webpack

ACADEMY

Web Performance

https://github.com/thelarkinn/webpack-workshop-2018

https://frontendmasters.com/courses/performance-webpack/

Web Performance

Top 3 Web Page Load Time Causes:

Amount of JavaScript For Initial Download

Amount of CSS For Initial Download

AMount of Network Requests on Initial Download

Goals:

<=200kB (uncompressed) Initial JavaScript [total]

<=100kB (uncompressed) Initial CSS [total]

HTTP: <= 6 Initial Network Calls
HTTP/2: <= 20 Initial Network Calls

90% Code Coverage (only 10% code unused)

Code Splitting

Code Splitting

code splitting

All Videos News Shopping Images More Settings Tools

About 36,300,000 results (0.68 seconds)

code splitting - Webpack

https://webpack.github.io/docs/code-splitting.html

This feature is called "**code splitting**". It's an opt-in feature. You can define split points in your code base. Webpack takes care of the dependencies, output files ...

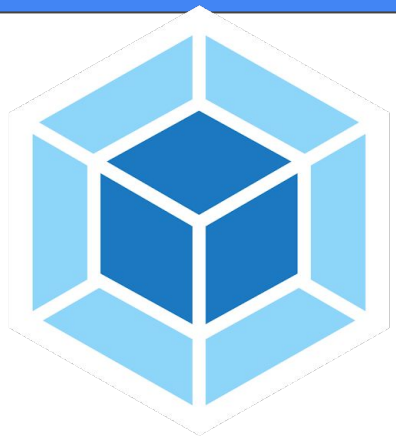

GWT

For example, here is the initial, unsplit Hello sample that comes with GWT:

```
public class Hello implements EntryPoint {
  public void onModuleLoad() {
    Button b = new Button("Click me", new ClickHandler() {
      public void onClick(ClickEvent event) {
        Window.alert("Hello, AJAX");
      }
    });
```

Suppose you wanted to split out the `Window.alert` call into a separate code download. The following code accomplishes this:

```
public class Hello implements EntryPoint {
  public void onModuleLoad() {
    Button b = new Button("Click me", new ClickHandler() {
      public void onClick(ClickEvent event) {
        GWT.runAsync(new RunAsyncCallback() {
          public void onFailure(Throwable caught) {
            Window.alert("Code download failed");
          }

          public void onSuccess() {
            Window.alert("Hello, AJAX");
          }
        });
      }
    });

    RootPanel.get().add(b);
  }
}
```

