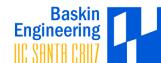


# CSE183

Lecture will start at 09:20



## Web Applications

# CSE183

Fall 2020

## Single Page Applications



## Notices

- Assignment 4 due 23:59 **Thursday October 29**
- Quiz 2 during class **Friday October 30**

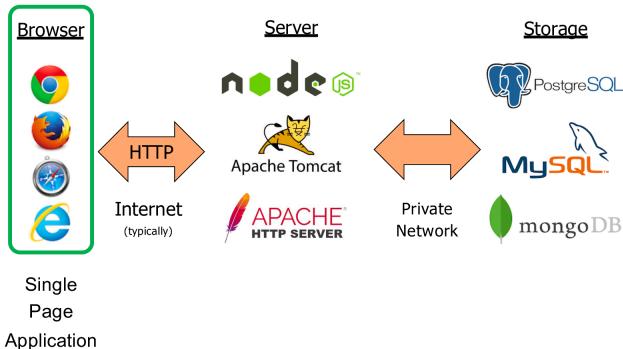
## Today's Lecture

- React Review
- Problems with Multipage Web Apps
- Deep Linking
- Single Page Applications (SPA)
- React Routers and SPA

- Assignment 4 - Secret Sauce
- Questions



## Full Stack Web Applications



5

## React Review

- JavaScript framework for web applications
- Uses the Model View Controller pattern
- Minimal server-side support required
- Intended for large and **single page applications**
- Simplifies reacting to changes of state
- Includes JSX Templating
- Handles Component Lifecycles

6

## Problems with Multipage Web Apps

- By definition, web applications run in browsers
- Users are familiar with browsers ☺
  - A history of URLs visited is maintained
    - Back & Forward buttons - navigate the URL history
  - Easily move to a new page
    - Type into location bar or use forward/back buttons
    - Selecting a bookmarked URL
    - Clicking the page refresh button
- Browser deletes the current JavaScript context when user moves to a different page
  - Necessary to preserve resources and for security, but...
  - Problematic for JavaScript frameworks ☹
    - URL (and cookies) are the only information preserved

UCSC 850E CSE153 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

7

## Problems with Multipage Web Apps

- Static Apps: HTML Pages served from web server
  - Each page had a URL and app switched between
- Early Dynamic Apps: JavaScript pages
  - Problem: Restart web app on navigation ( could lose all your work ☹ )
 

```
window.addEventListener('beforeunload', function (e) {
  e.preventDefault();
  e.returnValue = '';
});
```


- Users expect app to support well-known behaviors:
  - Navigate with forward & back buttons and page history
  - Navigate away and come back to the app
  - Bookmark a specific place in the app
  - Copy the URL from the location bar and share it
  - Click the page refresh button and application state is maintained

UCSC 850E CSE153 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

8

## MPA in HTML

### Folders & Files

```
help.html
tech.html
index.html
common.css
```

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

9

### index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML / JavaScript MPA</title>
  <link href="common.css" rel="stylesheet"/>
</head>
<body>
  <h1>CSE183 Fall 2020</h1>
  <ul class="navigation">
    <li><a href="index.html" class="active">Home</a></li>
    <li><a href="tech.html">Tech</a></li>
    <li><a href="help.html">Help</a></li>
  </ul>
  <h2>Welcome</h2>
  <p>Wise, you are, in taking this class.</p>
</body>
</html>
```

### CSE183 Fall 2020

- Home
- Tech
- Help

### Welcome

Wise, you are, in taking this class.

10

### help.html

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML / JavaScript MPA</title>
  <link href="common.css" rel="stylesheet"/>
</head>
<body>
  <h1>CSE183 Fall 2020</h1>
  <ul class="navigation">
    <li><a href="index.html">Home</a></li>
    <li><a href="tech.html">Tech</a></li>
    <li><a href="help.html" class="active">Help</a></li>
  </ul>
  <h2>Got Questions?</h2>
  <p>Easiest thing to do is mail me a letter.</p>
</body>
</html>
```

### CSE183 Fall 2020

- Home
- Tech
- Help

### Got Questions?

Easiest thing to do is mail me a letter.

