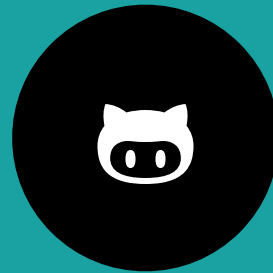


big pipe



3rdEden



3rd-Eden



arnout.kazemier

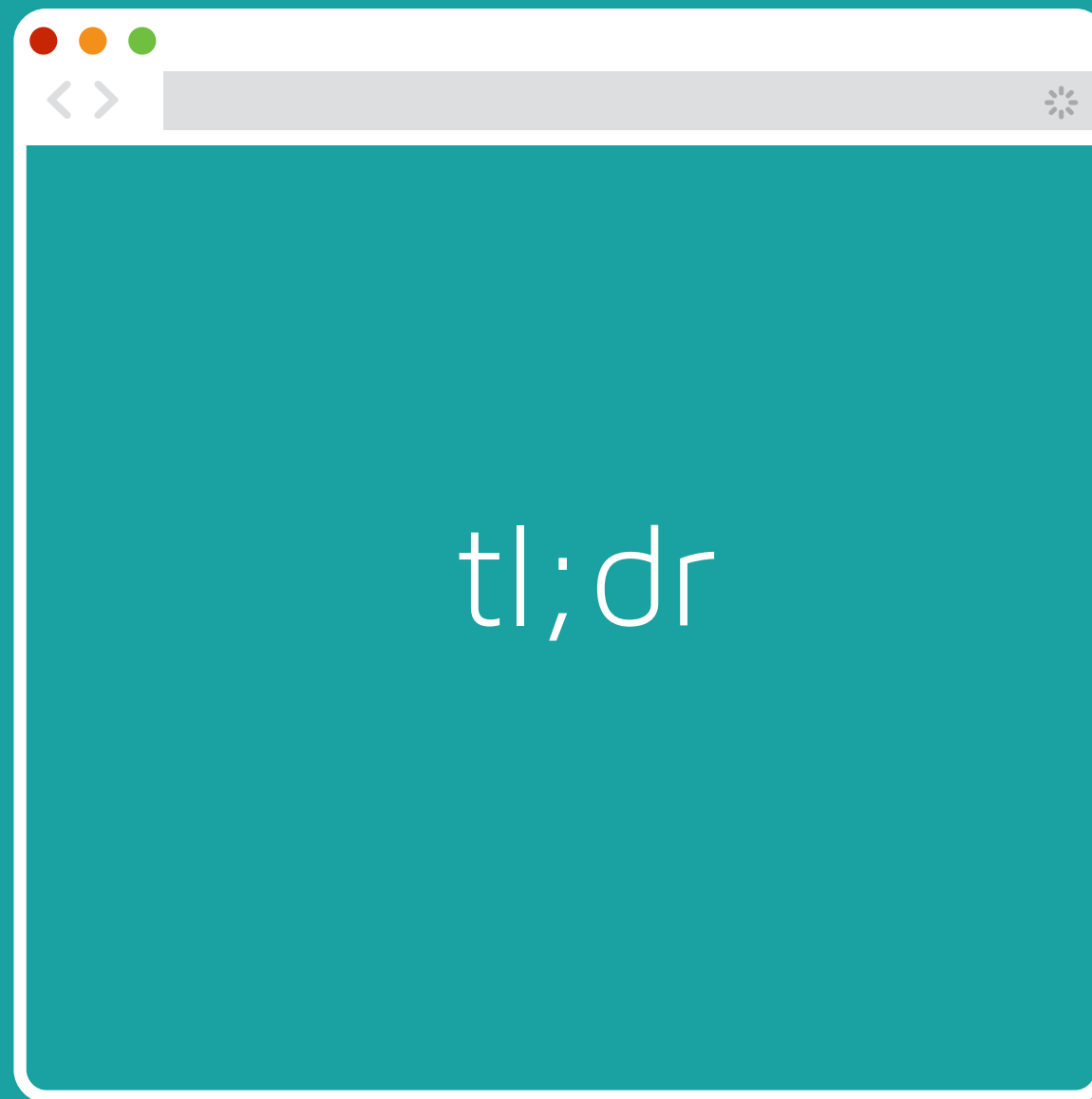
what is bigpipe

And why should you care

facebook

Introduced it on June 4, 2010







3rd-Eden

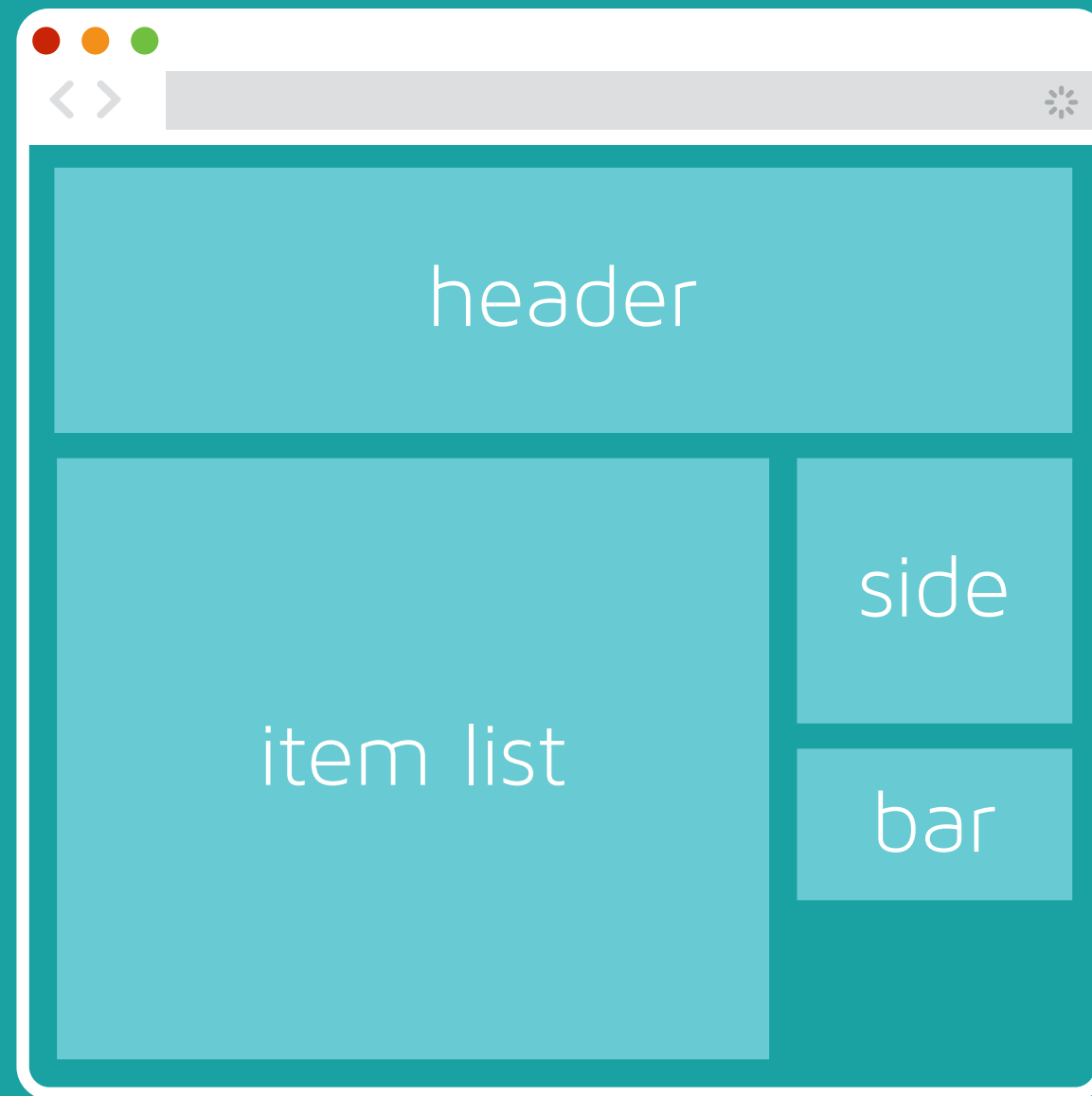
Arnout Kazemier

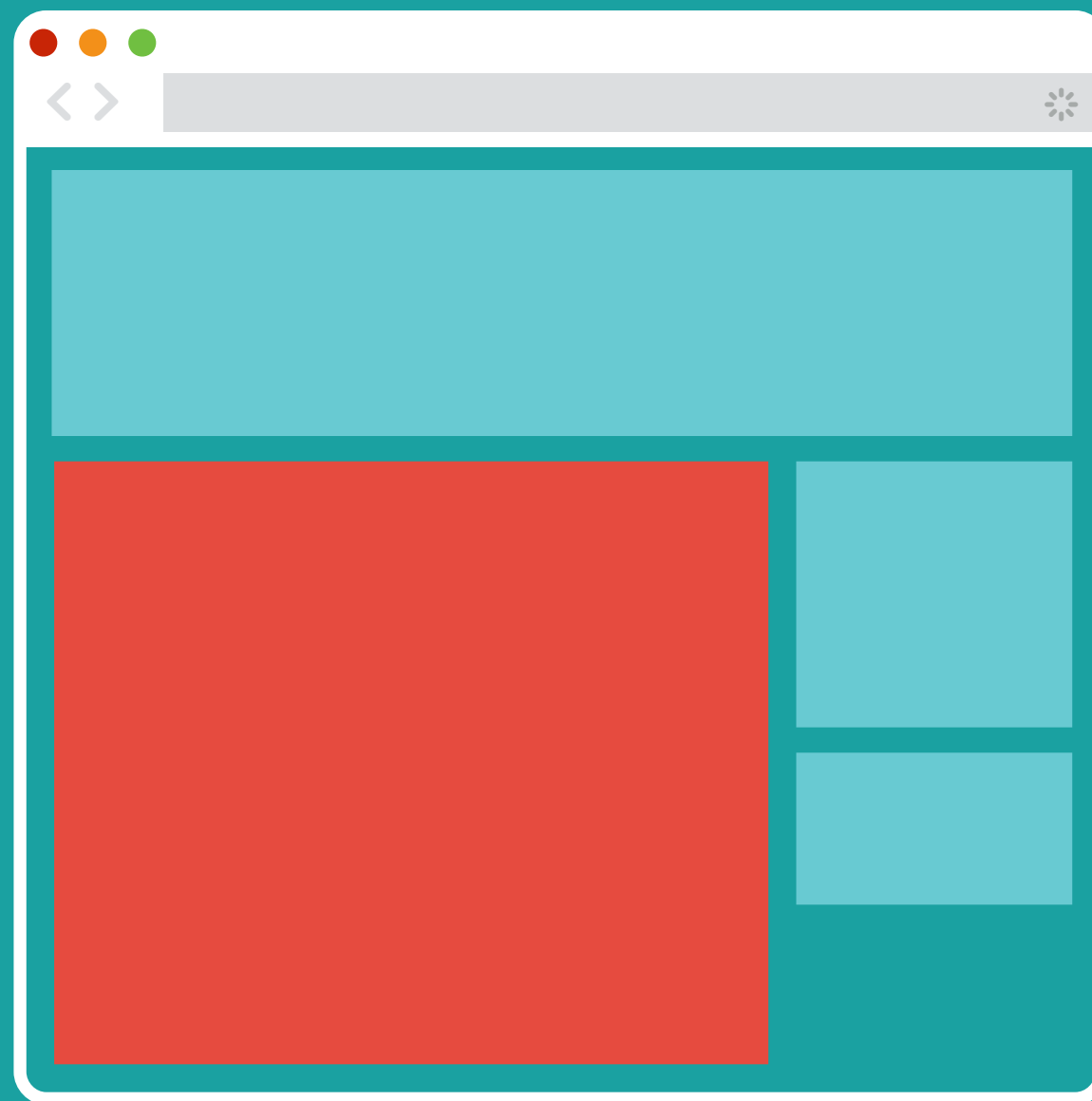


Swaagie

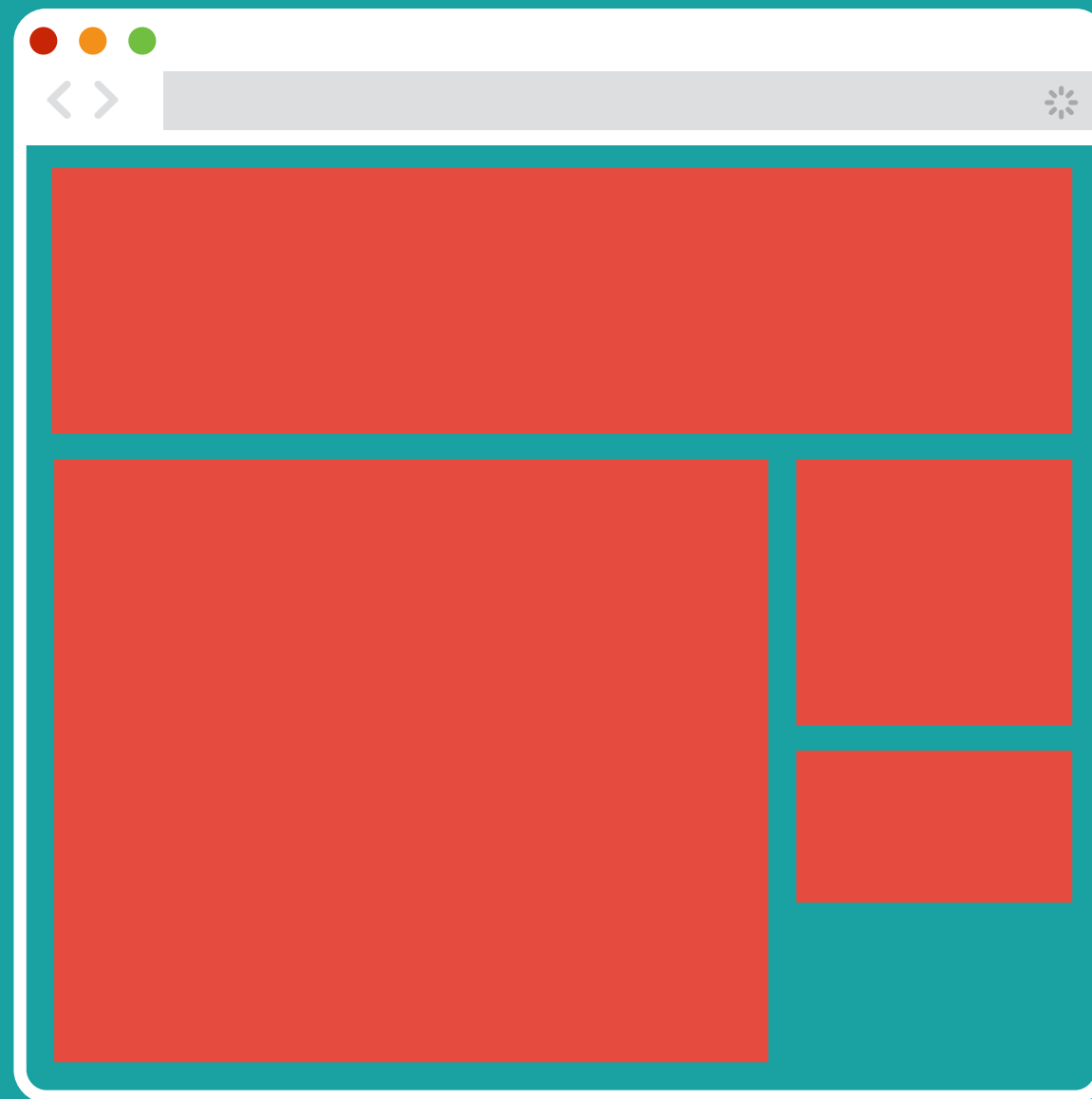
Martijn Swaagman







pagelet



page

```
1 <!DOCTYPE html><html class="no-js"><meta charset="UTF-8"><noscript><meta http-equiv="refresh" content="0; URL=/manage?no_pagelet_js=1"></noscript><link
rel="stylesheet href="https://versions.nodejitsu.com/css/jitsu.min.css" /><link rel="stylesheet href="//81bfe3c3da841e066e118f72eb92bc452465067.css" />
<script src="/b442ec74924ec474016223a187f91c309ea9c76.js"></script><script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js">
</script><script src="/acd076ba667d07ac40a198495e8ff408c159ef73.js"></script><script>pipe = new BigPipe(undefined,
{"pagelets":2,"id":"IOXWT78KIVS1NHFRKWSV999711H2RZFR06I7DMSIAYKTE290OZ3BQIQWS4W8KT9"} );</script><meta name="viewport" content="width=device-width,
initial-scale=1.0"><title>npm Registry Management - Nodejitsu</title><link rel="shortcut icon" href="//favicon.png"><script type="text/javascript"
src="//use.typekit.net/gzx4hto.js"></script><script type="text/javascript">try{Typekit.load();}catch(e){}</script>
2 </script></html><body class="container"><section class="navigation" id="navigation">
3   <nav class="row">
4     <h1>
5       <a href="/" class="logo" title="Back to the homepage">Nodejitsu</a>
6     </h1>
7     <a href="#navigation" class="mobile btn ss-row"></a>
8
9
10
11
12
13
14
15
16     <a href="/logout" class="login right btn" title="Logout">
17       <s class="ss-unlock" role="presentation"></s>
18       Logout
19     </a>