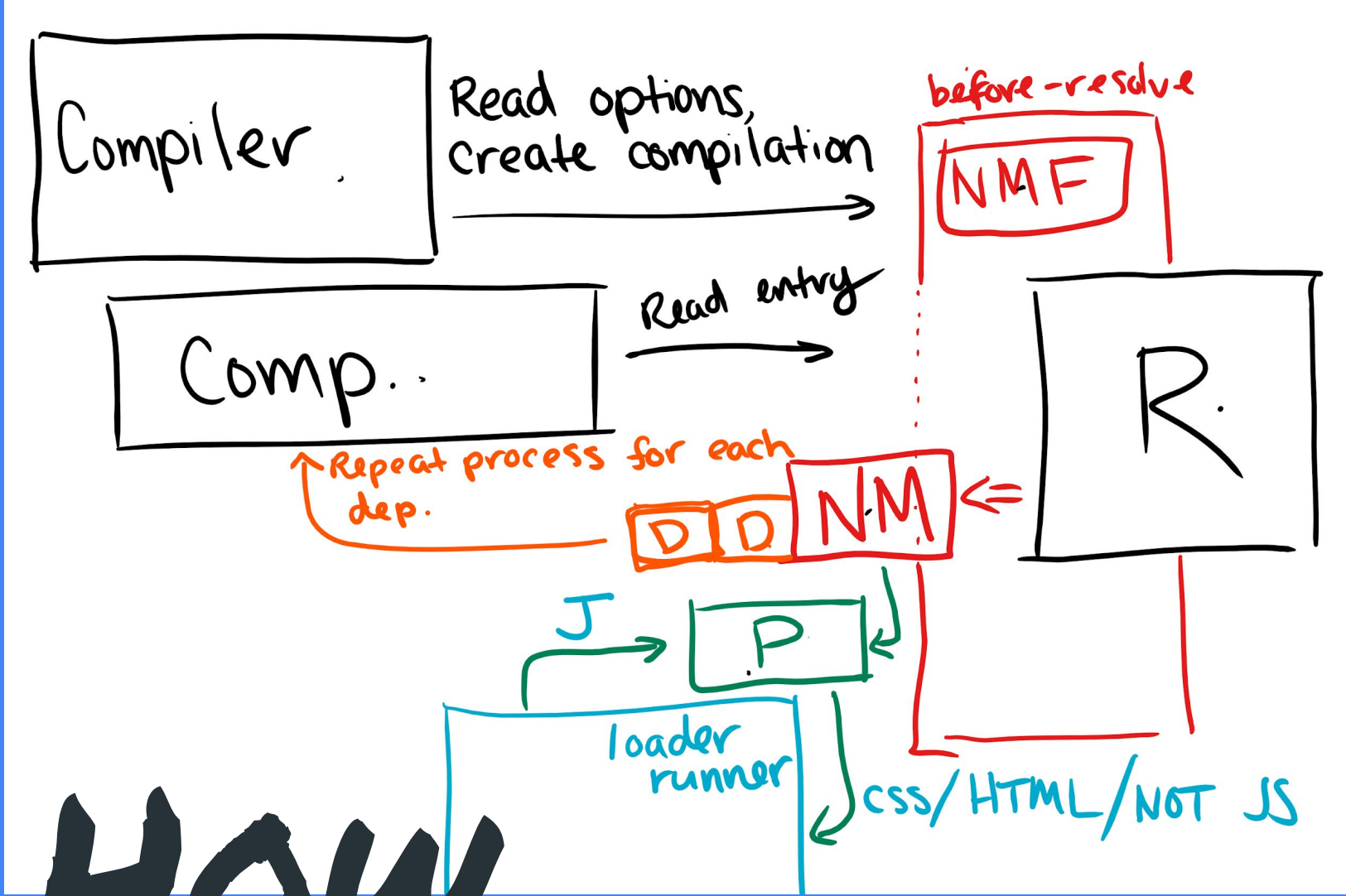


webpack

Code Splitting

Process of splitting pieces of your code into async chunks [at build time]

How Does it Work?


Compiler
Read options, create compilation
before-resolve
NMF
Comp..
Read entry
R.
Repeat process for each dep.
D
D
NM
J
P
loader runner
CSS/HTML/NOT JS

HOW...

Why should I care?

Why...

The future of web is mobile

The average mobile website takes 14 seconds to get interactive

Load less code => interactive faster.

Two Types

Two Types

static

"dynamic"

Static

When to use:

