Single Page Applications

Mendel Rosenblum
Web Apps and Browsers

- Web apps run in browsers (by definition)
- Users are use to browsing in browsers
  - Browser maintains a history of URLs visited
    - Back button - Go back in history to previous URL
    - Forward button - Go forward in history to next URL
  - Can move to a different page
    - Typing into location bar or forward/back buttons
    - Selecting a bookmarked URL
    - Page refresh operation
- Browser tosses the current JavaScript environment when navigating to a different page
  - Problematic for JavaScript frameworks: URL (and cookies) are the only information preserved
Problem with some web apps

- **Initial**: pages served from web server
  - Each page had a URL and app switched between pages served by web server

- **Early JavaScript apps**: Website a single page/URL with the JavaScript
  
  Problem: Restart web app on navigation (Can lose a lot of work!)
  ```javascript
  window.onbeforeunload = function(e) { return 'All will be lost!'; }
  ```

- **Users expect app in browser to do the right thing**: 
  - Navigate with forward and back buttons, browser page history
  - Navigate away and come back to the app
  - Bookmark a place in the app
  - Copy the URL from the location bar and share it with someone
  - Push the page refresh button on the browser
Changing URL without page refresh

- Can change hash fragment in URL without reload
  - http://example.com
  - http://example.com#fragment
  - http://example.com?id=3535
  - http://example.com?id=3535#fragment

- HTML5 give JavaScript control of page reload
  - Browser History API - window.history - Change URL without reloading page
Deep linking

- Concept: the URL should capture the web app's context so that directing the browser to the URL will result the app's execution to that context
  - Bookmarks
  - Sharing

- Context is defined by the user interface designer!
  Consider: Viewing information of entity and have an edit dialog open

  Should the link point to the entity view or to the entity & dialog?
  Does it matter if I'm bookmarking for self or sharing with others?
  How about navigating away and back or browser refresh?
Deep linking in Single Page Apps

Two approaches:

1. Maintain the app's context state in the URL
   + Works for browser navigation and refresh
   + User can copy URL from location bar

2. Provide a share button to generate deep linking URL
   + Allows user to explicitly fetch a URL based on need
   + Can keep URL in location bar pretty

Either way web app needs to be able to initialize self from deep linked URL
Ugly URLs

http://www.example.org/dirmod?sid=789AB8&type=gen&mod=Core+Pages&gid=A6CD4967199

versus

http://www.example.org/show/A6CD4967199

What is that ugly thing in the location bar above my beautiful web application?

https://www.flickr.com/photos/jarnasen/24593000826/in/explore-2016-01-26/
ReactJS support for SPA

- ReactJS has no opinion! Need 3rd party module.

- Example: React Router Version 5 [https://v5.reactrouter.com/](https://v5.reactrouter.com/)
  - Idea: Use URL to control conditional rendering
  - Newer version 6 is available using same concepts as v5 but slightly different syntax

- Various ways of encoding information in URL
  - In fragment part of the URL: HashRouter
  - Use HTML5 URL handler: BrowserRouter

- Import as a module:

  ```javascript
  import {HashRouter, Route, Link, Redirect} from 'react-router-dom';
  ```
Example React Router V5

<HashRouter>
  <div>
    ...
    <Route path="/states" component={States} />
    ...
    <Link to="/states">States</Link>
    ...
  </div>
</HashRouter>

- JSX block controlled by URL enclosed in HashRouter
- Route will render the component if URL matches.
- Use Link component to generated hyperlink: 
  <a href="#/states">States</a>
Passing parameters with React Router

- Parameter passing in URL

  ```
  <Route
    path="/Book/:book/ch/:chapter"
    component={BookChapterComponent}
  />
  ```

- Parameters put in prop.match of the component

  ```javascript
  function BookChapterComponent({ match }) {
    return ( <div>
      <h3>Book: {match.params.book}</h3>
      <h3>Chapter: {match.params.chapter}</h3>
    </div> );
  }
  ```
Route: component=, render=, children=

- **component={BookChapterComponent}**
  - Mounts components on match (unmounts on URL change)
  - Passes match object with: params, url, history

- **render={props => <BookChapterComponent book={props.match.params.book} chapter={props.match.params.chapter} />}**
  - Calls function with props having match object from above.
  - Doesn't mount/unmount component (does update it)

- **children= - Like render= except is called regardless of the match**
  - match will be null if URL doesn't match
  - Useful if you want to have something always render but only active on matching URL.

