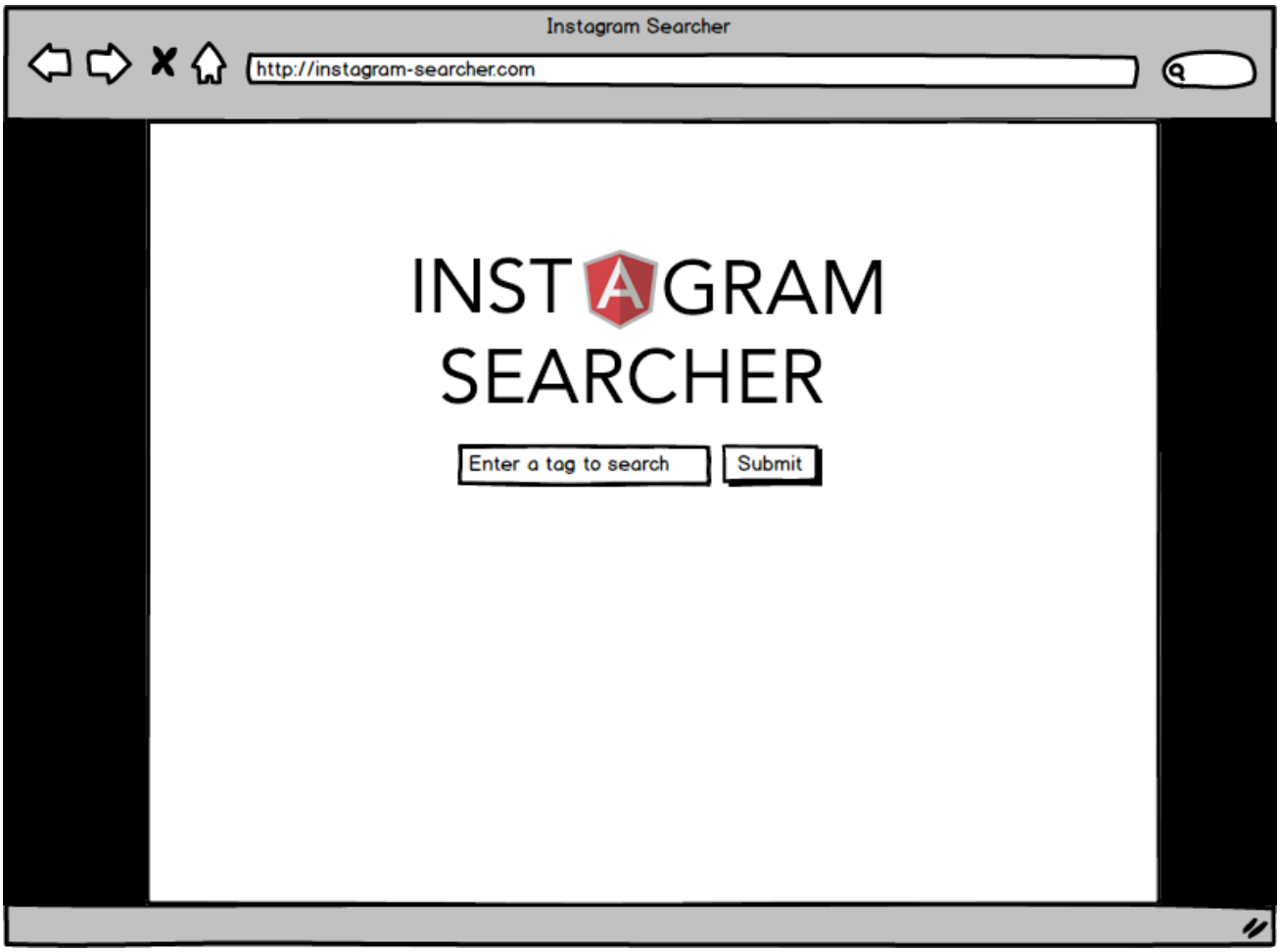
To close out this lesson and unit, you're going to build your most challenging app yet, a tool that allows users to search instagram for pictures.

We provide a wireframe and spec of the behaviors for the app, and it will be up to you to build it up from scratch. To complete this assignment, you'll draw on your knowledge of templates, modules, controllers, scope, XHR, and animation basics in Angular.

By mastering the basics that this project requires, you'll be ready to tackle the more complex and advanced topics in Unit 2.