"Heavy" JavaScript

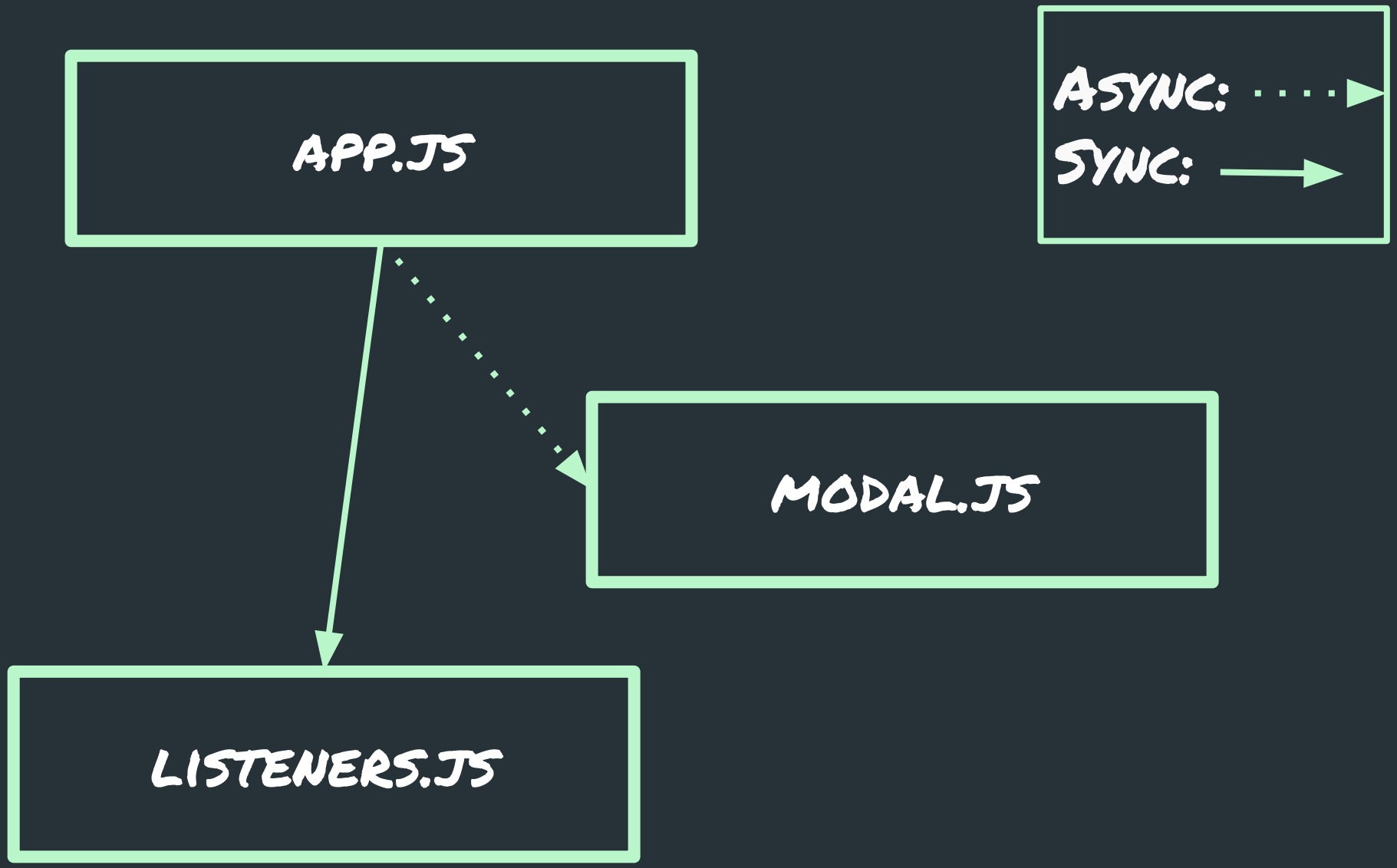
Anything temporal

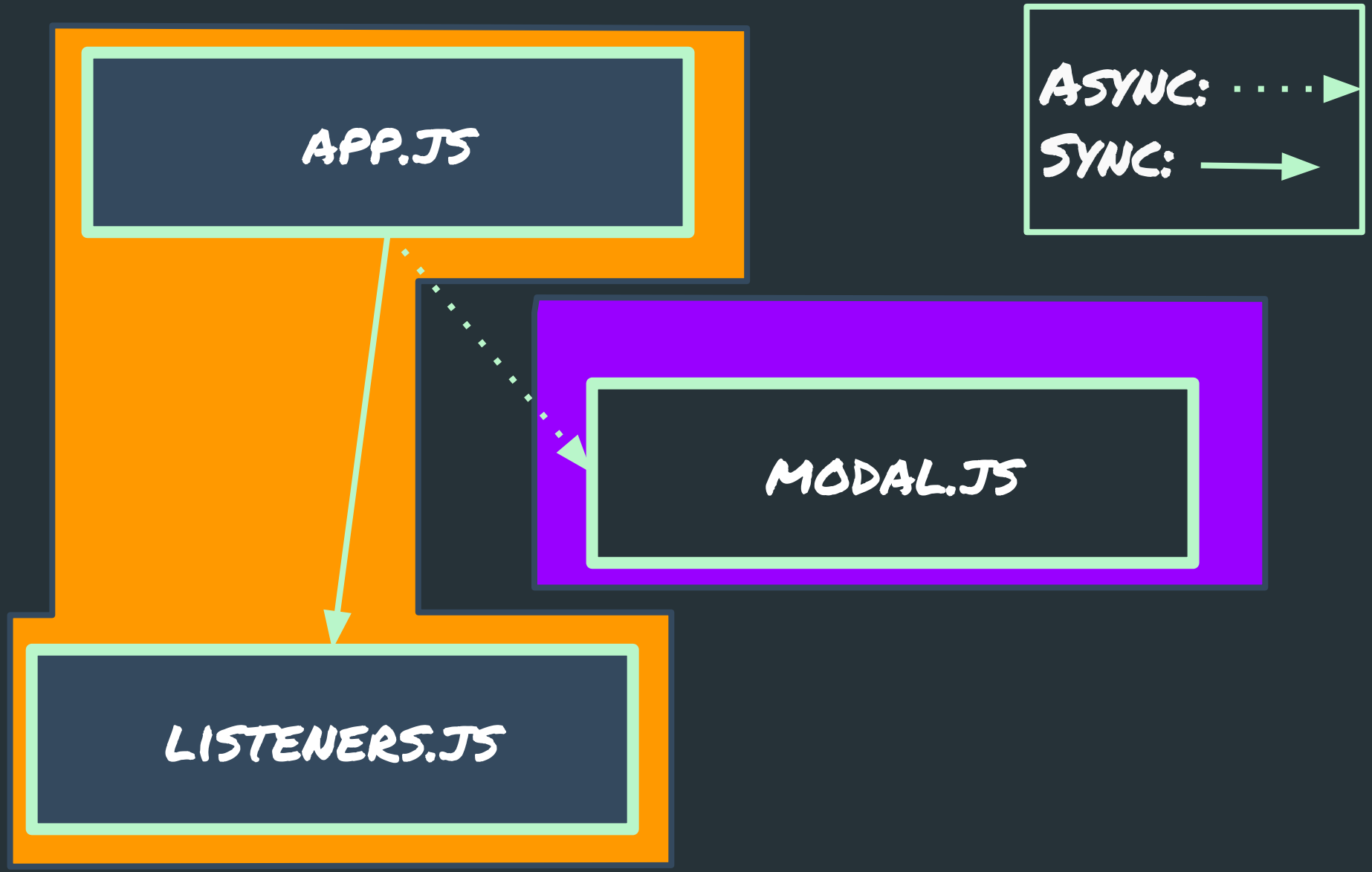
Routes

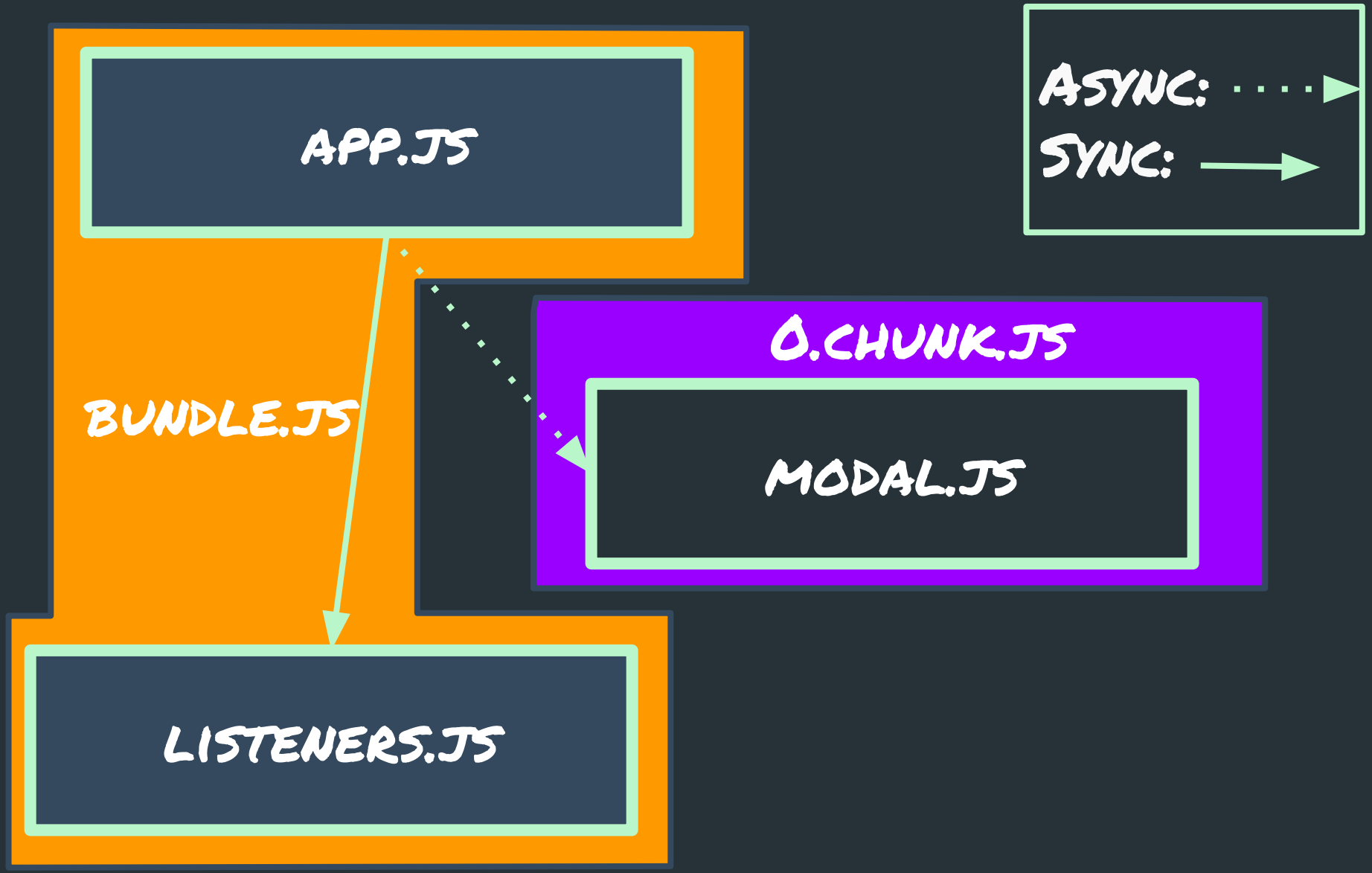
```
import Listener from './listeners.js';

const getModal = () => import('./src/modal.js');

Listener.on('didSomethingToWarrentModalBeingLoaded', () => {
  // Async fetching modal code from a separate chunk
  getModal().then((module) => {
    const modalTarget = document.getElementById('Modal');
    module.initModal(modalTarget);
  });
});
```

```
import Listener from './listeners.js';

