

Optimizing Modules for Distribution

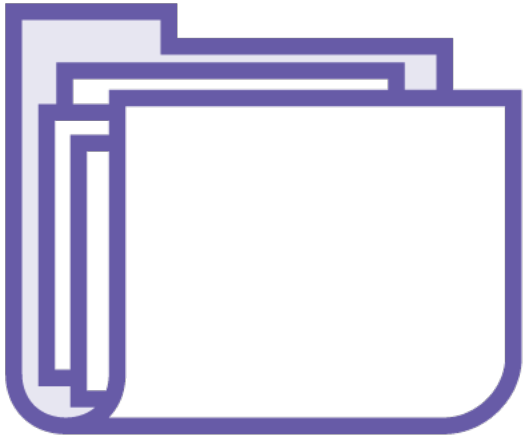


Kevin Murray

murmeister@hotmail.com



Modular Files



Increased number of
project files



Browsers limit
concurrent downloads



Tools available to
consolidate files



Overview



RequireJS optimization tool

- Node
- Java
- Browser (some limitations)

Not installing Node

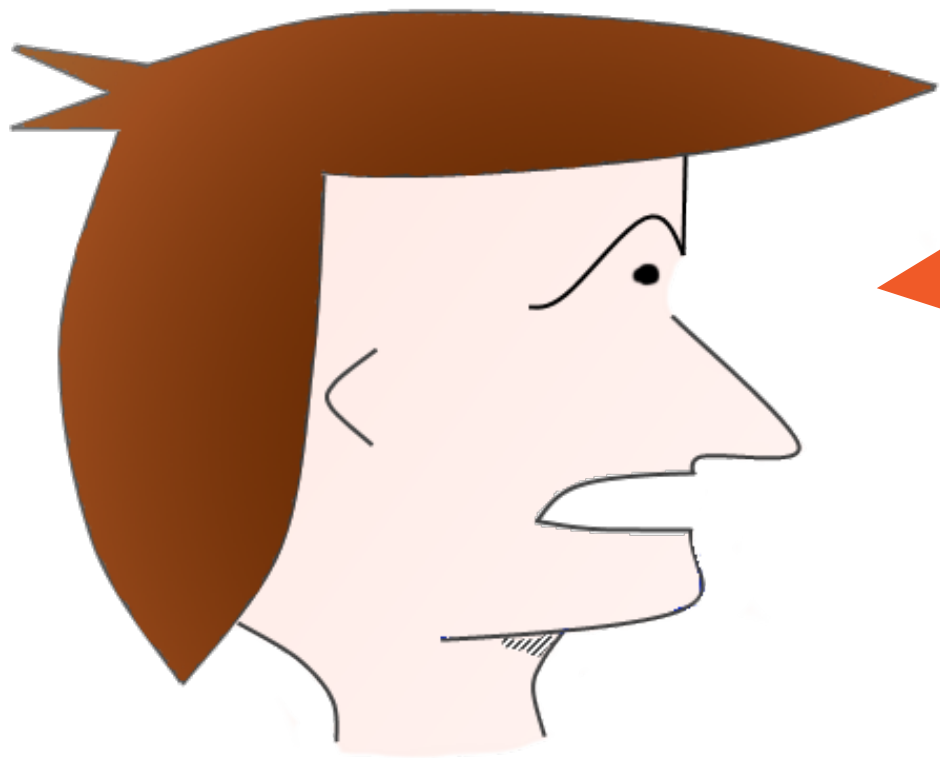
Download optimizer (r.js)