20
21 </nav>
22 </section>
23
24
25 <div data-pagelet="replication" class="replication"></div><section style="padding:20px 0" class="row"><article data-pagelet="packages" class="sixcol
product box"></article><article data-pagelet="users" class="sixcol last product box"></article></section><footer class="doormat">
26 <section class=" row">
27
28   <dl class="twocol">
29     <dt>Private npm registry</dt>
30
31     <dd>
32       <a href="mailto:support@nodejitsu.com">Support</a>
33     </dd>
34
35     <dd>
36       <a href="https://nodejitsu.uservoice.com">Feedback</a>
37     </dd>
38   </dl>
39
40
41
42   <dl class="newsletter twocol right">
43
44     <dd>
```

<html>

<head>

<meta charset=“utf-8”>

<script src=“{bigpipe.js}”></script>

<noscript>

<meta http-equiv=“refresh” content=“/?no_pagelet_js=1”>

</noscript>

...

</head>

<body>

...

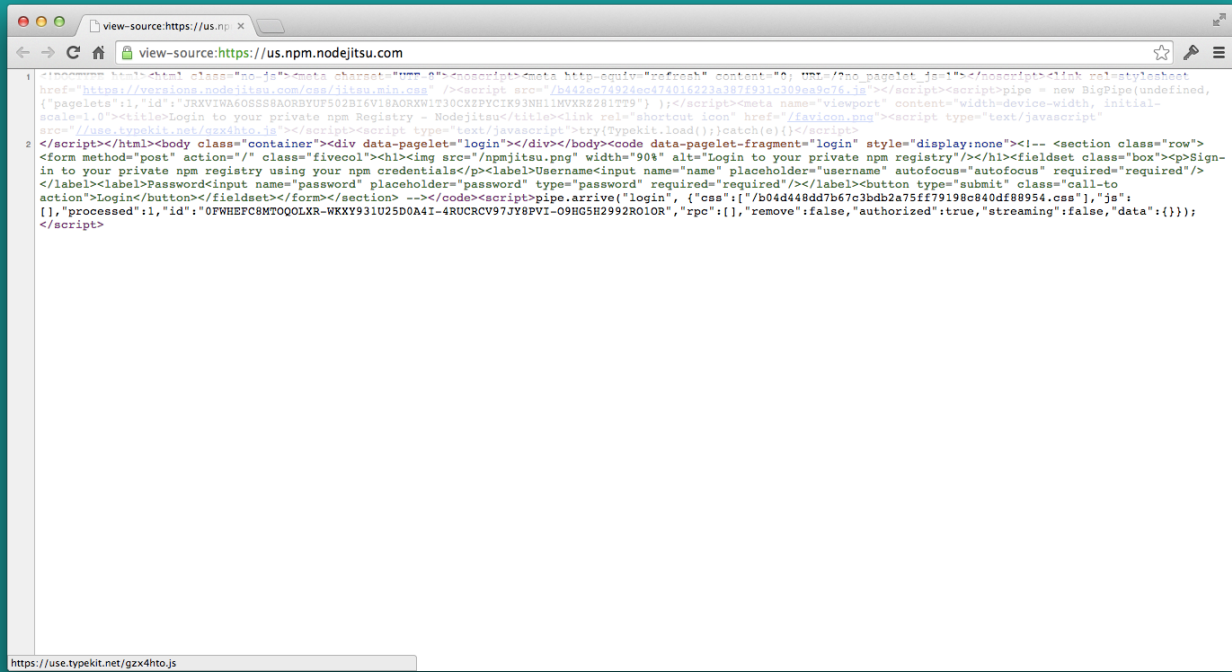
<header data-pagelet=“header”></header>

<section data-pagelet=“feed”></section>

<footer data-pagelet=“recommended”></section>

...

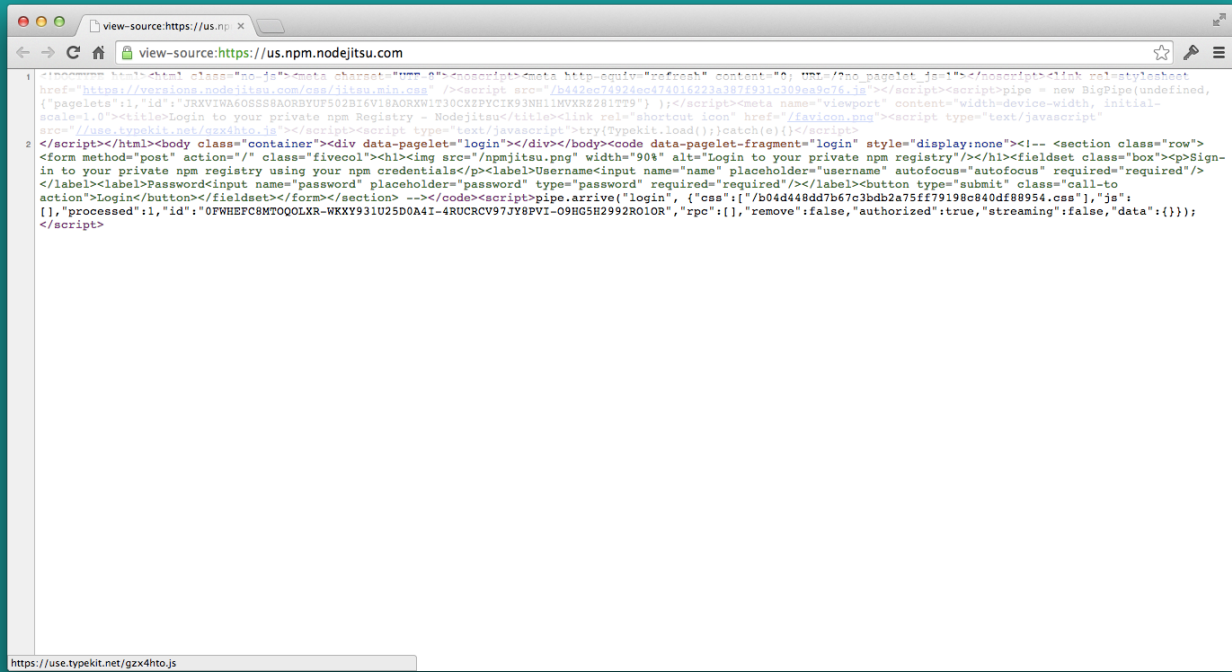
</body>



```
<code data-pagelet-fragment="login" style="display:none">
<!-- <section class="row"><form method="post" action="/"
class="fivecol"><h1>
```

```
..
</form></section> --></code>
```

```
<script>
  pipe.arrive("login", {
    "css":["/b04d448dd7b67c3bdb2a75ff79198c840df88954.css"],
    "js":[],
    "processed":1,
    "id":"0FWH-EFC8MTOQ-OLXRO9HG5H2-992RO1OR",
    "rpc":[],
    "remove":false,
    "authorized":true,
    "streaming":false,
    "data":{}
  });
</script>
```



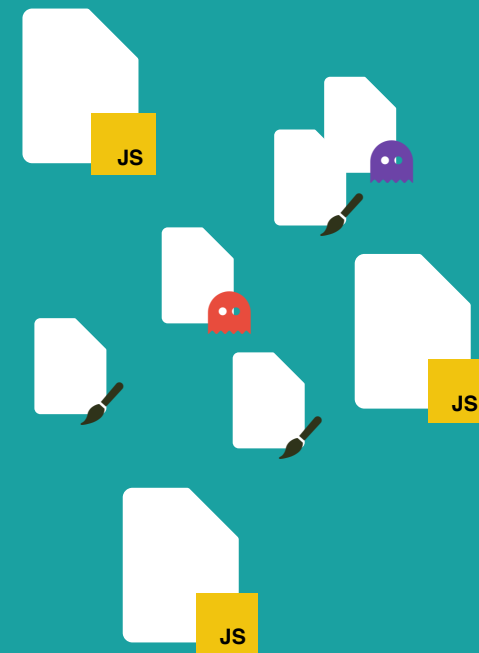
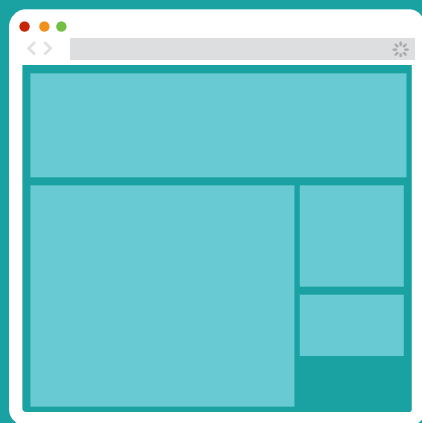
```
<code data-pagelet-fragment="login" style="display:none">
<!-- <section class="row"><form method="post" action="/"
class="fivecol"><h1>
```

```
..
</form></section> --></code>
```