const getModal = () => import('./src/modal.js');

Listener.on('didSomethingToWarrentModalBeingLoaded', () => {
  // Async fetching modal code from a separate chunk
  getModal().then((module) => {
    const modalTarget = document.getElementById('Modal');
    module.initModal(modalTarget);
  });
});
```

ALWAYS RETURNS A PROMISE


ASYNC:
SYNC:
APP.JS
MODAL.JS
LISTENERS.JS


ASYNC:
SYNC:
APP.JS
MODAL.JS
LISTENERS.JS


ASYNC:
SYNC:
APP.JS
BUNDLE.JS
0.CHUNK.JS
MODAL.JS
LISTENERS.JS

"DYNAMIC"

```javascript
const getTheme = (themeName) => import(`./src/themes/${themeName}`);

// Using `import()` 'dynamically'
if (window.feeling.stylish) {
  getTheme("stylish").then((module) => {
    module.applyTheme();
  });
} else if (window.feeling.trendy) {
  getTheme("trendy").then((module) => {
    module.applyTheme();
  });
}
```

```
const getTheme = (themeName) => import(`./src/themes/${themeName}`);

// Using `import()` 'dynamically'
if (window.feeling.stylish) {
  getTheme("stylish").then((module) => {
    module.applyTheme();
  });
} else if (window.feeling.trendy) {
  getTheme("trendy").then((module) => {
    module.applyTheme();
  });
}
```