Focus on helpful options



Over 450 Files





450 Files? No way!



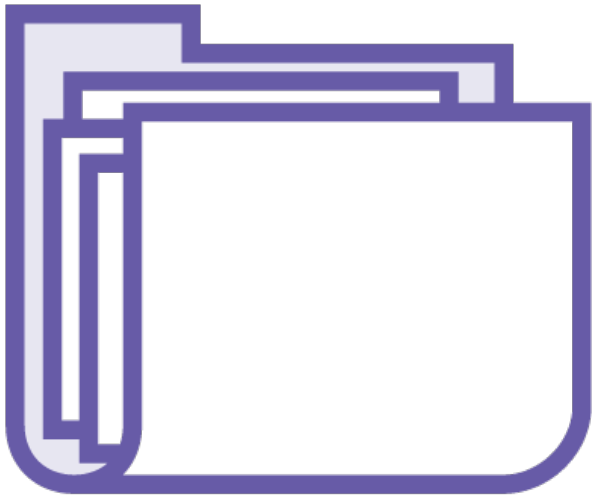
jQWidgets

in
co

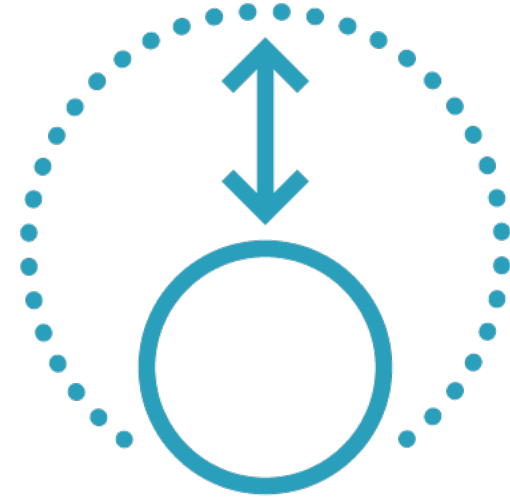
ssible CSS
emes



Actual Project Files



Between 30 and 40 files



Bigger than expected

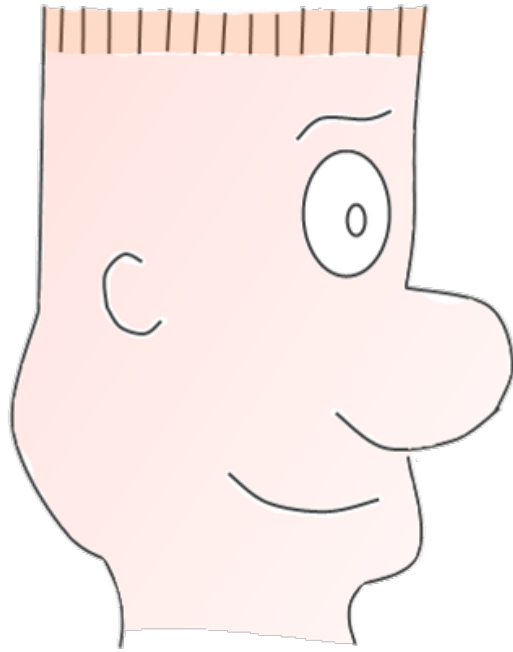
Going Live

**Files must be
downloaded**

**Development
environment has
immediate access**

**Users have
bandwidth and
browser
limitations**

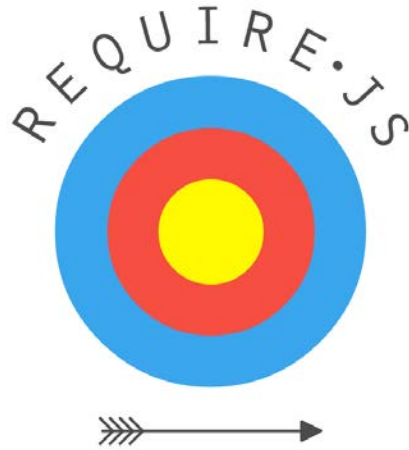




Now that I know AMD, I'm not
building large libraries again!



