

Module 5 Coding Assignment

Click [here](#) to SIGN UP for the *Single Page Web Applications with AngularJS* course on Coursera. It's FREE!

It's coding time! Woohoo!!!

Time to Complete

It should take about an 1 hours or less. (This is just an estimate. People's backgrounds differ, so for some people it will take longer to complete.)

Ask questions in [Week 5 Discussion Forum](#) if you get stuck! We are all learning, and going through getting stuck and then unstuck (even with someone's help) can be a very valuable learning experience!

Assignment Instructions

General Idea

Your starting point for this assignment will be the code in the `examples/Lecture59` folder.

Your task is fairly simple.

First, change the 2 buttons at the top of the website to say `My Info` and `Sign Up` instead of `About` and `Awards`.

Task 1: When the user clicks on the `sign up` button, they should be taken to a view that lets them "sign up" for the newsletter (that will never come :-)) of the restaurant. The sign up form should ask for the following information: first name, last name, email address, and phone number. It should also ask the user to specify the menu number that's their favorite dish. (The menu number is the `short_name` attribute of each menu item).

Everything in the form (except the `short_name`) must be validated through AngularJS validation we learned in this module. When the user clicks the `submit` button (after all other fields are validated), try to retrieve the menu item the user specified as their favorite through the following REST endpoint https://YOUR-CHOSEN-SUBDOMAIN.herokuapp.com/menu_items/SHORT-NAME.json, where `YOUR-CHOSEN-SUBDOMAIN` is the subdomain discussed in Lecture50 (Restaurant Server Setup), and the `SHORT-NAME` is the name the user provided. If the server returns an error, you will know that the item name specified is not valid. If that's the case, display a message next to the favorite choice saying `No such menu number exists`. (See bonus for more)

If everything is valid, save the user's preference somewhere in your client app such that you can retrieve it in another view/component/controller/etc. (Hint: think service). Once saved, display a

message below the form saying `Your information has been saved.`

Task 2: When the user clicks on `My Info` button, they should be taken to another view where it shows them their "registered" information, including the favorite menu item they picked. When showing the favorite menu item, display the picture of the menu item as well as its title and description.

If the user hasn't yet "registered" through the `Sign up` button, simply display a message saying: `Not Signed Up Yet. Sign up Now!`. The words "Sign up Now!" should be a link to the same view as the `Sign up` link points to.

› Bonus (not graded)

Bonus Task 3: We haven't covered this, so do some research and figure out how to setup the validation of the user's choice for the favorite item BEFORE the user hits the `submit` button. This way, the message `No such menu number exists` should show up pretty soon after the user types something into the favorite menu item textbox and it loses focus.

(Yes, I realize, it may be better to simply present the user with the list of items, but this is an exercise, after all.)

Bonus Task 4: Write a simple test which exercises the function that will determine if the favorite item exists in the menu or doesn't exist. Note, that you will need to use the `$httpBackend` service (and probably look up the docs for it as well) to achieve this test properly.

› Rules

Do it the Angular way... We've covered this enough in the previous assignments and having the restaurant codebase the way it is, it would simply be more "painful" for you to try to do this without AngularJS, so I trust that you will pretty much be forced to do things the Angular way.

› Steps

I will not detail the steps in this assignment simply because pretty much all of them have been listed in the previous assignments. At this point, you have done the tasks I am asking you to do in this assignment in the previous assignments (with the exclusion of form validation, which you should just look up the lecture and sample code for.)

› Important Implementation Notes

1. Make sure all of your Javascript code is inside of an IIFE. (*If you don't know what that is or why we'd want to use it, brush up on it by looking through module 4 of [HTML, CSS, and Javascript for Web Developers](#) course I teach.*)
2. Make sure all of your dependency injections are protected from minification.
3. After you are done and satisfied with your solution, don't forget to add/commit/push your code to your repository.

⁹ IMPORTANT REMINDERS:

- Closely follow the submission guidelines for this assignment on Coursera.org
- Make **sure** you provide the correct URL in your submission (it should be GitHub.**io**, *not* GitHub.**com**)
- Make **sure** to TEST your assignment not just on your local machine, but ALSO once you deploy it on GitHub, using the URL you are providing as part of your submission.
- This assignment will be peer-reviewed (and graded). The guidance will be given such that if submission instructions are not followed, the assignment is to be failed. This includes providing the wrong URL for your deployment. Following instructions is very much part of software development. After all, that's what software requirements are - instructions to follow.