Loading an async bundle based on runtime conditions

Breakdown

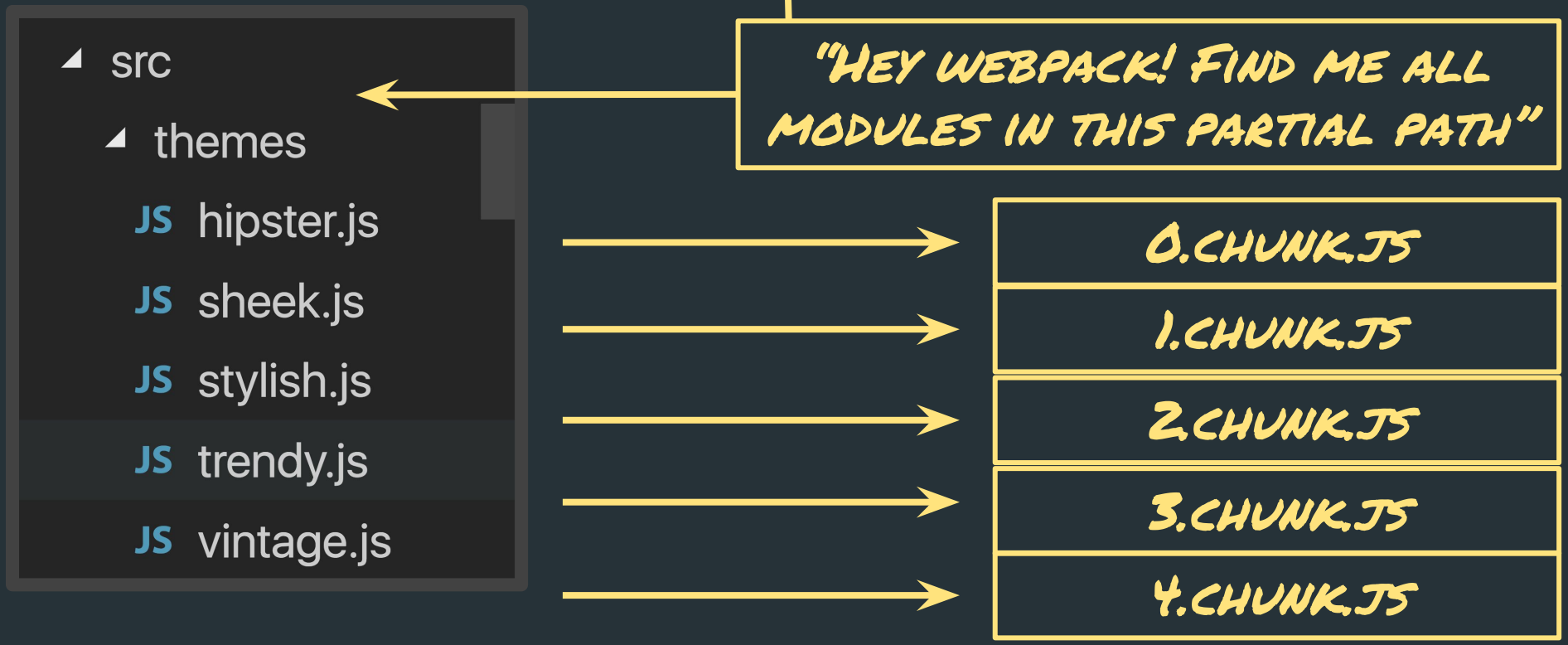
```
const getTheme = (themeName) =>
  import(`./src/themes/${themeName}`);
```

```
const getTheme = (themeName) =>
  import(`./src/themes/${themeName}`);
```

PARTIAL PATH	EXPRESSION
DIRECTORY	RESOLVABLE
CONTEXT	MODULE

CONTEXTMODULE!!!

```
const getTheme = (themeName) =>
  import(`./src/themes/${themeName}`);
```



When to use:

AB Testing

Theming

Convenience

Exercise Time

NOTE: ALWAYS FOCUS ON SPLITTING BEFORE CACHING

Perf Scenarios

HTTP/2

Service Worker

Progressive Web Applications

Performance Hints

Building For Node?

Should you?

Building For Node?

Should you?

Building for Electron?

Should you?

Building For Libs?

Should you?