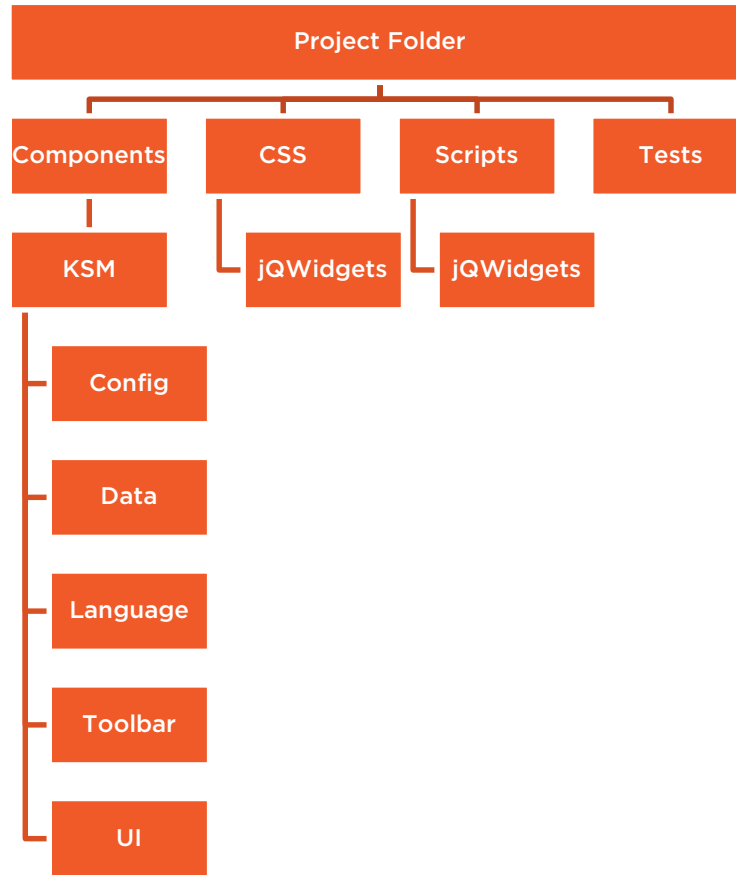
Consolidating Modules

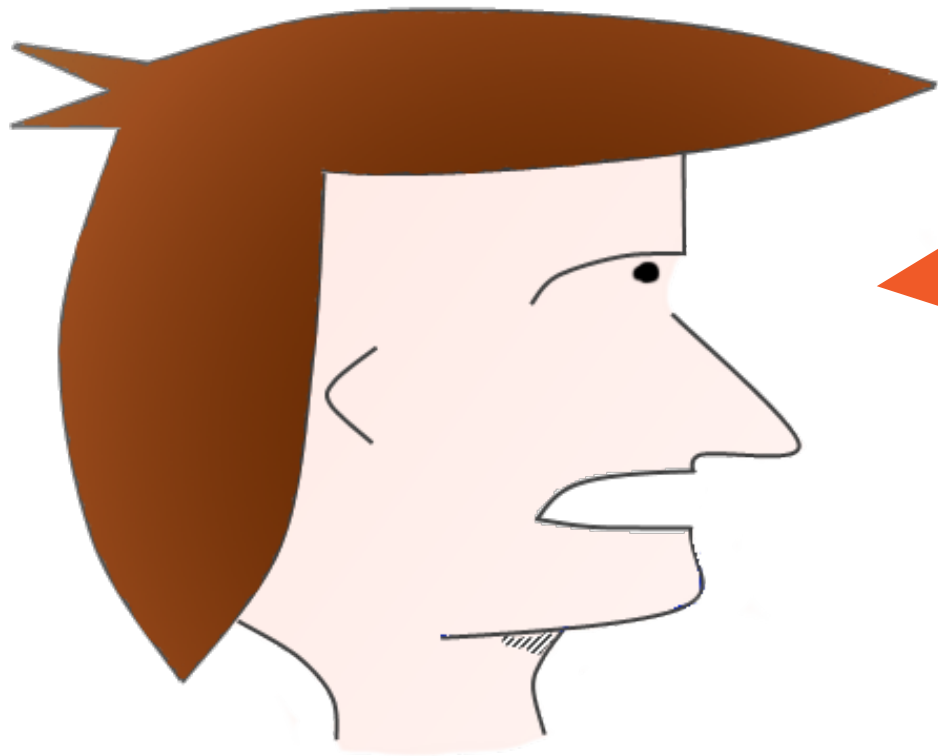


RequireJS optimization tool



Project Organization





Modules are maintainable -
libraries are a headache!



RequireJS Optimizer

Node version runs via
command line

Accepts numerous parameters

Most options can be applied
with the command line

Command line option quickly
become impractical



Build Configuration File



Place all options in configuration file



“out” Property



File name for consolidated modules

KSM_Optimized.js

“name” Property



Starting point for code



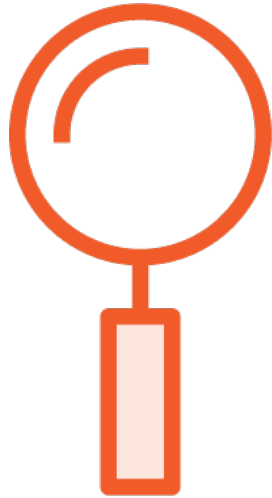


Missing modules

- Toolbar
- Language
- Footer
- UI, loaders, and proxies

Included in paths property

Default Behavior



Only inspects dependencies listed in
start code (not paths)

