# *Web Programming VI (420-H60-HR)*

# *Lab 10 – SPA, React (consuming ReST)*

Date due: **Monday, October 21, 16h00**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will have:

* Created a React application that consumes ReST with step by step assistance
* Analyzed the React application to understand the basics of how it runs
* Applied the knowledge gained to extend the project to support other ReST methods.

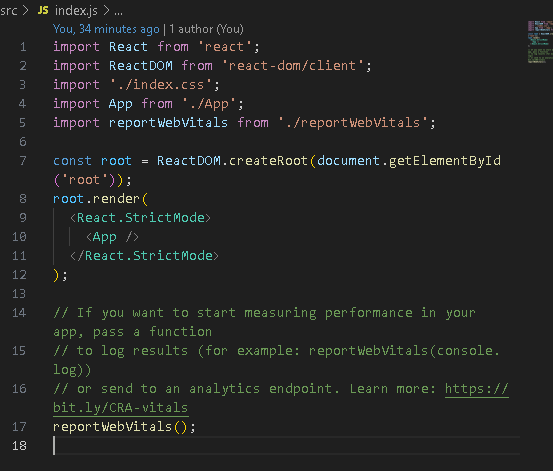
Setup

1. Create a folder to use for the lab called *username\_H60L10*
2. Move your React “Consume Rest API” walk through project to the *username\_H60L10*
3. Make sure everything still runs.

Part A: Analysis – How does this thing work

Follow the steps provided in the walk through slides.

1. In order for page information (HTML) to be generated, a React component must call render(). Where is this called?



1. Explain this code snippet resolves to output. i.e. how does <http://localhost:3000> turn into the output you get?

When going to localhost:3000, the tool create-react-app serves the app. Index.js renders the app component into the root div. In App.js REST API calls are handled, fetching data from backend server and updating the component with that data.

1. Where is the code snippet that retrieves the list of users and displays it? Explain the key lines.

Setup:

  
fetching data:  


Rendering:  


1. What is the purpose of the useState()? How is it used? Provide a URL with documentation on the useState(). Make sure you get the gist of it.

<https://react.dev/reference/react/useState>  
It is used to add state management to components. It allows you to declare state variables within a component and provides a way to update that state. When you call useState() you pass the initial state as an arg, an it returns an array with two elements: the current state and a function to update the state. When the state is updated, React re-renders the component to reflect the new state in the UI.

1. Go into browser debugging and capture the sequence of network messages to generate the user detail. Is the server that serves up the React application interact at all with the ReST api where user data is retrieved? Show screenshot and explanation and proof:

The React app server does not interact with the REST API, the browser does. The server hosting the React app only serves the React assets (html, js, css). The network request happens after the page loads, and is trigger by the fetch() call in the useEffect() hook. API requests are handled independently by the browser through JS, w/o the React server acting as an intermediary.



Part B: Build your own

Resources:

<https://jsonplaceholder.typicode.com/guide/> - ReST JS fetch samples

<https://www.techgeeknext.com/react/react-crud-example> - CRUD samples uses axios rather than fetch, but good concepts to steal

1. Go back to Topic 8 where we built our first ReST service. (Remember RestEasy and Pizza dude?). Copy that project into your *username\_H60L10* folder. Run it to make sure it works. Test the swagger pages or via PostMan to refresh your knowledge of the service and the data formats. Fill in the following table:

|  |  |  |
| --- | --- | --- |
|  | HTTP verb and URL | Body format |
| Create |  |  |
| Retrieve (one) |  |  |
| Retrieve (many) |  |  |
| Update |  |  |
| Delete |  |  |

1. Build a React Application to do full CRUD on your Pizza service. You should have all the functionality you filled in the table (above).
2. Significant marks for making it look nice/professional. Brand it and make it look professional.

Marks

|  |  |
| --- | --- |
| **Web Programming VI (420-H60-HR ) Marks Lab 10 (React SPA)** | **Out of** |
|  |  |
| **Part A - Walk through and Analysis** |  |
| render() | 2 |
| code path and explanation | 4 |
| fetch and list generation | 4 |
| useState() | 4 |
| client side packet capture | 2 |
|  |  |
|  |  |
|  |  |
| **Part B - Pizza CRUD site** |  |
| Analysis (Table) | 5 |
| Client |  |
| GetMany | 2 |
| GetOne | 2 |
| Update | 2 |
| Create | 3 |
| Delete | 2 |
| Styling – make it look professional | 10 |
|  |  |
| Handed in properly | 3 |
| **Total** | 45 |

**To submit**

When you have completed the lab exercise, create a zip file of the folders and load it to the course page.

**Note: Before zipping up your React project, delete the node\_modules directory. It can be recovered by running “npm install” (This is equivalent to “Build Clean” in other frameworks).**