



Hello.



FULLSTACK
ACADEMY *of* CODE



FULLSTACK
ACADEMY *of* CODE



Browserify

“I built Browserify because I
wanted to keep writing node”

- James Halliday (@**substack**)



- Allows for the use of the **CommonJS module system** in the browser

- Allows for the use of the **CommonJS module system** in the browser
- Shims **node** core libraries for the browser*

- Allows for the use of the **CommonJS module system** in the browser
- Shims **node** core libraries for the browser*
- Allows for **node_modules** to be used in the browser*

Getting Started

CLI: `npm install -g browserify`

or

your favorite build process tool



node core modules

- require(**url**) — work sanely with url paths!
- require(**streams**) — .pipe in the browser!
- require(**crypto**) — hashing and encryption!
- require(**events**) — event emitters!
- new **Buffer**(0) — UInt8Arrays and ArrayBuffer!

~~fs.readFile()~~

require('os').platform()

===

'browser'

~~http.createServer()~~



npm modules



transforms





1.3.14 instantaneous-browserification (2015-02-24)

Features

- **CommonJS:** - angular modules are now packaged for npm with helpful exports



npm install **angular**

where angular and browserify clash

- Two module systems

where angular and browserify clash

- Two module systems
- Passing application value around is weird

where angular and browserify clash

- Two module systems
- Passing application value around is weird
- Many requires are used to “run” the file, rather than work with an export

why you should use browserify anyway

- Synchronous (technically pre-processed)

why you should use browserify anyway

- Synchronous (technically pre-processed)
- Your development process is your build process

why you should use browserify anyway

- Synchronous (technically pre-processed)
- Your development process is your build process
- Reusable server-side and client-side code

why you should use browserify anyway

- Synchronous (technically pre-processed)
- Your development process is your build process
- Reusable server-side and client-side code
- Ease of transformations

why you should use browserify anyway

- Synchronous (technically pre-processed)
- Your development process is your build process
- Reusable server-side and client-side code
- Ease of transformations
- Transferrable to future exploration of technologies



Thanks!