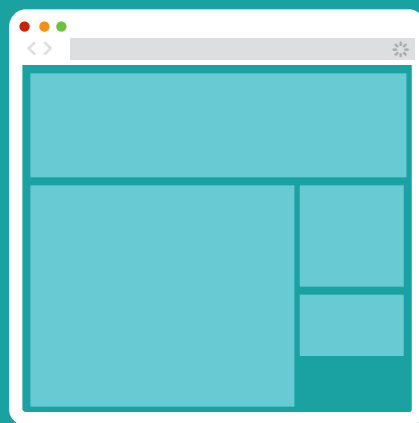
```
<script>
  pipe.arrive("login", {
    "css":["/b04d448dd7b67c3bdb2a75ff79198c840df88954.css"],
    "js":[],
    "processed":1,
    "id":"0FWH-EFC8MTOQ-OLXRO9HG5H2-992RO1OR",
    "rpc":[],
    "remove":false,
    "authorized":true,
    "streaming":false,
    "data":{}
  });
</script>
```

why

What do we gain by adopting this pattern



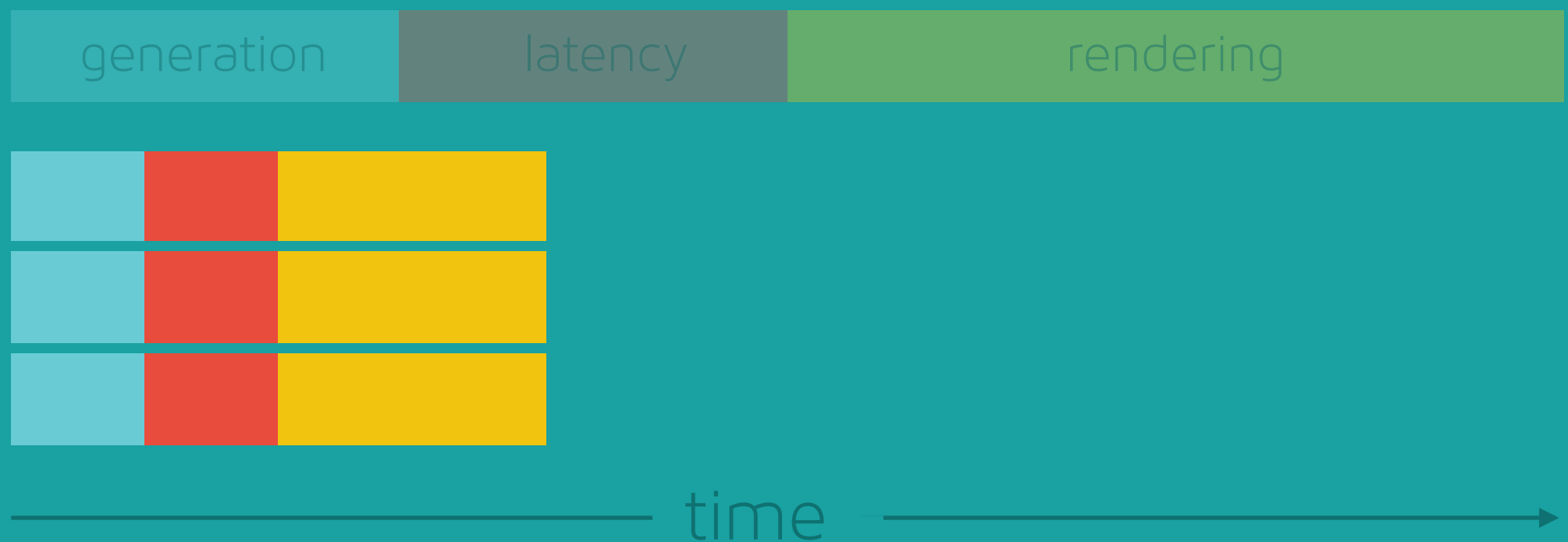
front-end/client & render performance



optimal usage/performance



improved perceived performance & time to interact



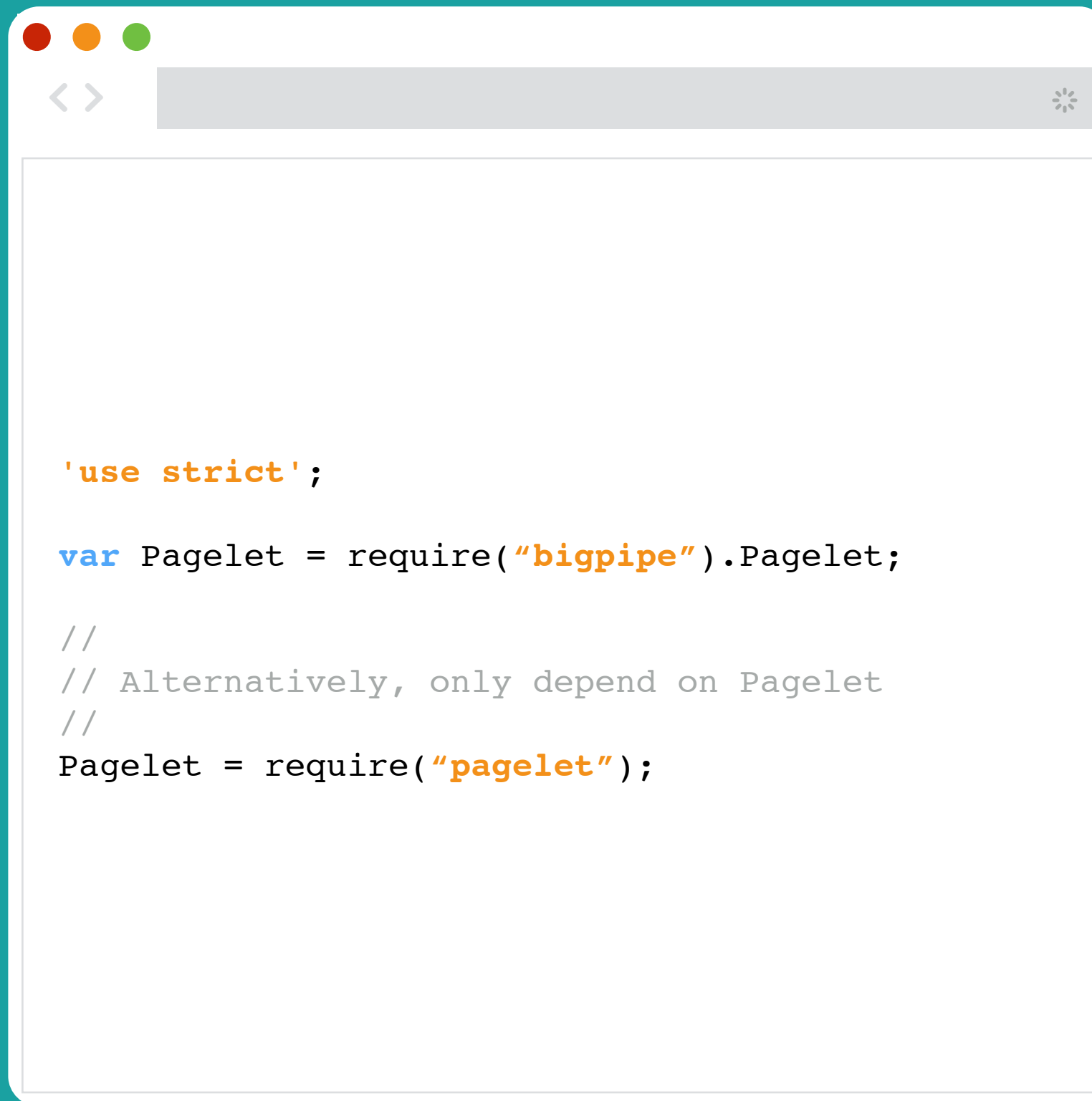
perceived performance & time to interact



developer satisfaction

pagelet

A small re-usable fragment which contains HTML, CSS and JavaScript

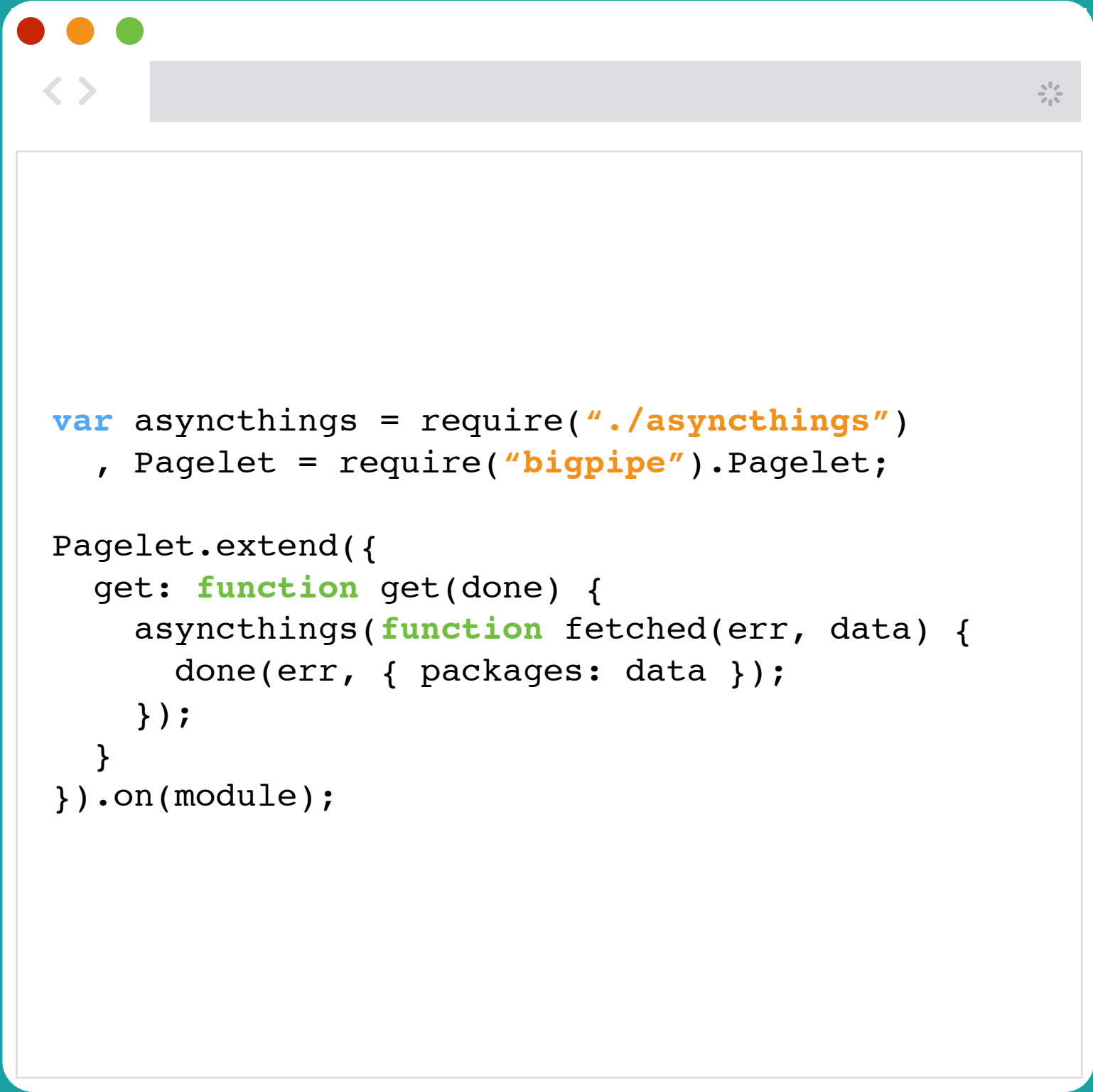


```
'use strict';

var Pagelet = require("bigpipe").Pagelet;

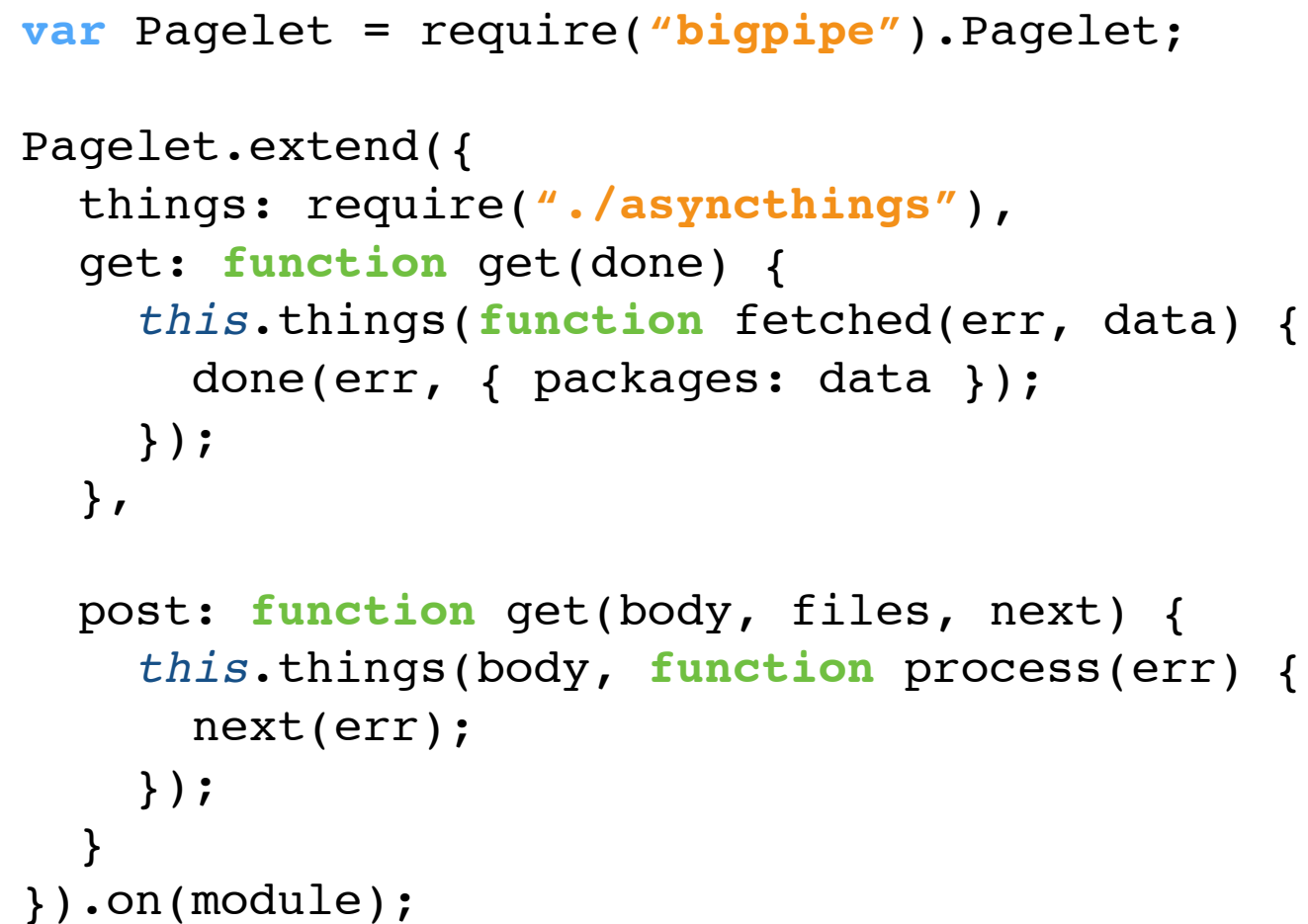
//
// Alternatively, only depend on Pagelet
//
Pagelet = require("pagelet");
```





```
var asyncthings = require("./asyncthings")
  , Pagelet = require("bigpipe").Pagelet;

Pagelet.extend({
  get: function get(done) {
    asyncthings(function fetched(err, data) {
      done(err, { packages: data });
    });
  }
}).on(module);
```



```
var Pagelet = require("bigpipe").Pagelet;

Pagelet.extend({
  things: require("./asyncthings"),
  get: function get(done) {
    this.things(function fetched(err, data) {
      done(err, { packages: data });
    });
  },

  post: function get(body, files, next) {
    this.things(body, function process(err) {
      next(err);
    });
  }
}).on(module);
```