Start File

jQuery

Config

UI

Only four files associated with
startup code



Waiting for Optimizer



30 second delay – feels much longer



Optimizing Issue

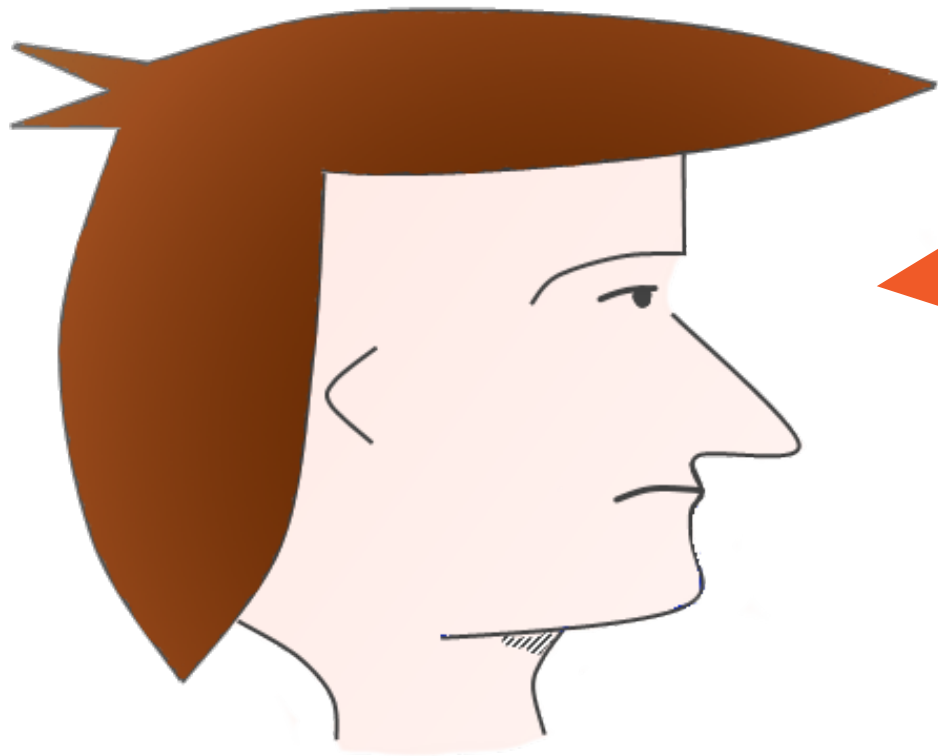
**Large JavaScript libraries
causing delays**

**Exclude third-party libraries
from optimization**

**Many libraries available via
CDN**

**Structure delivery based on
client platform – or need**





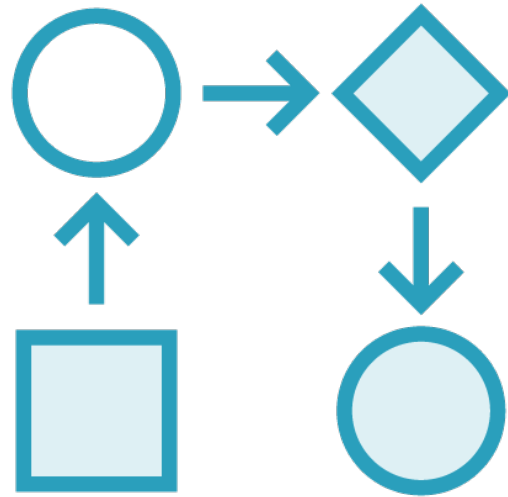
Just use an "exclude" property!



There is no property that
lets you specify one or
more files to exclude from
optimization



Another Way



Build Configuration “paths”



Copied from startup
configuration



Duplicate
configurations



Single configuration is
desirable

Optimizer Solution



Available since version 2.1.10

“mainConfigFile”



“mainConfigFile”

