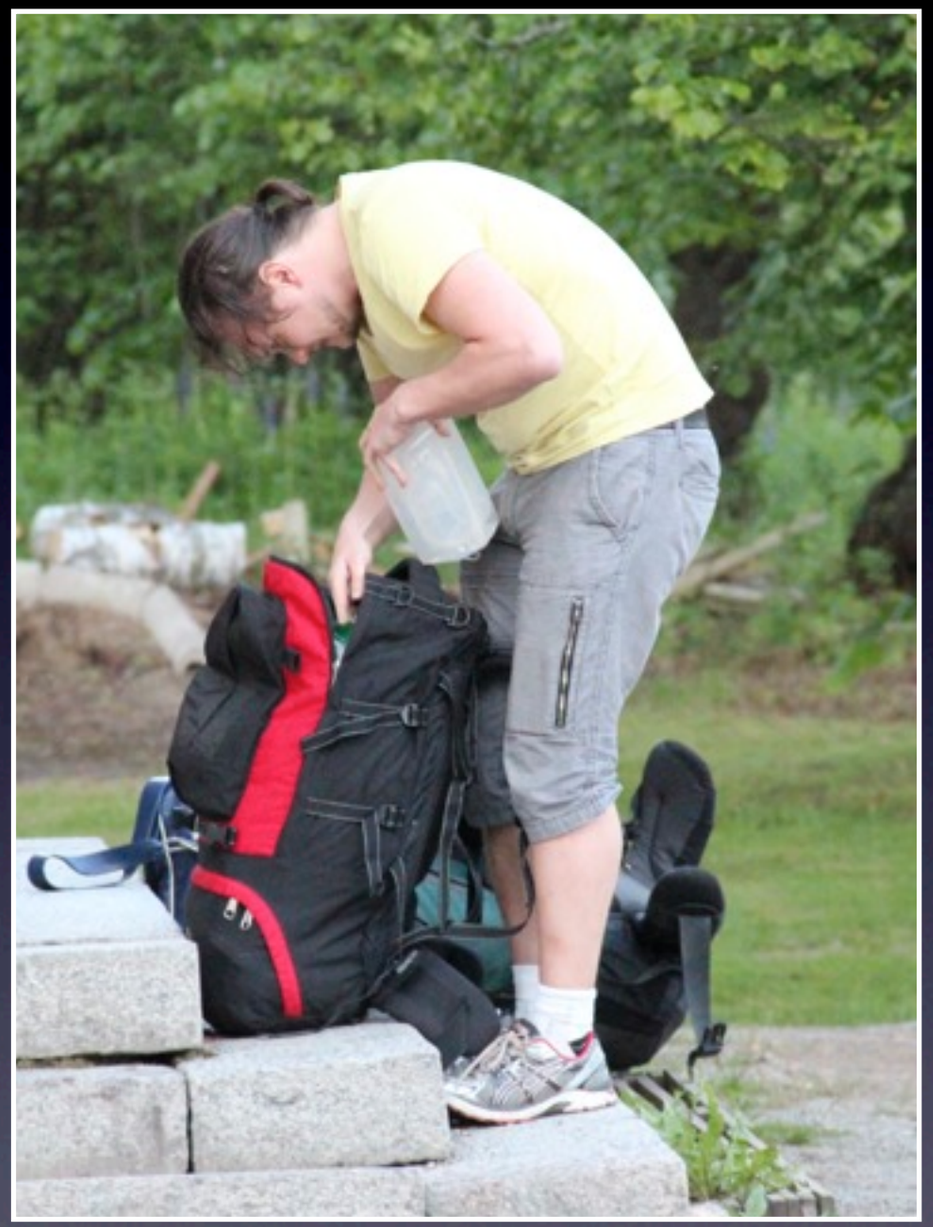
Efficient Beautiful Maintainable Modular Javascript Codebase with RequireJS

HelsinkiJS 2012 June
Mikko Ohtamaa

Bio

Mikko Ohtamaa
moo9000 @ Twitter
http://opensourcehacker.com



Audience

- Front end developers working with medium and large size Javascript codebases(> 2k lines of code)

Grab the example code

- https://github.com/miohtama/require-js-mooapp-tutorial

Javascript module systems

- Global namespace objects: YUI, ExtJS, jQuery
- Node: CommonJS
- **RequireJS**

RequireJS

- Javascript module loader for client-side
- Version 2.0 still fresh
- MIT

Asynchronous Module Definition (AMD)

- No wait for <script> tag to complete before loading the next Javascript file
- UMD (Universal Module Definition): boilerplate which makes your JS file to co-operate with several module systems
- https://github.com/umdjs/umd

Y U Require JS?