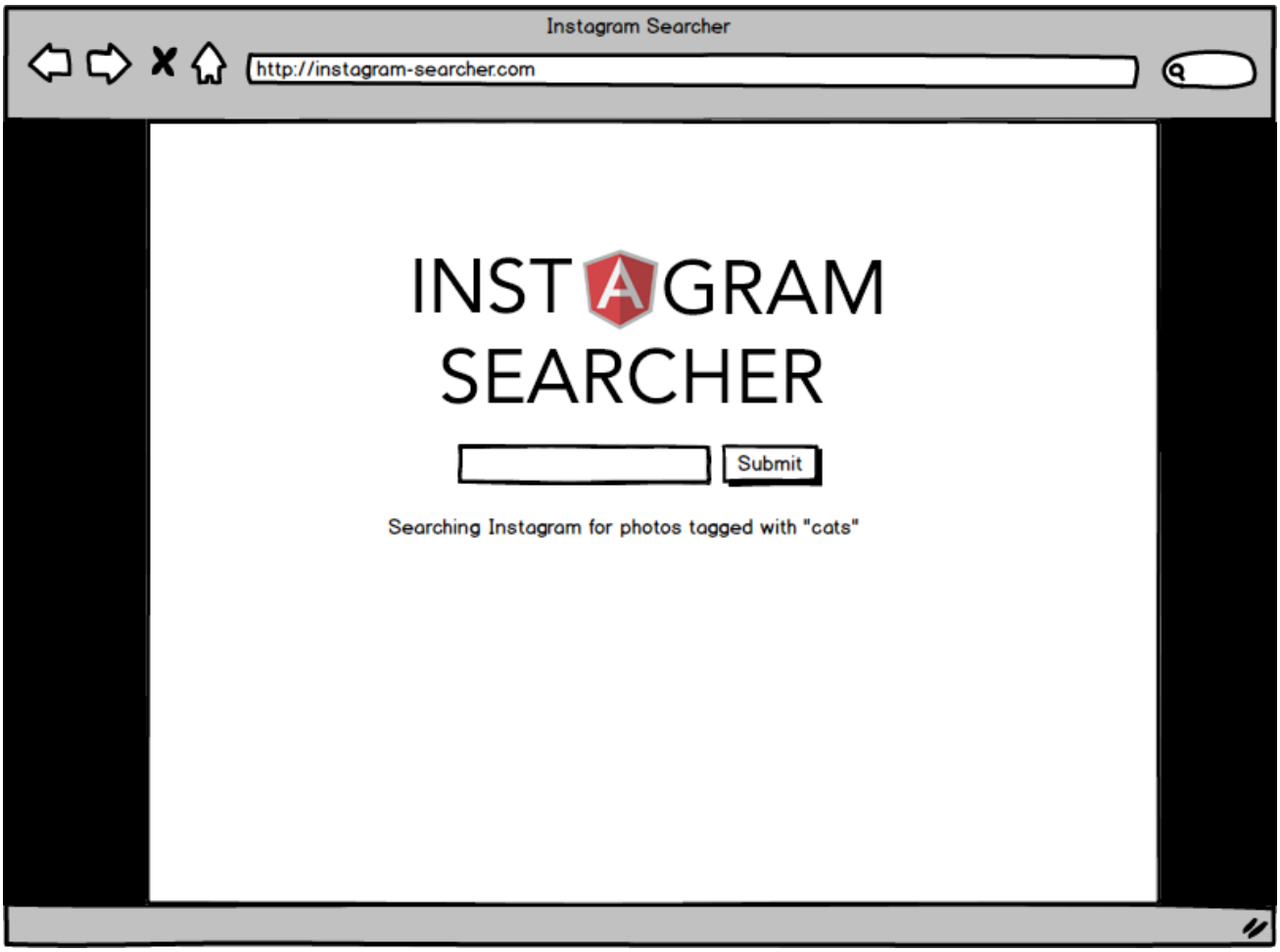
Sketch

Let's have a look at the sketch for our Instagram Searcher app. First, let's look at what the user will see when they first visit the page:



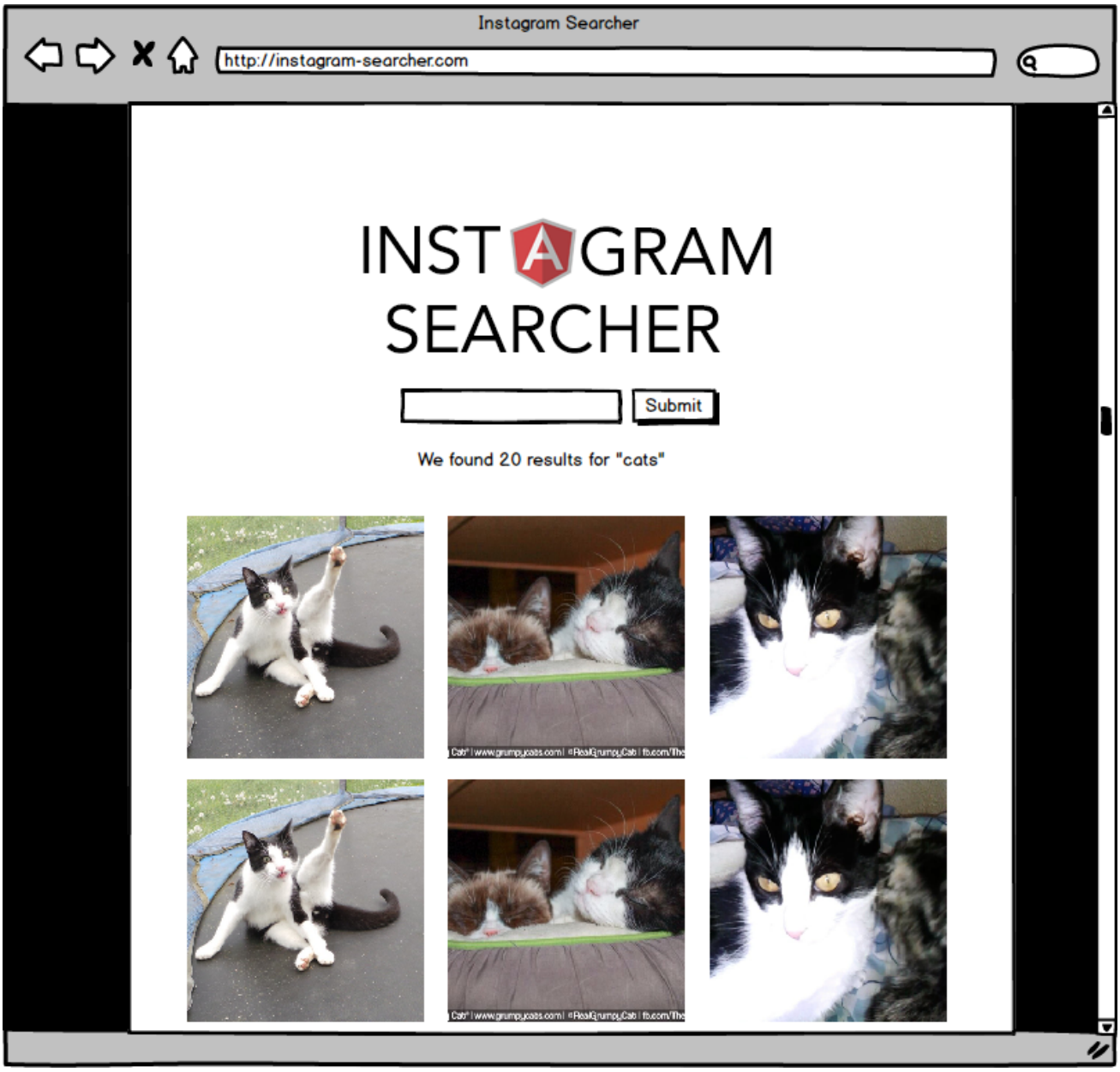
When the user visits the page, they see the logo and a single text input where they are prompted to enter search term. When the user clicks submit, the app first validates the form to ensure that a tag was supplied.

On valid form submission, two things happen: an asynchronous query is made to the Instagram API for recent images tagged with the search term, and the view in this sketch is rendered:



Two small details to note about this view. First, the text input has been cleared. Second, there's text telling us that Instagram is being queried for our search term.

Finally, assuming the query is successful, the app should take the URL parameter for each item returned in the data array and display an image on the page. This part of the app should resemble this sketch:



There should be some text that tells the user how many images were found, and the photographs themselves should be displayed. Each image should link to the actual instagram page where you can view it.