views

Multiple built-in template engines supported

```
$ npm install --save ejs
```

```
..
```

```
$ vim index.js
```

```
'use strict';
```

```
var Pagelet = require("bigpipe").Pagelet;
```

```
Pagelet.extend({  
  view: "sidebar.ejs"  
}).on(module);
```


views

Custom error template when
this particular pagelet fails.
Can be a different language.

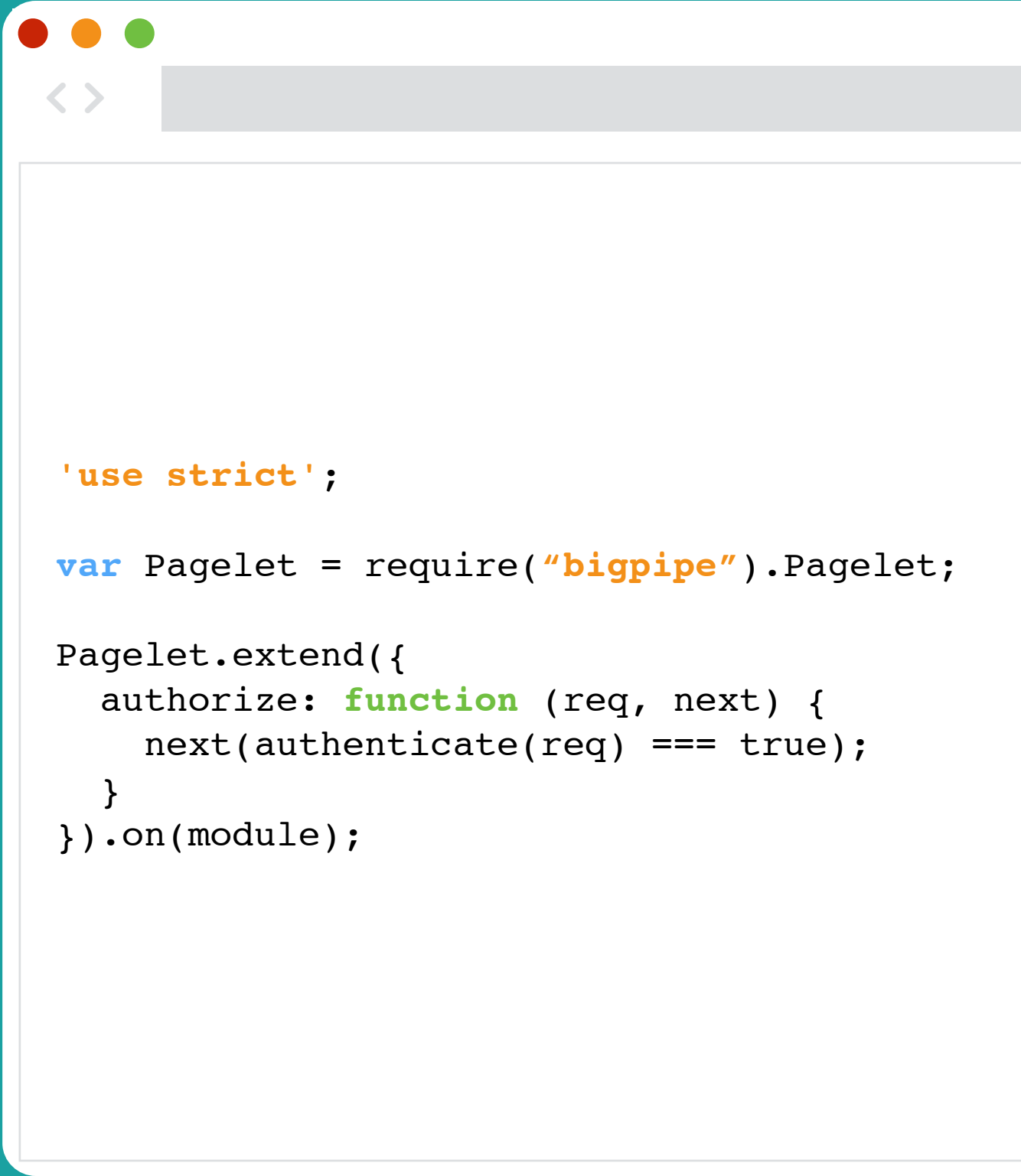
```
'use strict';

var Pagelet = require("bigpipe").Pagelet;

Pagelet.extend({
  view: "sidebar.ejs",
  error: "unavailable.jade"
}).on(module);
```

auth

Pagelet only shown when
authorized



```
'use strict';

var Pagelet = require("bigpipe").Pagelet;

Pagelet.extend({
  authorize: function (req, next) {
    next(authenticate(req) === true);
  }
}).on(module);
```

auth

Don't remove the placeholder
from the DOM

```
'use strict';

var Pagelet = require("bigpipe").Pagelet;

Pagelet.extend({
  authorize: function (req, next) {
    next(authenticate(req) === true);
  },
  remove: false
}).on(module);
```

rpc

Full real-time remote
procedure calls

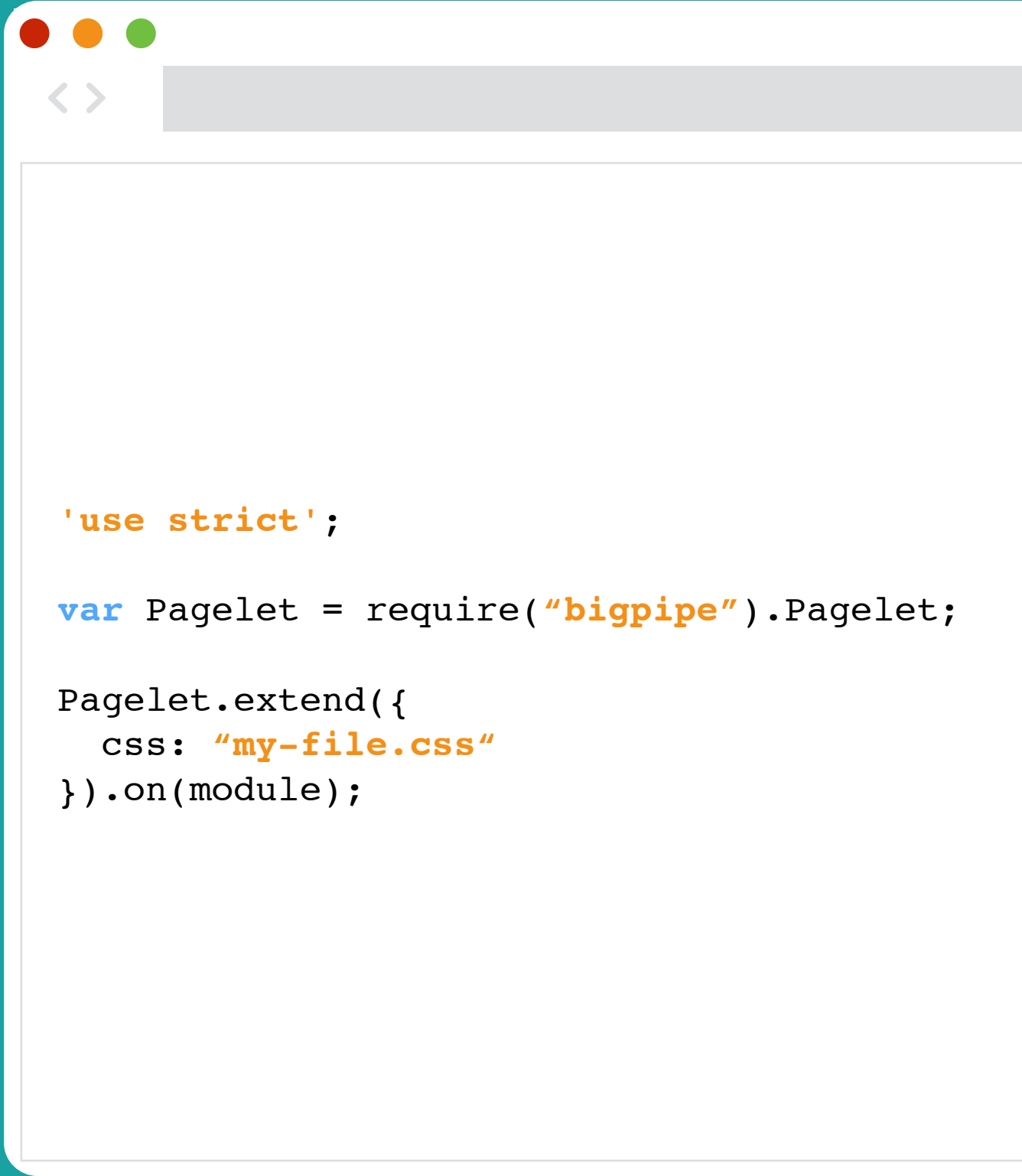
```
'use strict';

var Pagelet = require("bigpipe").Pagelet
    , asyncthings = require("./asyncthings");

Pagelet.extend({
  RPC: ["search"],
  search: function search(reply, arg) {
    asyncthings(function (err, data) {
      reply(err, data);
    });
  }
}).on(module);
```

styling

Name-spacing, pre-processing all happening transparently



```
'use strict';  
  
var Pagelet = require("bigpipe").Pagelet;  
  
Pagelet.extend({  
  css: "my-file.css"  
}).on(module);
```

styling

Name-spacing, pre-processing all happening transparently

```
ul
  &.foo
    list-style: none
    padding: 0
```

```
[data-pagelet="sidebar"] ul.foo {
  list-style: none;
  padding: 0;
}
```

distribute

Publish using npm

```
'use strict';

var Pagelet = require("bigpipe").Pagelet;

Pagelet.extend({
  view: "view.jade",
  css: "my-file.sass",
  get: function get(done) {
    setTimeout(function () {
      done(undefined, { packages: 1 });
    }, 10);
  }
}).on(module);
```

```
$ npm publish packages-pagelet
```

distribute

Share using npm

```
$ npm install packages-pagelet
```

```
'use strict';
```

```
var Page = require("bigpipe").Page;
```

```
Page.extend({  
  path: "/packages/:name",  
  pagelets: {  
    "packages": require("packages-pagelet")  
  }  
}).on(module);
```