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

11

### tech.html

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML / JavaScript MPA</title>
  <link href="common.css" rel="stylesheet"/>
</head>
<body>
  <h1>CSE183 Fall 2020</h1>
  <ul class="navigation">
    <li><a href="index.html">Home</a></li>
    <li><a href="tech.html" class="active">Tech</a></li>
    <li><a href="help.html">Help</a></li>
  </ul>
  <h2>NERP - It's a thing</h2>
  <ul>
    <li>Node.js</li>
    <li>Express</li>
    <li>React</li>
    <li>PostgreSQL</li>
  </ul>
</body>
</html>
```

### CSE183 Fall 2020

- Home
- Tech
- Help

### NERP - It's a thing

- Node.js
- Express
- React
- PostgreSQL

12

3

## Change URL Without Reloading Page

- URL hash fragment can change without reload
 

```
http://example.com
http://example.com#fragment
http://example.com?id=1234
http://example.com?id=1234#fragment
```
- HTML5 Browser History API via `window.history`
  - Gives JavaScript control of page reload
  - Allows change of URL without reloading page

UCSC 850E CSE153 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

14

## Deep Linking

- General Idea:
  - URLs should capture a web app's context so that browsing to a particular URL persuades the app to switch to the relevant context
  - Supports:
    - Bookmarks (private-ish)
    - Sharing (public)
- App context is defined by the user interface designer:
  - E.g. Viewing a data entity with an edit dialog open
    - Should the link point to the data entity view or to the entity & dialog?
    - Does it matter if user bookmarking for personal use or sharing with others?
    - How about navigating away and back or browser refresh?

UCSC 850E CSE153 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

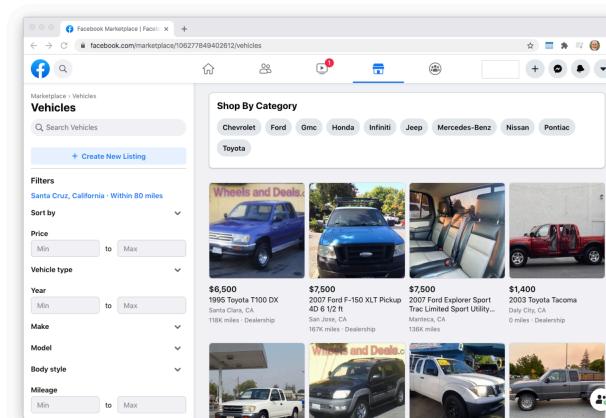
15

## Deep Linking in Single Page Apps

- Encapsulate the app's context in the URL
  - Works for browser navigation and refresh
  - User can copy URL from location bar
- Provide a share button to generate deeply linked URL
  - Allows user to explicitly fetch a URL based on need
  - Can keep URL in location bar pretty
- In both cases the web app should be able to initialize itself from a deeply linked URL

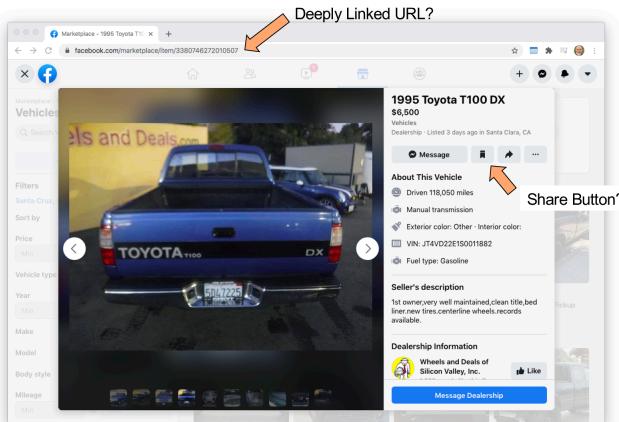
UCSC 850E CSE153 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

16



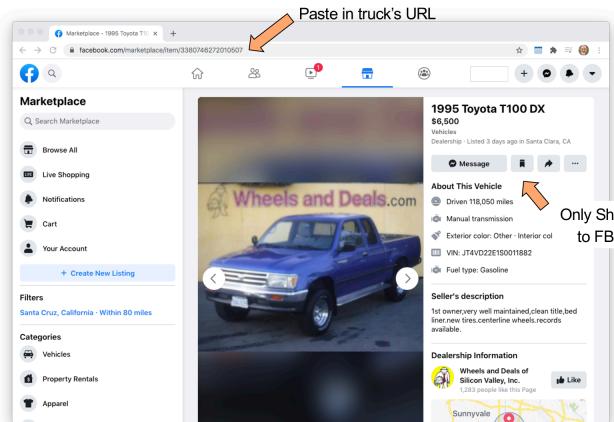
UCSC 850E CSE153 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

17



UCSC 890E CSE153 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

18



UCSC 890E CSE153 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

19

Perhaps not as deeply linked as we wanted, but deeply linked enough? Probably.

UCSC 890E CSE153 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

20

## URL Ugliness

- Compare:  
<http://www.example.org/dirmod?sid=789AB8&type=gen&mod=Core+Pages&gid=A6CD4967199>
- To:  
<http://www.example.org/show/A6CD4967199>
- And what the hey is this?  
<https://www.flickr.com/photos/dcharris/3578000165/in/explore-2020-10-228/>

UCSC 890E CSE153 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

21

## SPA in HTML / JavaScript

### Folders & Files