In the event that no images were found, the text should reflect this.

Finally, if there was an error calling the InstagramAPI, there should be a message indicating this to the user (this is not pictured in the sketches above.)

This gives you the overall story for the app. Here are the distinct requirements.

Requirements

Version Control: Use Git and Github from the beginning and commit frequently.

Design

Your app must implement the sketches and user story provided above.

You can design your own logo, or use the one from our sketch if you prefer available for download here.

You are free to embellish the UI. In the sketches we've kept the styling to a minimum. Feel free to prettify your app.

Behaviors

Form validates that user supplied text input

When user clicks submit, the text field in the form is cleared and text appears that indicates Instagram is being queried.

The app should query the Instagram API and search for recent pictures tagged with the term the user has supplied in the form.

When the API results return, the app should tell the user how many results were returned, and display the photos (if any).

You should be able to navigate to the original photo on Instagram by clicking on any one of the images that appear.

Your app **must use at least one animation.** We leave it up to you to decide where to put this.

In case your calls to the server fail for some reason, the app should somehow alert the user.

Users should be able to submit another search after one has completed.

Project completion - When you've completed, put the code your project:

Share a link to a live page (via Github Pages, CSSDeck, or

equivalent) to your app. Share it with both your mentor and your community.

Share a link to your Github code repo with your mentor. Be ready for code review at your next mentor meeting.

Working with the Instagram API

For this project, you'll need to get a client id from Instagram that you can associate with your app. You'll also have to know how to interact with one endpoint from the API. We'll point you in the right direction here

Getting an Instagram client_id

In order to interact with the InstagramAPI, your app will need to supply a*client id*. To get an Instagram client id, you need to be on Instagram.

If you're not already on Instagram, to sign up for an account, you need to download the app for your iOS or Android mobile device. If you are either unable or unwilling to do this, please let us know as soon as possible, and we can set you up with a client id that we create.

1. Log in to Instagram.com.
2. Go to the Developer Welcome Page
3. Click on "Register Your Application".
4. Click on "Register a New Client". Fill out the form. You can call the app whatever you want, but it cannot have "insta" in the name. Since we don't have a live website we'll be deploying to just yet, you can make something up for the *Website* and *OAuth redirect_uri* fields. After you hit submit, your new client will be created and you'll be taken back to the "Manage Clients" interface.
5. Note that there is a client id field for your app. You'll need to assign this client id as a query parameter when your searching the InstagramAPI later on.

Our Data Endpoint, Request, and Response

Instagram's API does not permit CORS, so your app will need to use JSONp to access the data.

Specifically, your app needs to query the following **endpoint**: `https://api.instagram.com/v1/tags/{tag}/media/recent` with these query parameters:

```
callback: 'JSON_CALLBACK'
client_id: 'your_client_id'
```

Remember that for callback, you must supply the exact string value 'JSON_CALLBACK'. However, client_id should be set to your app's unique client id, which you got from Instagram (see above).

The **response object** you get back from `$http` will have a *data* key. response.data will be an array of up to 20 data objects, each of which represents an Instagram photo. In order to display photos in your app, you'll need to parse the data objects returned by Instagram. You can see the data schema for a photo object in this Gist file, or you can code up a file that uses $http to make a call to Instagram, and throw a debugger in the success routine and inspect the response object.

If you head over to the Instagram Developer API Console, you can try out this and other endpoints available from Instagram. You'll need to select our endpoint from the "Select an API method" interface. In addition to supplying a tag to search for in the URL, you'll also need to supply the key/value pair 'client_id': 'my_client_id' (where 'my_client_id' is the ID you got from Instagram). You can input these parameters in the "Custom name/value" part of the query parameter interface.

Additional Hints

ng-src - In order to display the images in your app, you'll need to retrieve image urls from the response data, and you'll need to set the `ng-src` attribute in your image elements to these image url values. Take a moment and read the official docs on ng-src to learn why you need to

do this, or better yet, try running the app with your image element using `src` and you'll see why.

$http.jsonp vs. $http() - While it's possible to use the jsonp convenience method, this means you have to do the url encoding of your query parameters yourself, which you should avoid if possible. You're therefore encouraged to use $http() called with a config object where you can set the *params* parameter. You can read up again on $http() config objects here in the official docs.