distribute

And remix existing pagelets

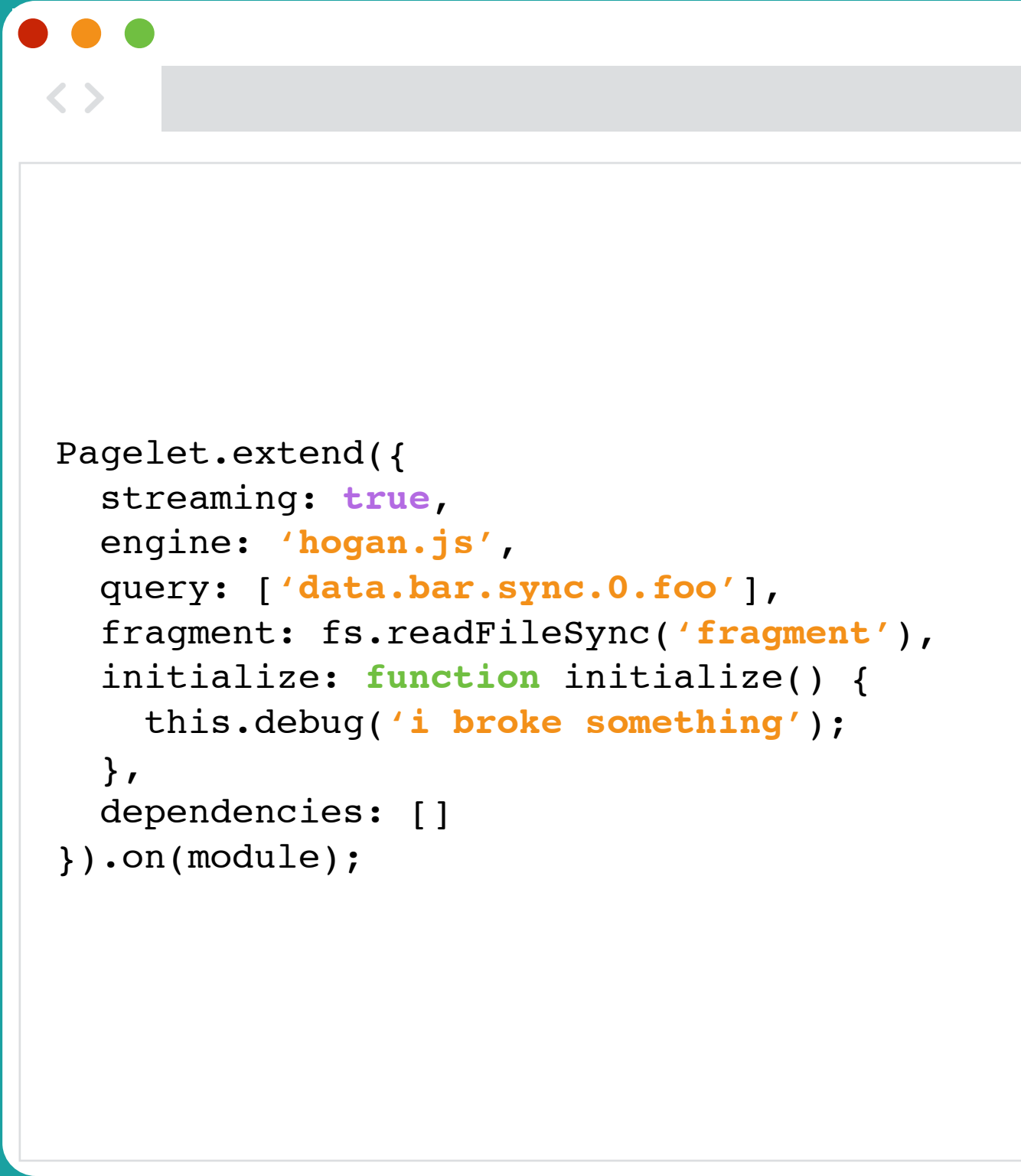
```
$ npm install packages-pagelet
```

```
'use strict';
```

```
var path = require("path")
    , Pagelet = require('packages-pagelet');

//
// Manually export the re-mixed instance
// which has the new custom css.
//
module.exports = Pagelet.extend({
  css: path.join(__dirname, 'custom.styl')
});
```

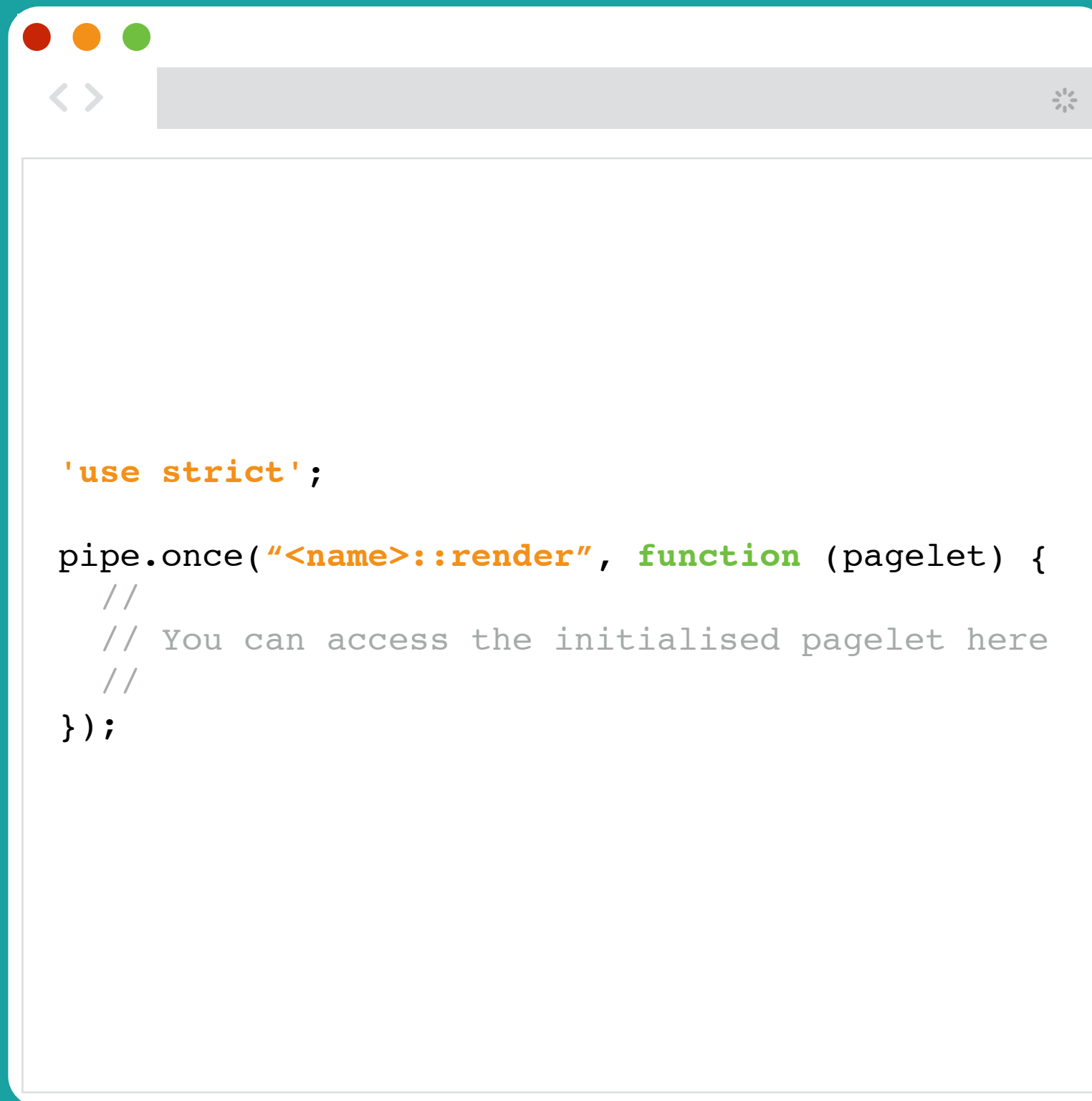
much
much
more



```
Pagelet.extend({  
  streaming: true,  
  engine: 'hogan.js',  
  query: ['data.bar.sync.0.foo'],  
  fragment: fs.readFileSync('fragment'),  
  initialize: function initialize() {  
    this.debug('i broke something');  
  },  
  dependencies: []  
}).on(module);
```

client

Minimal, transparent framework

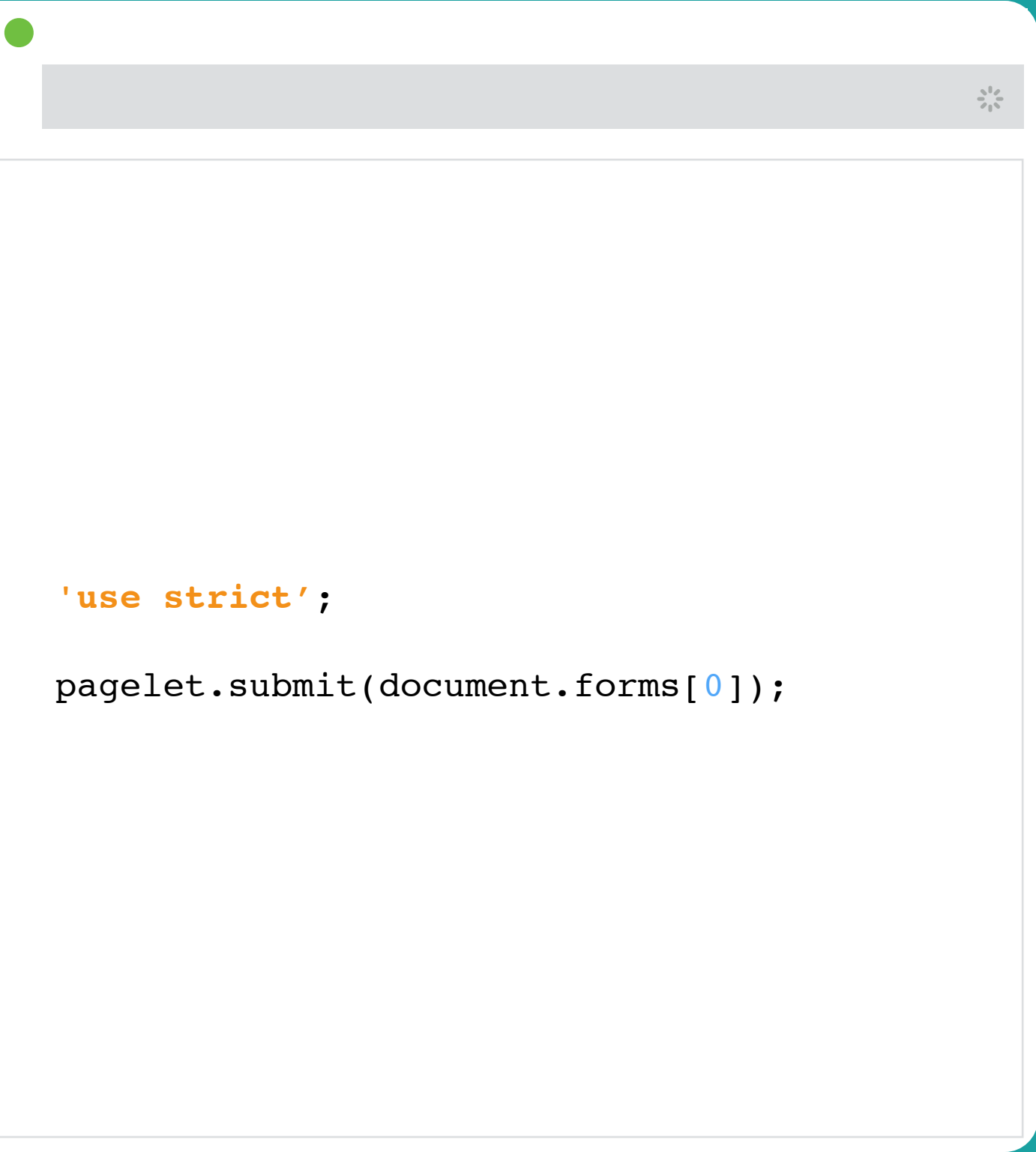




```
'use strict';  
pagelet.render( '<strong>hi</strong>' );
```

render

Refresh the HTML of every
paglet



```
'use strict';  
pagelet.submit(document.forms[0]);
```

submit

Submit a form over the real-time connection



```
'use strict';
```

```
pagelet.search('foo', function (err, row) {  
  console.log(err, row);  
});
```