```
Help.js
Home.js
Tech.js
index.html
index.css
index.js
```

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

22

### Tech.js

```
class Tech {
  constructor(containerId) {
    const container = document.getElementById(containerId);
    const h2 = document.createElement('h2');
    h2.textContent = 'NERP - It's a thing';
    const ul = document.createElement('ul');
    for (const item of ['Node.js', 'Express', 'React', 'PostgreSQL']) {
      const li = document.createElement('li');
      li.textContent = item;
      ul.appendChild(li);
    }
    container.appendChild(h2);
    container.appendChild(ul);
  }
}
```

### NERP - It's a thing

- Node.js
- Express
- React
- PostgreSQL

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

24

### Home.js

```
class Home {
  constructor(containerId) {
    const container = document.getElementById(containerId);
    const h2 = document.createElement('h2');
    h2.textContent = 'Welcome';
    const p = document.createElement('p');
    p.textContent = 'Wise, you are, in taking this class.';
    container.appendChild(h2);
    container.appendChild(p);
  }
}
```

### Welcome

Wise, you are, in taking this class.

### Help.js

```
class Help {
  constructor(containerId) {
    const container = document.getElementById(containerId);
    const h2 = document.createElement('h2');
    h2.textContent = 'Got Questions?';
    const p = document.createElement('p');
    p.textContent = 'Easiest thing to do is mail me a letter.';
    container.appendChild(h2);
    container.appendChild(p);
  }
}
```

### Got Questions?

Easiest thing to do is mail me a letter.

### index.html

```
<!DOCTYPE html>
<html>
<head>
  <title>HTML / JavaScript SPA</title>
  <link href="index.css" rel="stylesheet"/>
  <script type="text/javascript" src="index.js"></script>
  <script type="text/javascript" src="Home.js"></script>
  <script type="text/javascript" src="Tech.js"></script>
  <script type="text/javascript" src="Help.js"></script>
</head>
<body>
  <h1>CSE183 Fall 2020</h1>
  <ul class="navigation">
    <li><a id="homeRef" href="#/home">Home</a></li>
    <li><a id="techRef" href="#/tech">Tech</a></li>
    <li><a id="helpRef" href="#/help">Help</a></li>
  </ul>
  <div>
    <div id="home"></div>
    <div id="tech"></div>
    <div id="help"></div>
  </div>
</body>
<script>
  new Home("home");
  new Tech("tech");
  new Help("help");
</script>
</html>
```

### CSE183 Fall 2020

- Home
- Tech
- Help

### Welcome

Wise, you are, in taking this class.

25