Multiple route matches have precedence order: component, render, children

*Switch is useful with multiple Route - Renders the first matching one*
Example

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Phone</th>
<th>Email</th>
<th>Birthdate</th>
<th>Last Access</th>
<th>Rating</th>
<th>Name</th>
<th>Salary</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>Alexandra Nixon</td>
<td>(422) 644-3488</td>
<td><a href="mailto:rectract@virus.com">rectract@virus.com</a></td>
<td>12/01/1981</td>
<td>February 25, 2001</td>
<td>-3</td>
<td>41%</td>
<td>$46,672</td>
<td>7.5</td>
</tr>
<tr>
<td>17</td>
<td>Alisa Monroe</td>
<td>(859) 974-4442</td>
<td><a href="mailto:adcosa@ncam.edu">adcosa@ncam.edu</a></td>
<td>02/14/1990</td>
<td>April 30, 2003</td>
<td>6</td>
<td>95%</td>
<td>$103,099</td>
<td>5.9</td>
</tr>
<tr>
<td>10</td>
<td>Baker Osborn</td>
<td>(378) 371-0598</td>
<td><a href="mailto:turbula@kne.ac.edu">turbula@kne.ac.edu</a></td>
<td>03/29/1970</td>
<td>July 23, 2005</td>
<td>-7</td>
<td>61%</td>
<td>$2,868</td>
<td>0.1</td>
</tr>
<tr>
<td>9</td>
<td>Caldwell Larson</td>
<td>(830) 362-3177</td>
<td><a href="mailto:alit@alter.com">alit@alter.com</a></td>
<td>07/02/1983</td>
<td>June 22, 2004</td>
<td>-3</td>
<td>81%</td>
<td>$63,736</td>
<td>7.5</td>
</tr>
<tr>
<td>25</td>
<td>Charissa Manning</td>
<td>(438) 395-9392</td>
<td><a href="mailto:rtv.vuata@necno.org">rtv.vuata@necno.org</a></td>
<td>07/01/1980</td>
<td>April 02, 2005</td>
<td>-8</td>
<td>11%</td>
<td>$32,199</td>
<td>3.5</td>
</tr>
<tr>
<td>48</td>
<td>Charlie Hahn</td>
<td>(395) 200-9188</td>
<td><a href="mailto:ac@dueusys.edu">ac@dueusys.edu</a></td>
<td>08/04/1978</td>
<td>January 17, 2009</td>
<td>-2</td>
<td>86%</td>
<td>$3,246</td>
<td>3.3</td>
</tr>
<tr>
<td>30</td>
<td>Dorian Hodge</td>
<td>(304) 536-6850</td>
<td><a href="mailto:celius@secrect.org">celius@secrect.org</a></td>
<td>08/16/1978</td>
<td>February 21, 2007</td>
<td>6</td>
<td>6%</td>
<td>$28,057</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>Ezekiel Hart</td>
<td>(627) 536-4760</td>
<td><a href="mailto:tortor@ent.ca">tortor@ent.ca</a></td>
<td>12/02/1962</td>
<td>March 26, 2009</td>
<td>-7</td>
<td>7%</td>
<td>$73,229</td>
<td>6.9</td>
</tr>
<tr>
<td>12</td>
<td>Fletcher Briggs</td>
<td>(992) 962-9419</td>
<td><a href="mailto:amet.ante@lansue.edu">amet.ante@lansue.edu</a></td>
<td>08/12/1971</td>
<td>December 12, 2006</td>
<td>7</td>
<td>23%</td>
<td>$142,448</td>
<td>8.9</td>
</tr>
<tr>
<td>40</td>
<td>Fritz Benton</td>
<td>(323) 353-2984</td>
<td><a href="mailto:a@gammury.com">a@gammury.com</a></td>
<td>10/02/1957</td>
<td>June 16, 2002</td>
<td>-3</td>
<td>2%</td>
<td>$75,654</td>
<td>8.9</td>
</tr>
</tbody>
</table>

What to keep in URL: table length, viewport in table, search box, sort column, etc. Is it different for bookmark or share? Nav away and back?
Example: Not everything goes in URL