rpc

Simply call the same method
name on the client

sandboxing

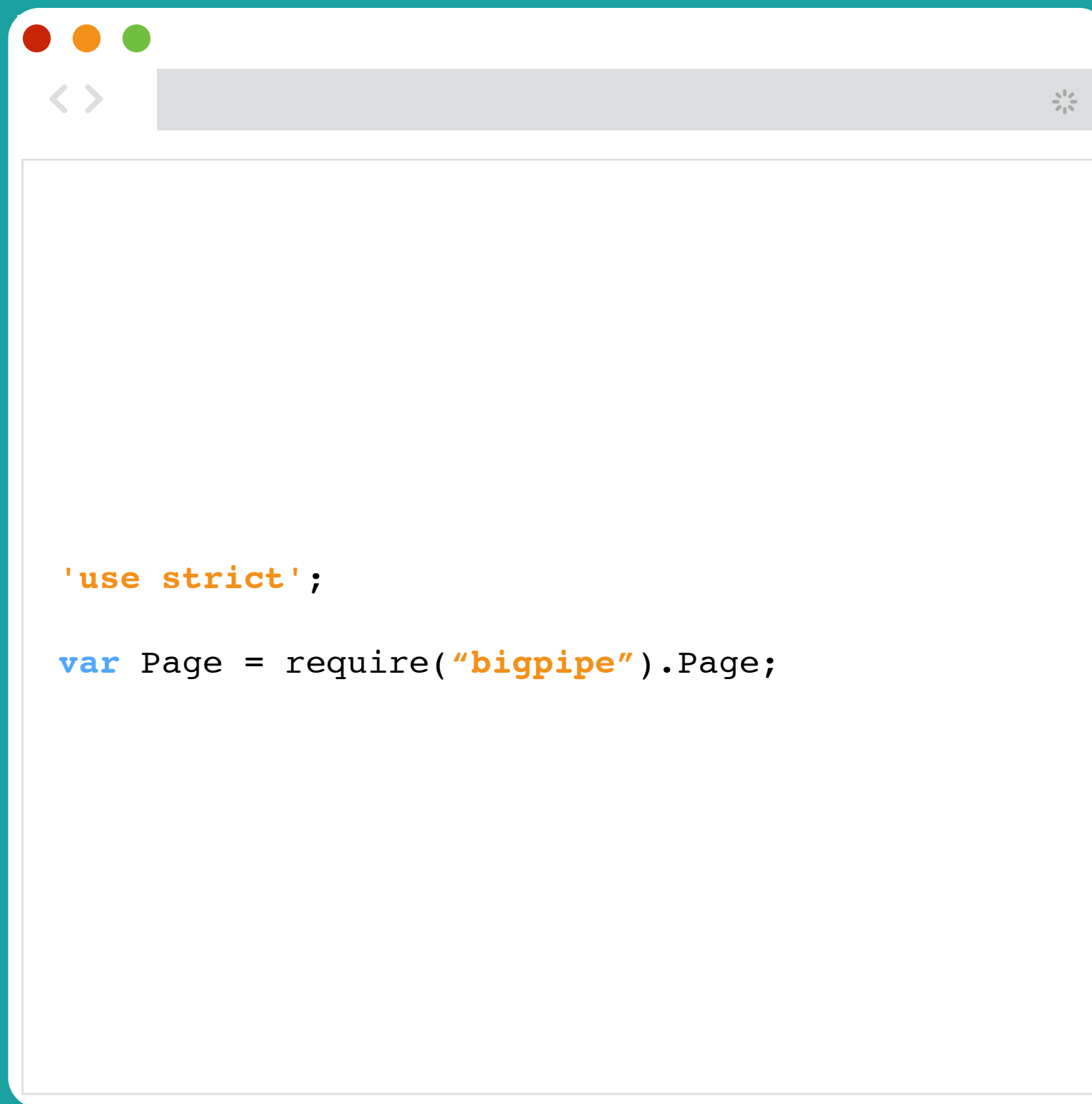
docker like containers for the browser

asset loading

async css & js

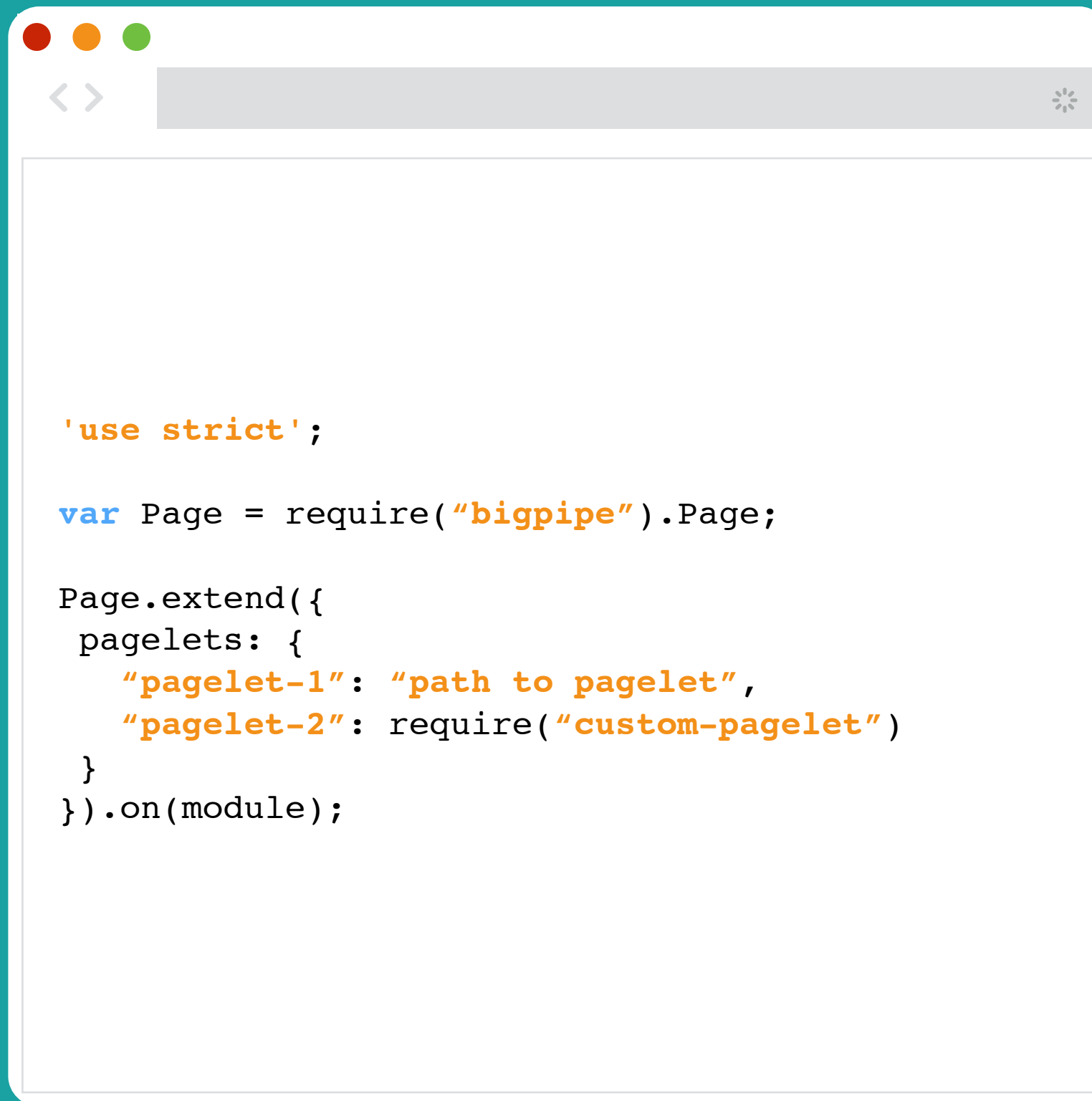
page

The entry point



```
'use strict';
```

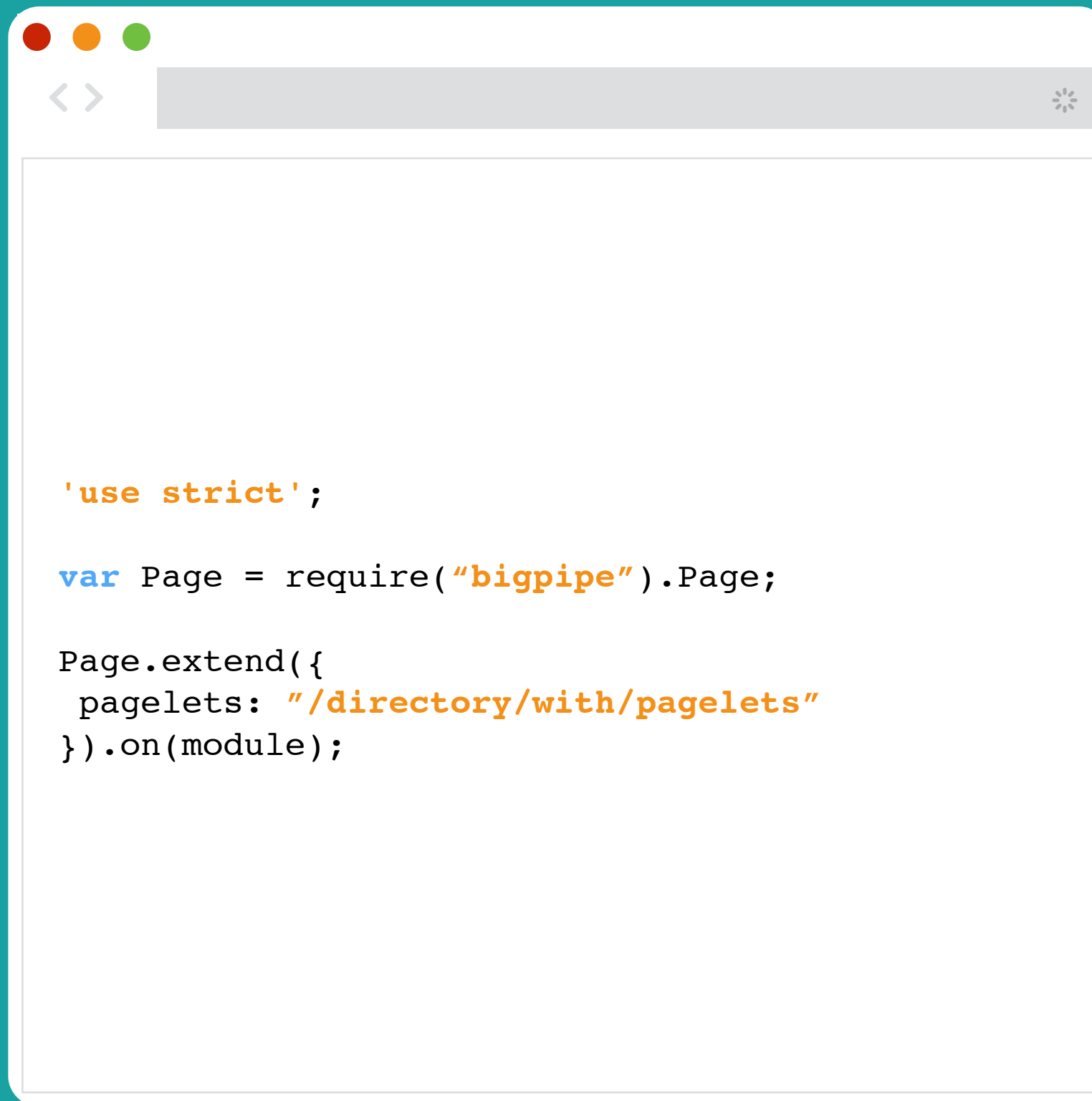
```
var Page = require("bigpipe").Page;
```

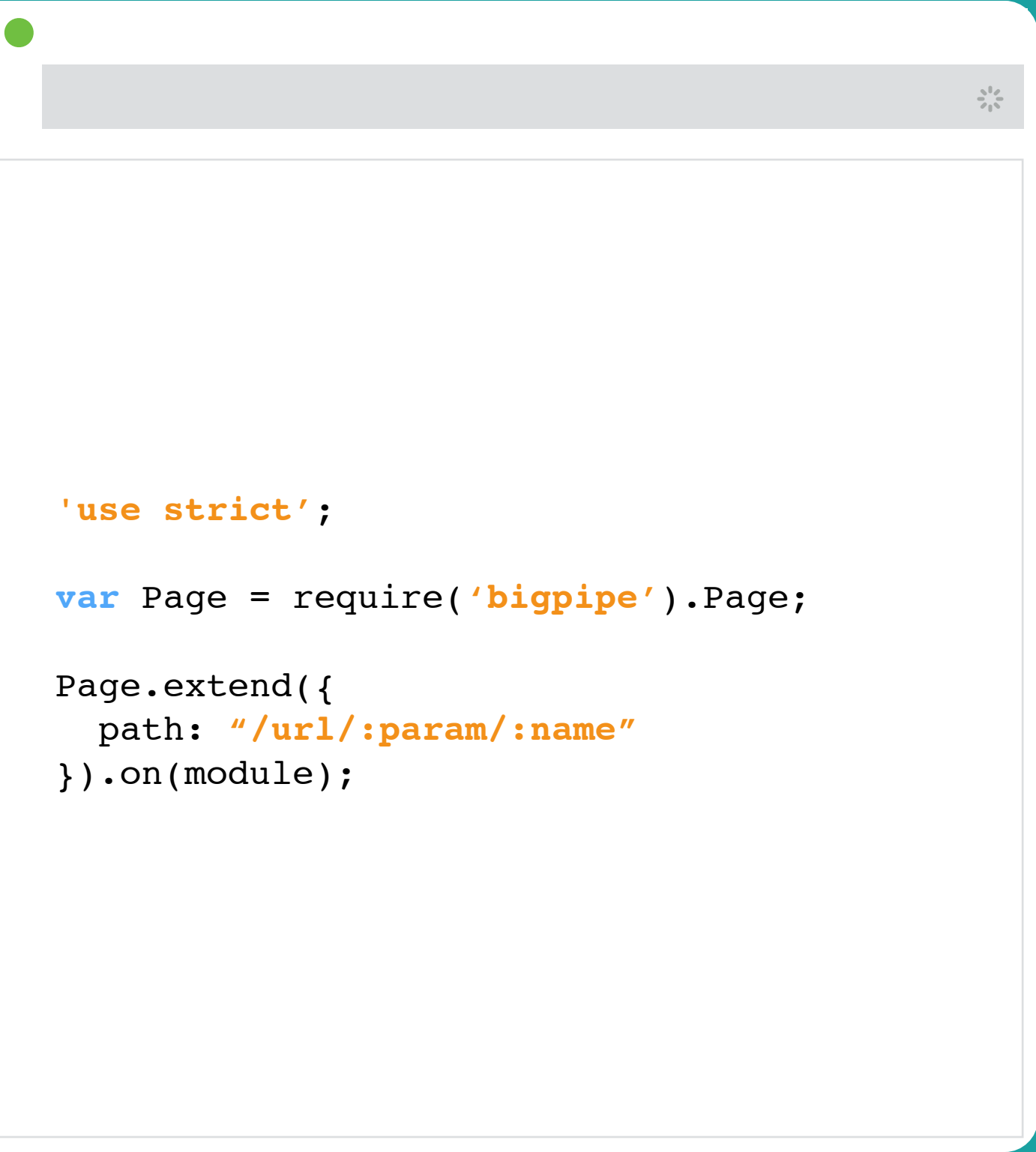


```
'use strict';

var Page = require("bigpipe").Page;

Page.extend({
  pagelets: {
    "pagelet-1": "path to pagelet",
    "pagelet-2": require("custom-pagelet")
  }
}).on(module);
```