```

index.js

function hide(...ids) {
  for (const id of ids) {
    document.getElementById(id).style.display = 'none';
    document.getElementById(id+'Ref').className = '';
  }
}

function hideAll() {
  hide('home','tech','help');
}

function show(id) {
  hideAll();
  document.getElementById(id).style.display = 'inline';
  document.getElementById(id+'Ref').className = 'active';
}

function attachListeners() {
  document.getElementById('homeRef').addEventListener('click', function (e) { show('home'); });
  document.getElementById('techRef').addEventListener('click', function (e) { show('tech'); });
  document.getElementById('helpRef').addEventListener('click', function (e) { show('help'); });
}

window.addEventListener('DOMContentLoaded', function (e) {
  attachListeners();
  show(window.location.hash ? window.location.hash.slice(2) : 'home');
});

```

UCSC 890E CSE153 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

26

## React Support for SPA

- React remains firmly unopinionated
  - If you want SPA, do it yourself or use a 3<sup>rd</sup> party library
- **React Router** <https://reactrouter.com>
  - Uses URL to control **conditional rendering**
  - HashRouter : Fragment part of the URL
  - BrowserRouter : HTML5 URL Handler

UCSC 890E CSE153 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

28

## React Router Example

### Folders & Files

```

build
public
  index.html
src
  component
    Help.js
    Home.js
    Tech.js
  App.js
  index.js
  package.json
}
  New

```

UCSC 890E CSE153 Fall 2020. Copyright © 2020 David C. Harmon. All Rights Reserved.

29

### src/component/Home.js

```

class Home extends React.Component {
  render() {
    return (
      <div>
        <h2>Welcome</h2>
        <p>Wise, you are, in taking this class.</p>
      </div>
    );
  }
}

```

### Welcome

Wise, you are, in taking this class.

### src/component/Help.js

```

class Home extends React.Component {
  render() {
    return (
      <div>
        <h2>Got Questions?</h2>
        <p>Easiest thing to do is mail me a letter.</p>
      </div>
    );
  }
}

```

### Got Questions?

Easiest thing to do is mail me a letter.

7

```
src/component/Tech.js
class Tech extends React.Component {
  render() {
    return (
      <div>
        <h2>NERP - It's a thing</h2>
        <ul>
          <li>Node.js</li>
          <li>Express</li>
          <li>React</li>
          <li>PostgreSQL</li>
        </ul>
      </div>
    );
  }
}
```

**NERP - It's a thing**

- Node.js
- Express
- React
- PostgreSQL

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

31

**src/App.js**

```
import React from "react";
import {Route, NavLink, HashRouter} from "react-router-dom";

import Home from "./component/Home";
import Tech from "./component/Tech";
import Help from "./component/Help";

class App extends React.Component {
  render() {
    return (
      <HashRouter>
        <h1>CSE183 Fall 2020</h1>
        <ul className="navigation">
          <li><NavLink exact to="/">Home</NavLink></li>
          <li><NavLink to="/tech">Tech</NavLink></li>
          <li><NavLink to="/help">Help</NavLink></li>
        </ul>
        <div>
          <Route exact path="/" component={Home}/>
          <Route path="/tech" component={Tech}/>
          <Route path="/help" component={Help}/>
        </div>
      </HashRouter>
    );
  }
}
```

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

**CSE183 Fall 2020**

- Home
- Tech
- Help

**Welcome**

Wise, you are, in taking this class.

32

**React Router Example**

Code for all three examples available for download after lecture

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

34

**Assignment 4 - Secret Sauce**


```
<head>
  <title> CSE183 Assignment 4 - Basic </title>
  <link href="templatser.css" rel="stylesheet" />
  <script type="text/javascript" src="Templatser.js"></script>
</head>
<body>
  <table>
    ...
    <div id="hidden">{{tr11}}</div>
    <textarea id="json" rows="8", cols="30">{ ... }</textarea>
    <button id="byTag">By Tag</button>
    <button id="byId">By Id</button>
  </body>
</html>
```

Need to create a `Templatser` instance and connect the  
By Tag By Id buttons to it

UCSC 850E CSE183 Fall 2020. Copyright © 2020 David C. Harrison. All Rights Reserved.

35

## Assignment 4 - Secret Sauce