Benefit #1: More maintainable code base

Benefit #2: parallel loading of non-interdependent modules

Benefit #3: automatic, more optimized, minification

Benefit #4: avoiding globals

```
COMMON_FILES = [
'Three.js',
'core/Clock.js',
'core/Color.js',
'core/Vector2.js',
'core/Vector3.js',
'core/Vector4.js',
'core/Frustum.js',
'core/Ray.js',
'core/Rectangle.js',
'core/Math.js',
'core/Matrix3.js',
'core/Matrix4.js',
'core/Object3D.js',
'core/Projector.js',
'core/Quaternion.js',
'core/Vertex.js',
'core/Face3.js',
'core/Face4.js',
'core/UV.js',
'core/Geometry.js',
'core/Spline.js',
'cameras/Camera.js',
'cameras/OrthographicCamera.js',
'cameras/PerspectiveCamera.js',
'lights/Light.js',
'lights/AmbientLight.js',
'lights/DirectionalLight.js',
'lights/PointLight.js',
'lights/SpotLight.js',
```

```
11
12
13    <script src="jquery.js"></script>
14    <script src="jquery-buddy.js"></script>
15    <script src="file1.js"></script>
16    <script src="file2.js"></script>
17    <script src="file3.js"></script>
18    <script src="file3-maybe-4.js"></script>
19    <script src="file-was-this-5.js"></script>
20
21
```

ENTERPRISE JAVASCRIPT IS:

Refresh

NAMESPACE CODE INTO A "PROPER PACKAGE STRUCTURE" TO MAKE BACKEND DEVS FEEL AT HOME.

```
var com = {,
    AwesomeCo: {
        util: {
            info: function ( message) {
                alert(message);
                return message;
            }
        }
    }
};

com.AwesomeCo.util.info("SRSLY!?");
```

Using RequireJS

Basic JS development challenges

- Maintaining internal code dependencies
- Integrating third party libraries
- Exporting packages for production and distribution

Example codebase: mooapp

```
# Fancy designer stuff
./css
./css/colorpicker.css
./images

# Entry point
./index.html

# Our code
./mooapp/main.js
./mooapp/moo.js
./mooapp/mooficator.js

# Something we stole from the internets
./thirdparty/colorpicker.js
./thirdparty/jquery.js
./thirdparty/require.js
```

Defining a module

- **define(deps, creator)** - RequireJS global function
- deps: Array of modules we depend on
- creator: function() which takes argument 1 modules as parameters
- creator **returns** module exports

```
define(["jquery","js/internalmod"], function($,
internalmod) {
    return { someVar : 5 }
});
```

moo.js

mooficator.js

Configuring RequireJS

- **require()** and **require.config()**
- Base URL
- Paths where modules can be found
- Shim and dependencies for non-AMD-compatible modules
- Do in JS so can be later consumed by optimizer

main.js

HTML file

is now clean

index.html

RequireJS limitations

- Source code tree SHOULD match module paths
- Shim existing libraries
- Not file:// compatible

```
$ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Deployment

Let's get the party started

r.js - the optimizer

- Analyzes require() and define() dependencies of a JS entry point
- Will merge and bundle everything to one minified JS file preserving the correct dependency loading order
- No globals > every var can be minified > more compact minification

The Magic

define() and require() functions perform correctly in the merged bundle and do not actually need separate JS files

No HTML changes on production needed!

...because main.js is the only JS reference in index.html...

Makefile

Bundled

```
$ ls -lh dist/mooapp
total 148K
-rw-r--r-- 1 moo staff 147K Jun 28 17:12 main.js

$ ls -lh mooapp
-rw-r--r-- 1 moo staff 1.5K Jun 28 17:11 main.js
-rw-r--r-- 1 moo staff 1.3K Jun 28 16:20 moo.js
-rw-r--r-- 1 moo staff  840 Jun 28 16:18 mooficator.js
```

Thank You

@moo9000

http://opensourcehacker.com