**COMP 3123 – Full Stack Development – Lab 11**

* Angular Services
* Consuming HTTP Services & Observables

**Developer Note:**

Only submit the files components, typescript files that you have created and not the entire application.

**Homework:** do the Angular Service Tutorial

<https://angular.io/tutorial/toh-pt4>

**Exercise #1 – HTTP GET Requests**

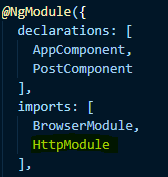
1. Go to <http://jsonplaceholder.typicode.com> and scroll to Resources to view the various fake API service end points. Click the POST and inspect the JSON list of static data.
2. Use the following ***Angular CLI*** command to generate a ***component named post***



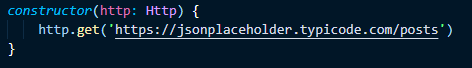
1. Navigate to the ***app.module.ts*** file and import the ***HTTP client***.



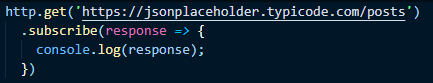
Add the ***HTTPModule*** to the list of imports, if it is not already there.



1. In our ***PostComponent*** inject the ***HTTP*** service into our constructor and add import statement for HTTP. Then use the http to do a get request to the typicode.com post endpoint.



1. The Http GET request returns an observable. We need to subscribe to it, so that we can get access to the response and console log it. Implement the following and view in the browser debugger to see the response.



1. Convert the response object to json object to be console logged. View the result in browser (Chrome Dev tools) debugger to see response.
2. Add a variable named posts of type array any.



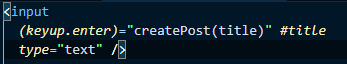
1. In the ***subscription*** of the ***Http GET response*** set the response json object to the array of posts.



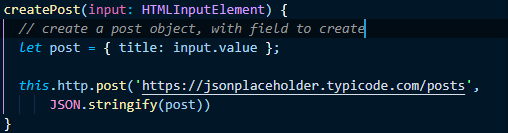
1. Get the HTML ***template post.componet.html*** and render this posts list in a <ul> and <li> tags.

**Exercise #2 – HTTP CREATE Requests**

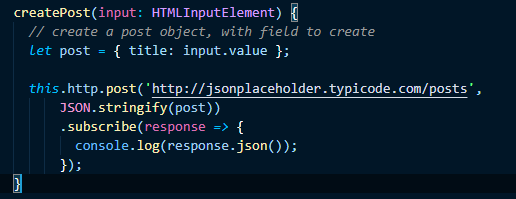
1. Add the following input box in our HTML markup to hand the enter key and call a method in the ***Post Component via Event Binding***. We are setting a template variable for title and passing it to the event method.



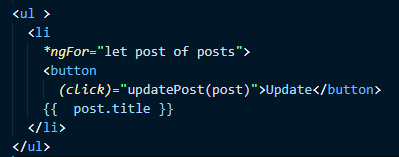
1. Implement the new ***createPost*** method to the ***post.component.ts*** file in the following way to handle creating a new Post object.



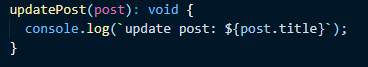
1. ***Subscribe*** to the ***Post Observable*** and console log the result. The finished method should be as follows.

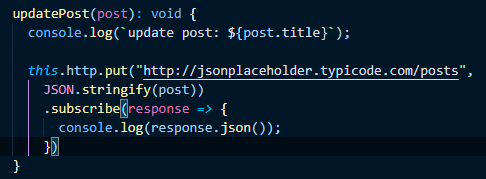


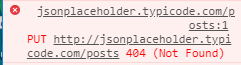
1. When the ***Post Observable*** responds, push the new Post item to the top of the post array. The UI will add to the top of the list after enter action.



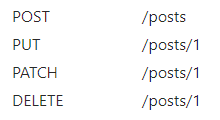
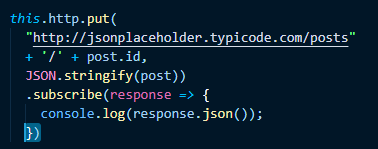
2. In the ***post.component.ts*** add the ***updatePost*** method and console log the result. Test the action in the chrome browser tools



3. Add a HTTP Put request, subscribe to the Observable response and console log the result.  
  
  
4. Triggering the ***PUT request*** will cause the following error in the browser console.



As per the typicode site example the PUT method requires a ID in the request.

  
5. Add the ID to the URL and then test the result  
.   
  
6. Test the ***PUT request*** and review the messages in the browser console log.

**Exercise #4 – HTTP DELETE Requests**

1. Implement a Delete button in the UI, that call a method deletePost. This method will take post id and make an HTTP Delete request to the mock service. Handle the response and log to console.

**Exercise #5 – Extract to Service**

1. ***Use Angular CLI*** to create a new service named ***post.service***



1. ***Extract*** the HTTP post service related code to the ***post.service.ts,*** inject and call the methods in ***post.component.ts***