Require.js

Objectives

- Understand module loading in JavaScript
- Understand why asynchronous loading is necessary
- Understand how to use require
- Use require with jQuery

Why modules?

- As we build 'apps' not 'sites' code complexity grows
  - Need to manage the complexity
  - Fewer globals
  - Assembly of sites gets more complex
  - Would like to be able to optimize the code

Solution

- Use some for of modules
  - cf 'import', 'include' in other languages

CommonJS

- Defined an API for loading modules
- Synchronous
  - Great on the server
  - Not so great in the browser

```
var User = require("types/User");

function UserManager () {
}

//Error if require call is async
UserManager.prototype = new User();
```

Attempts to provide async loading

- XHR
  - uses eval
  - eval is evil
- Web Workers
  - Not all browsers (IE < 10)
- document.write
  - Need to know all the required scripts ahead of time
  - Does not work after pageload (perceived performance is bad)

Enter AMD

- Asynchronous Module Definition
  - Provides a mechanism to encapsulate modules
  - and a way to specify dependencies

AMD Example

- define – define the module
- 'jquery' – specify dependencies
- $ - reference to the loaded dependency
- function($) – factory, executed after dependencies load

```
define(['jquery'] , function ($) {
    return function () {};
});
```

AMD - dependencies

- Dependencies are string values
  - 'uitls/helper'
  - 'jquery'
- Follows CommonJS practice

AMD Modules

- Modules wrapped in a 'define' call
  - Allows AMD to resolve dependencies
  - Execute inner (factory) function after dependencies loaded
  - Does not litter global namespace

Naming modules

- Can also name modules
  - May have multiple modules per file
  - This is discouraged

AMD Loaders

- AMD needs a loader to load the JavaScript
  - Dojo (1.7+)
  - curl
  - lsjs
  - require

require.js

- AMD loader
  - Provides a standard way to load modules
  - Also provides an optimizer (uses node.js or Java)

Require Examples

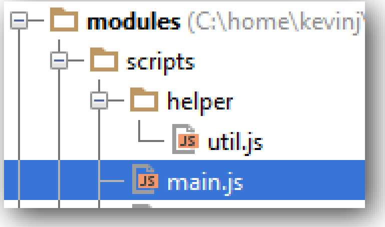
- Set up the page to use required
  - Include require.js
  - Use data-main to reference initial JavaScript

```
<html>
<head>
    <title>Test Suite</title>
    <script data-main="scripts/main"
            src="scripts/require.js"></script>
</head>
<body>
</body>
</html>
```

Initial JavaScript

- This is the page JavaScript
  - Can reference other Javascript to be used in this
  - The 'references' are relative to the load location

```
// main.js
require(["helper/util"], function(util) {
    util.doSomething();
});
```



modules (C:\home\kevinj\
scripts
helper
util.js
main.js

```
// util.js
define(function() {
});
```

References

- Code is loaded relative to base url
  - set in data-main (scripts/main)
  - set via config (later)
  - if neither of these then url of loaded html is used

Loading Modules

- RequireJs assumes that all dependencies are scripts
  - No need to specify .js at end of module ids
- Can override this behavior and use regular URL
  - End module with .js
  - Start module with '/'
  - Start module with url protocol (e.g. 'http:' or 'https:')

Define a module

- **With no dependencies**

```
define(function() {
    return {
        name: 'Kevin',
        id: 1
    }
});
```

Define a module

- **With dependencies**
  - **function is not called until module is loaded**

```
define(['resources'], function(res) {
    return {
        name: res.resource('name'),
        id: 1
    }
});
```

Define a module

- Modules can return 'anything'

```
define(['...'], function(a) {
   function BlogPost(item, parent) {
   };

    return BlogPost;
});
```

Configuration

- Sometimes need to configure requires
  - Set base path
  - Set module names
  - Set loading time

Configuration

- Specify configuration at startup
  - Can be used for versioning
  - Useful for changing relative paths

```
<script src="scripts/require.js"></script>
<script>
    require.config({
        baseUrl: "/someUrl",
        paths: {
            "services": "app/services",
            "viewmodels": "app/viewmodels",
            "knockout": "knockout-2.2.1"
        },
        waitSeconds: 15
    });
</script>
```

Summary

- Very useful to be able to modularize applications
- requirejs provides a way to load modules
- Can test
  - different test libraries may require different setup