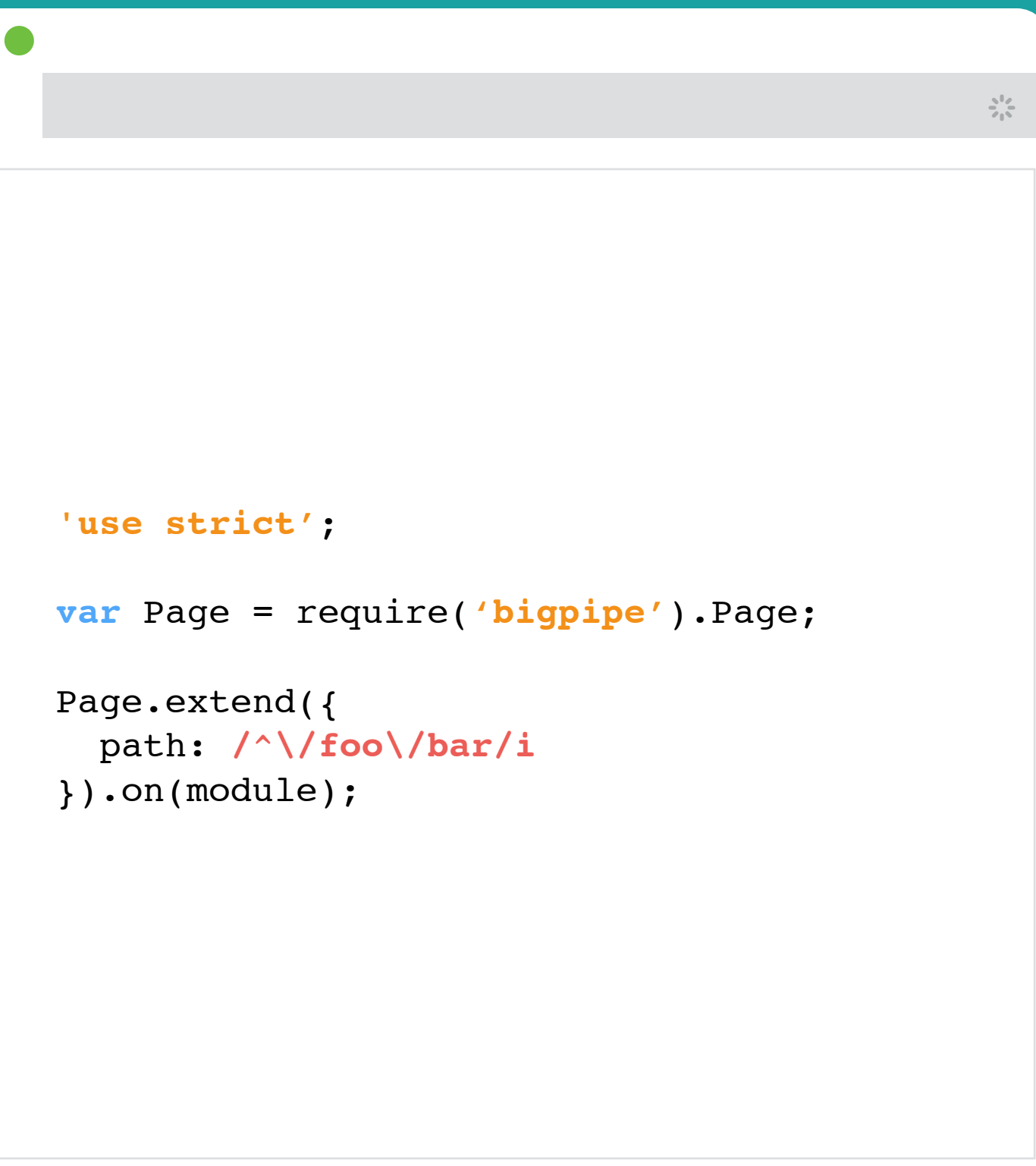




```
'use strict';  
  
var Page = require('bigpipe').Page;  
  
Page.extend({  
  path: "/url/:param/:name"  
}).on(module);
```

path

smart URL routing



```
'use strict';  
  
var Page = require('bigpipe').Page;  
  
Page.extend({  
  path: /^\/foo\/bar/i  
}).on(module);
```

path

regex based routing


```
'use strict';

var Page = require('bigpipe').Page;

Page.extend({
  path: '/^\\/(?<named>[\\d\\.]+)\\/foo/'
}).on(module);
```

path

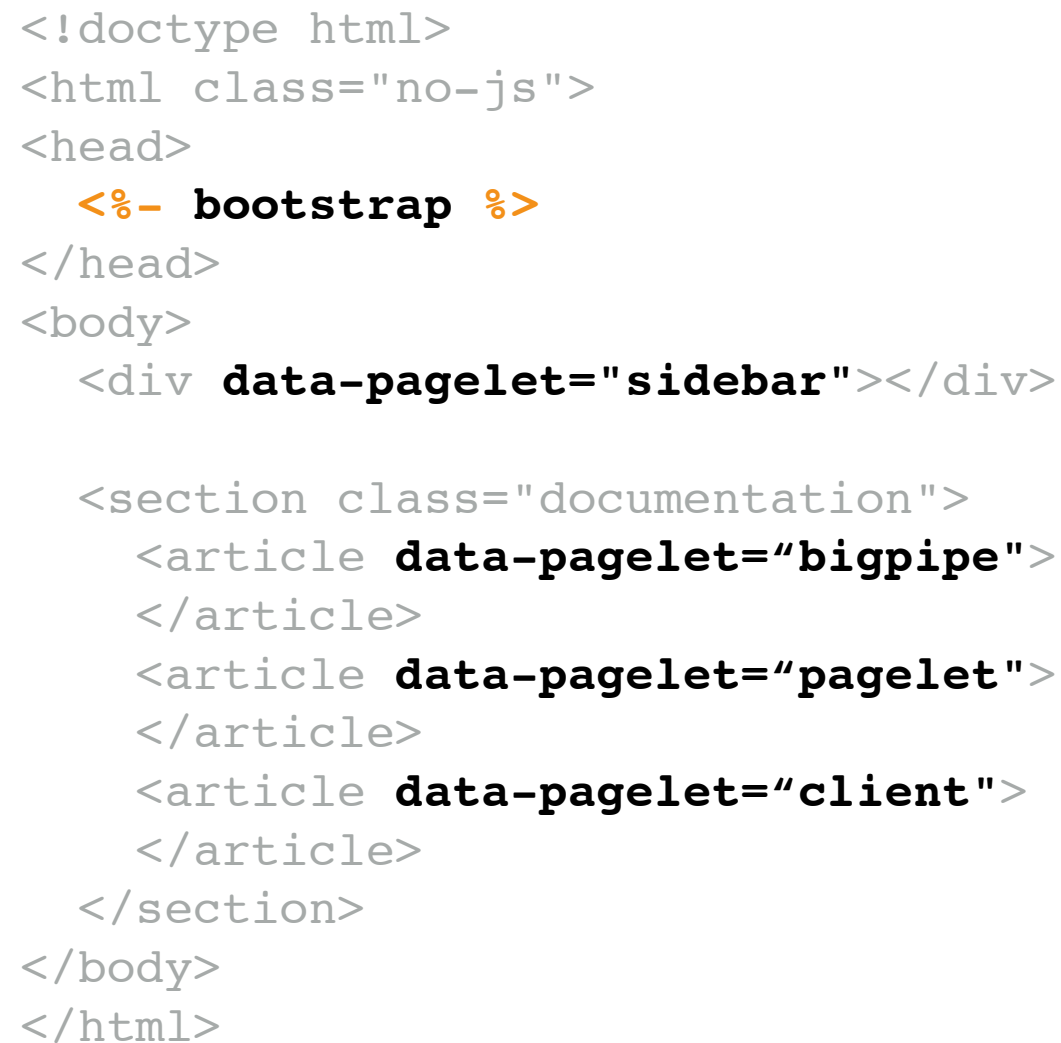
xRegExp name based routing



```
'use strict';  
  
var Page = require('bigpipe').Page;  
  
Page.extend({  
  path: "/",  
  view: "index.ejs"  
}).on(module);
```

view

The first chunk and
placeholders




```
<!doctype html>
<html class="no-js">
<head>
  <%- bootstrap %>
</head>
<body>
  <div data-pagelet="sidebar"></div>

  <section class="documentation">
    <article data-pagelet="bigpipe">
    </article>
    <article data-pagelet="pagelet">
    </article>
    <article data-pagelet="client">
    </article>
  </section>
</body>
</html>
```

view

Manually include the
{bootstrap} code



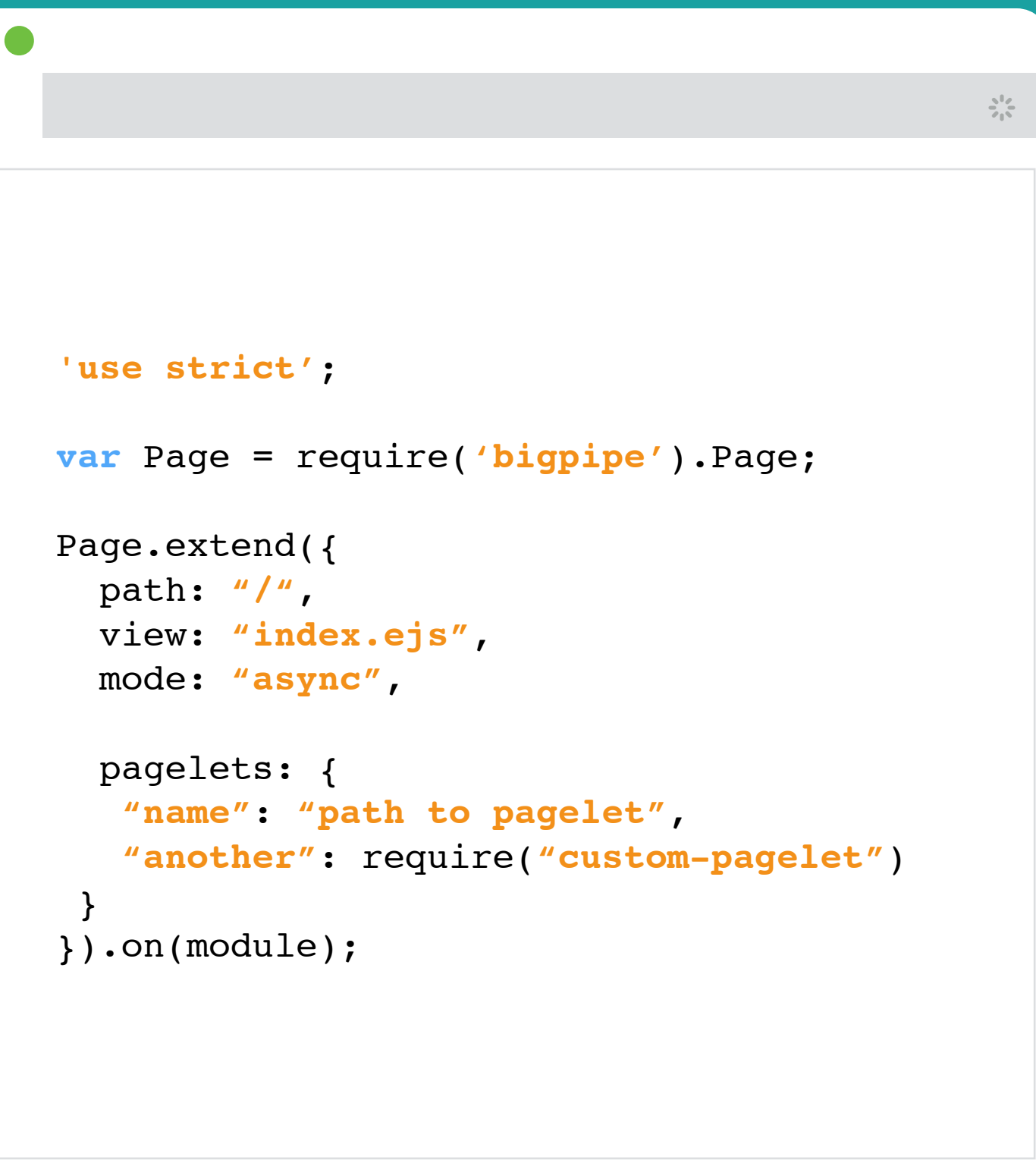
```
'use strict';

var Page = require('bigpipe').Page;

Page.extend({
  path: "/",
  view: "index.ejs",
  authorize: function auth(req, next) {
    next(false);
  }
}).on(module);
```