Specifies startup
module containing
configuration

Startup “paths”
will be used for
optimization

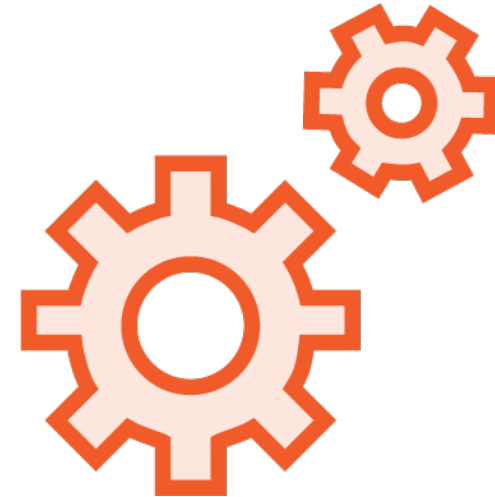
Single “paths”
property to
maintain



Pointer Problem



Code needs actual pointer to
libraries



Build file needs “empty:” reference

Optimizer Solution

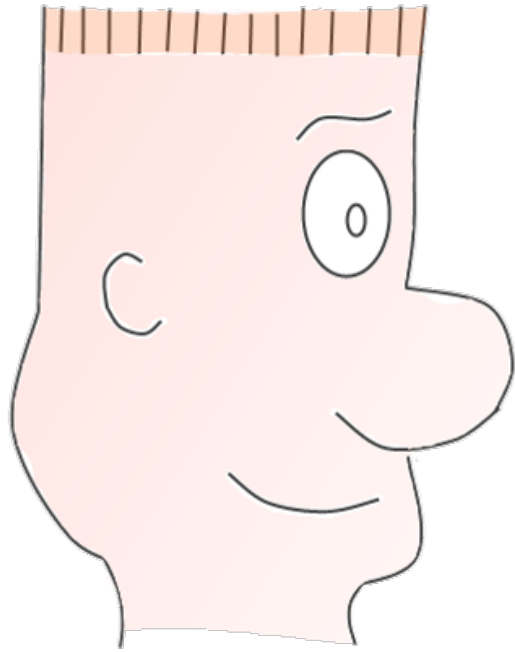


Results



New modules automatically
participate in optimization





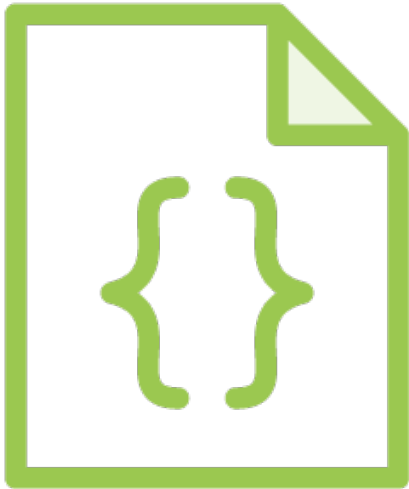
How is the optimized code
referenced in the HTML?



Same Architecture



More Optimizer Information

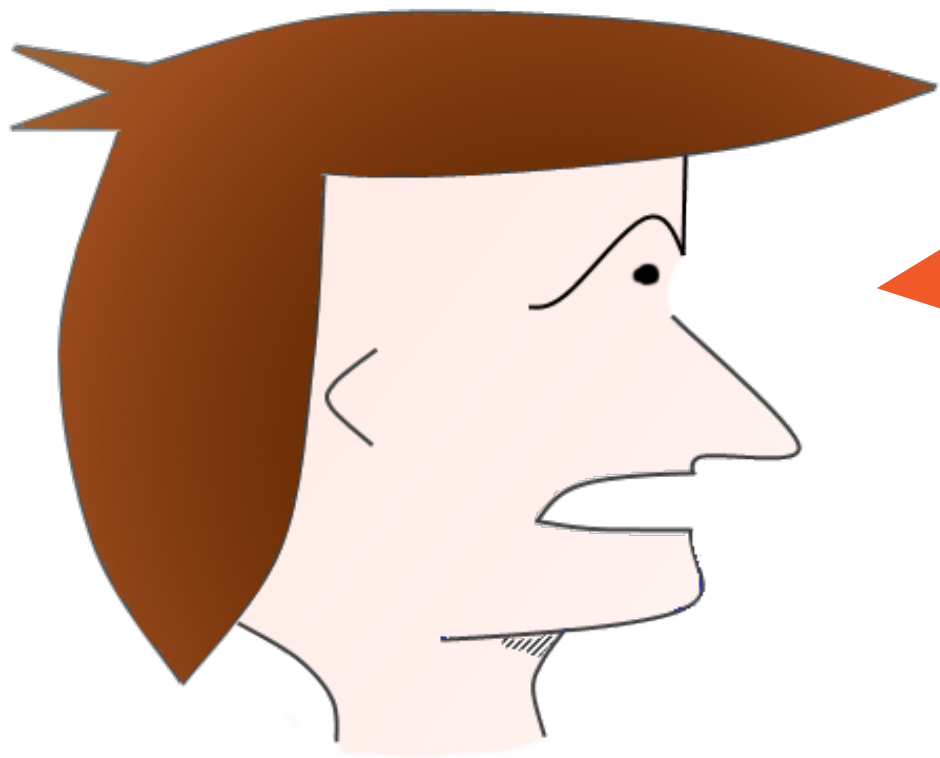


Optimize CSS files



requirejs.org/docs/optimization.html





At the end already?



Summary



Refactored legacy libraries

- Only when necessary

Footer module is template

“shim” options

Sample UI library

Decoupled modules

Bundled modules with optimizer



JavaScript Asynchronous Module Definition (AMD) Explained



Kevin Murray

murmeister@hotmail.com