```
<head>
<title> CSE183 Assignment 4 - Basic </title>
<link href="templater.css" rel="stylesheet" />
<script type="text/javascript" src="Templater.js"></script>
</head>
<body>
<table>
...
</table>
<div id="hidden">{(tr11)}</div>
<textarea id="json" rows="8", cols="30">{ ... }</textarea>
<button id="byTag">By Tag</button>
<button id="byId">By Id</button>
</body>
<script>
const templater = new Templater();
document.getElementById("byTag").addEventListener('click', function(e) {
    templater.byId(document, document.getElementById("json")).value;
});
document.getElementById("byId").addEventListener('click', function(e) {
    templater.byId(document, document.getElementById("json")).value;
});
</script>
</html>
```

### Problem?

The **By Tag** **By Id** buttons exist before the event handlers are attached ☺



36

## Assignment 4 - Secret Sauce

```
<head>
<title> CSE183 Assignment 4 - Basic </title>
<link href="templater.css" rel="stylesheet" />
<script type="text/javascript" src="Templater.js"></script>
<script>
const templater = new Templater();
document.getElementById("byTag").addEventListener('click', function(e) {
    templater.byId(document, document.getElementById("json")).value;
});
document.getElementById("byId").addEventListener('click', function(e) {
    templater.byId(document, document.getElementById("json")).value;
});
</script>
</head>
<body>
<table>
...
</table>
<div id="hidden">{(tr11)}</div>
<textarea id="json" rows="8", cols="30">{ ... }</textarea>
<button id="byTag">By Tag</button>
<button id="byId">By Id</button>
</body>
</html>
```

### Problem?

The **By Tag** **By Id** buttons do not exist when we try to attach the event handlers ☺



37

## Assignment 4 - Secret Sauce

```
<head>
<title> CSE183 Assignment 4 - Basic </title>
<link href="templater.css" rel="stylesheet" />
<script type="text/javascript" src="Templater.js"></script>
<script>
const templater = new Templater();
function byTag() {
    templater.byId(document, document.getElementById("json")).value;
}
function byId() {
    templater.byId(document, document.getElementById("json")).value;
}
</script>
</head>
<body>
<table>
...
</table>
<div id="hidden">{(tr11)}</div>
<textarea id="json" rows="8", cols="30">{ ... }</textarea>
<button id="byTag" onclick="byTag()">By Tag</button>
<button id="byId" onclick="byId()">By Id</button>
</body>
</html>
```

### Problem Solved

Attach event handlers as the buttons are being created ☺



38

## Assignment 4 - Secret Sauce

```
<head>
<title> CSE183 Assignment 4 - Basic </title>
<link href="templater.css" rel="stylesheet" />
<script type="text/javascript" src="Templater.js"></script>
<script>
window.addEventListener('DOMContentLoaded', function (e) {
    const templater = new Templater();
    document.getElementById("byTag").addEventListener('click', function (e) {
        templater.byId(document, document.getElementById("json")).value;
    });
    document.getElementById("byId").addEventListener('click', function (e) {
        templater.byId(document, document.getElementById("json")).value;
    });
});
</script>
</head>
<body>
<table>
...
</table>
<div id="hidden">{(tr11)}</div>
<textarea id="json" rows="8", cols="30">{ ... }</textarea>
<button id="byTag">By Tag</button>
<button id="byId">By Id</button>
</body>
</html>
```

### Alternate Solution

Attach event handlers after DOM is loaded ☺



39

## Upcoming Lectures

- Friday: Quiz 2
  - Plus Assignment 4 Review & Assignment 5 Introduction
- Monday: React Global State & Responsive Web Applications

## Tasks

- **Assignment 4** due 23:59 **Thursday October 29**
- Study for **Quiz 2** during class **Friday October 30**