Single Page Applications

Mendel Rosenblum
What does browser do on navigation ops?

- **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

- **Browser tosses the current JavaScript environment when navigating to a different page**
  - Typing into location bar or forward/back buttons
  - Selecting a bookmarked URL
  - Page refresh operation
  - Assignments to window.location

- **Challenge: Any state in JavaScript goes away and browser restarts page**
  - URL (and cookies) are the only information preserved
Changing URL without page refresh

- Can change hash fragment and query parameters 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
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
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

VS

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/
Angular Support for SPA

- Idea: Do client-side routing of URLs to view components
  - Several different Angular URL routing frameworks written, no clear winner at this point

- ngRoute - provides routing and deep linking
  - We'll use this one in class

- ngRoute is an Angular API that provides:
  - A directive (ngView) to indicate where view components should be inserted in template
  - A service ($route) that watches window.location for changes and updates the displayed view.
  - A configuration ($routeProvider) that allows the user to specify the mappings of URLs to view components.
Using ngRoute

● In a view template (frequently the "shell" of the web application)

   `<div ng-view></div>`

● In your Angular module add ngRoute as a dependency

   `angular.module('cs142App', ['ngRoute'])`

● Configure the routing table in a module config block

   `cs142App.config(['$routeProvider', function($routeProvider) {`
ngRoute - Specify URL ⇒ View mapping

cs142App.config(['$routeProvider', function($routeProvider) {
    $routeProvider
        .when('/Book/:bookId', {
            templateUrl: 'book.html',
            controller: 'BookController',
        })
        .when('/Book/:bookId/ch/:chapterId', {
            templateUrl: 'chapter.html',
            controller: 'ChapterController',
        });
}]);
ngRoute - Passing parameters to controllers

```javascript
<a href="#/Book/Moby">...</a>

cs142App.controller('BookController', ['$routeParams', function($routeParams) {
    $routeParams.bookId // Will be "Moby"
}]);

<a href="#/Book/Gatsby/ch/3">...</a>

cs142App.controller('ChapterController', ['$routeParams', function($routeParams) {
    $routeParams.bookId // Will be "Gatsby"
    $routeParams.chapterId // Will be "3"
}]);
```
ngRoute handles query params as well

```html
<a href="/Book/Moby?noShow=true&upsideDown=Yes">...</a>
```

```javascript
cs142App.controller('BookController', ['$routeParams',
    function($routeParams) {
        $routeParams.bookId // Will be "Moby"
        $routeParams.noShow // Will be "true"
        $routeParams.upsideDown // Will be "Yes"
    }]);
```
Example

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

<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Expire Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Trash Can]</td>
<td>![Trash Can]</td>
<td>Milk</td>
</tr>
<tr>
<td>![Trash Can]</td>
<td>![Trash Can]</td>
<td>Eggs</td>
</tr>
<tr>
<td>![Trash Can]</td>
<td>![Trash Can]</td>
<td>Steak</td>
</tr>
<tr>
<td>![Trash Can]</td>
<td>![Trash Can]</td>
<td>Oranges</td>
</tr>
</tbody>
</table>

Delete 'Eggs'
Are you sure you want to remove this item?

- [✓] OK
- [✗] Cancel