auth

Authorize access to pages.



```
'use strict';

var Page = require('bigpipe').Page;

Page.extend({
  path: "/",
  view: "index.ejs",
  mode: "async",

  pagelets: {
    "name": "path to pagelet",
    "another": require("custom-pagelet")
  }
}).on(module);
```

mode

Different pagelet render modes.

```
'use strict';

var Page = require('bigpipe').Page;

Page.extend({
  path: "/",
  view: "index.ejs",
  mode: "async",

  pagelets: {
    "name": "path to pagelet",
    "another": require("custom-pagelet")
  }
}).on(module);
```

mode

async: fully async, unordered rendering

```
'use strict';

var Page = require('bigpipe').Page;

Page.extend({
  path: "/",
  view: "index.ejs",
  mode: "pipeline",

  pagelets: {
    "name": "path to pagelet",
    "another": require("custom-pagelet")
  }
}).on(module);
```

mode

pipeline: render in the specified order.

```
'use strict';


var Page = require('bigpipe').Page;

Page.extend({
  path: "/",
  view: "index.ejs",
  mode: "sync",

  pagelets: {
    "name": "path to pagelet",
    "another": require("custom-pagelet")
  }
}).on(module);
```

mode

sync: single flush, no fancy
pancy rendering



```
'use strict';

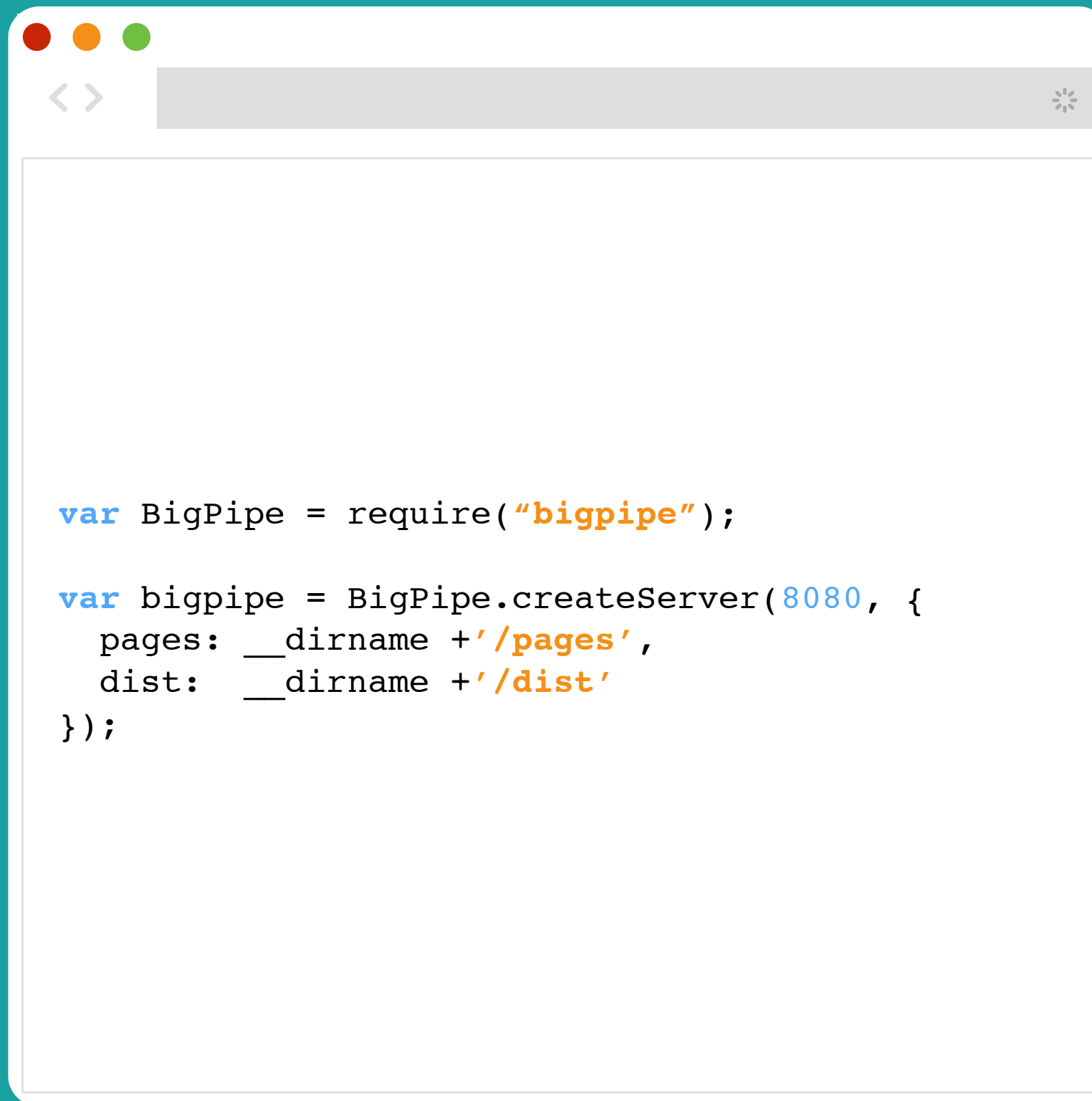
var Page = require('bigpipe').Page;

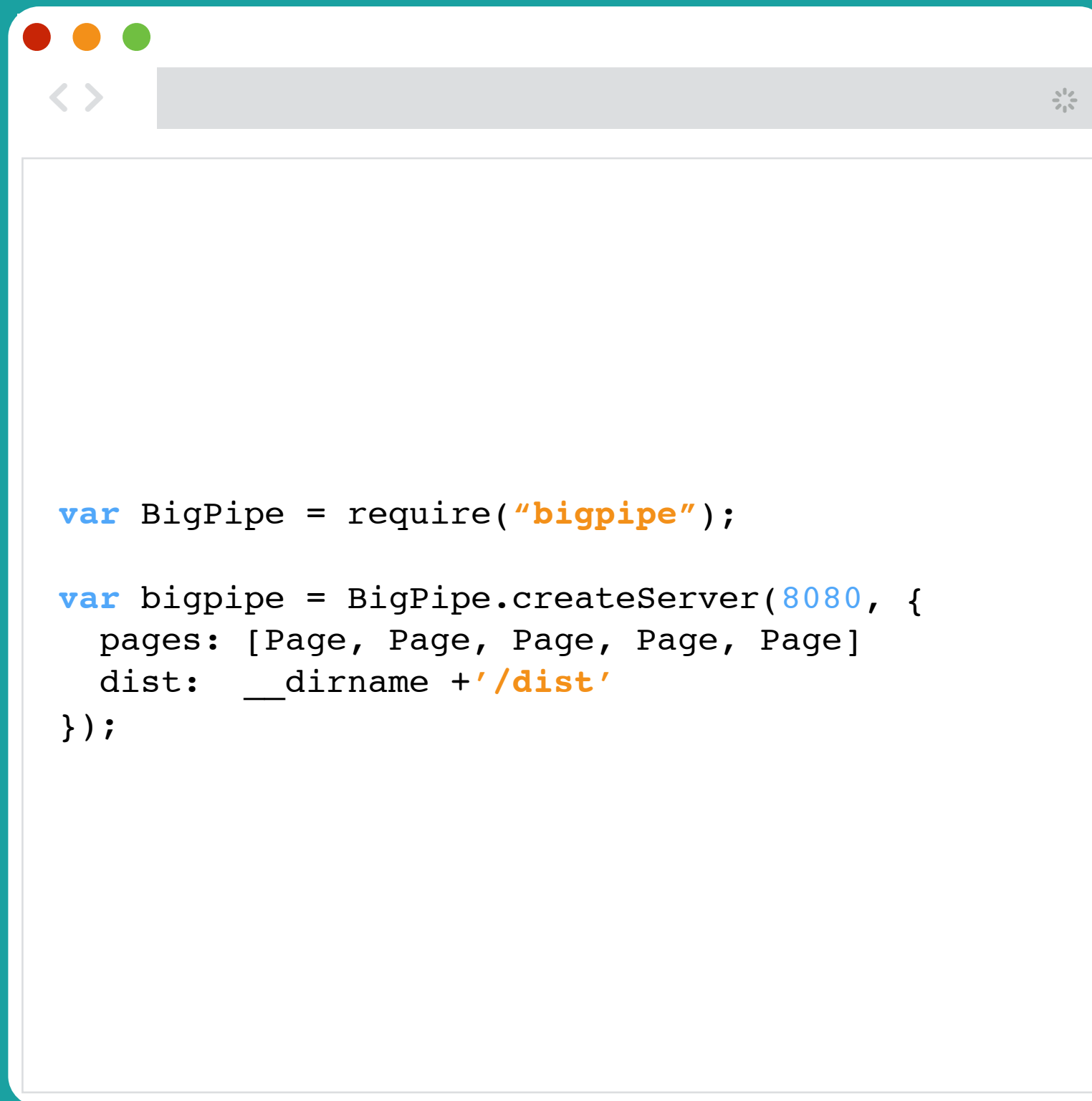
Page.extend({
  statusCode: 301,
  method: 'GET, POST, HEAD',
  contentType: 'custom/html',
  data: { template: 'data' },
  parsers: {},
  charset: 'UTF-7',
  dependencies: [],
  initialize: function () {}
}).on(module);
```

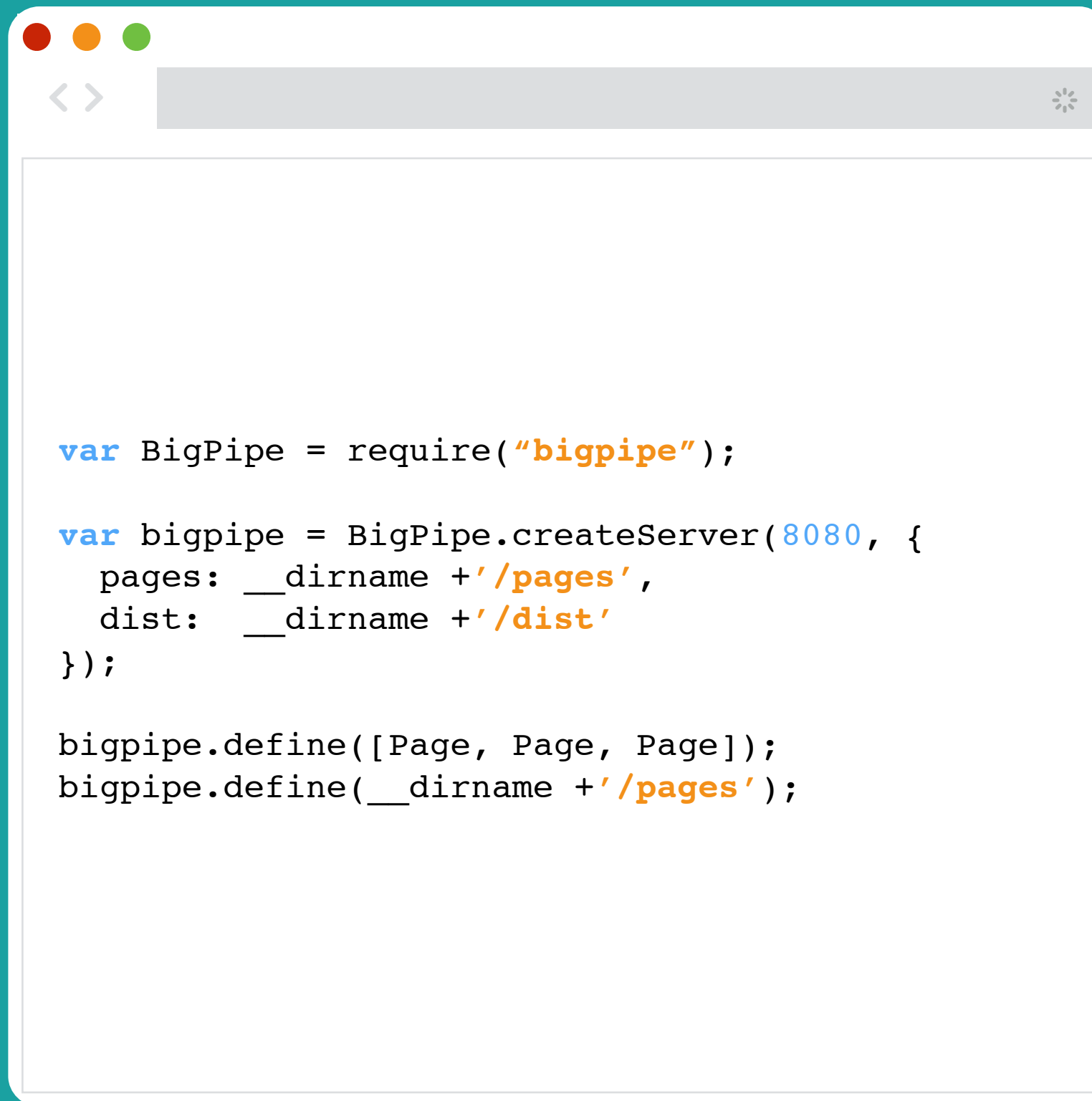
much
much
more

putting it all together

Creating your first BigPipe instance



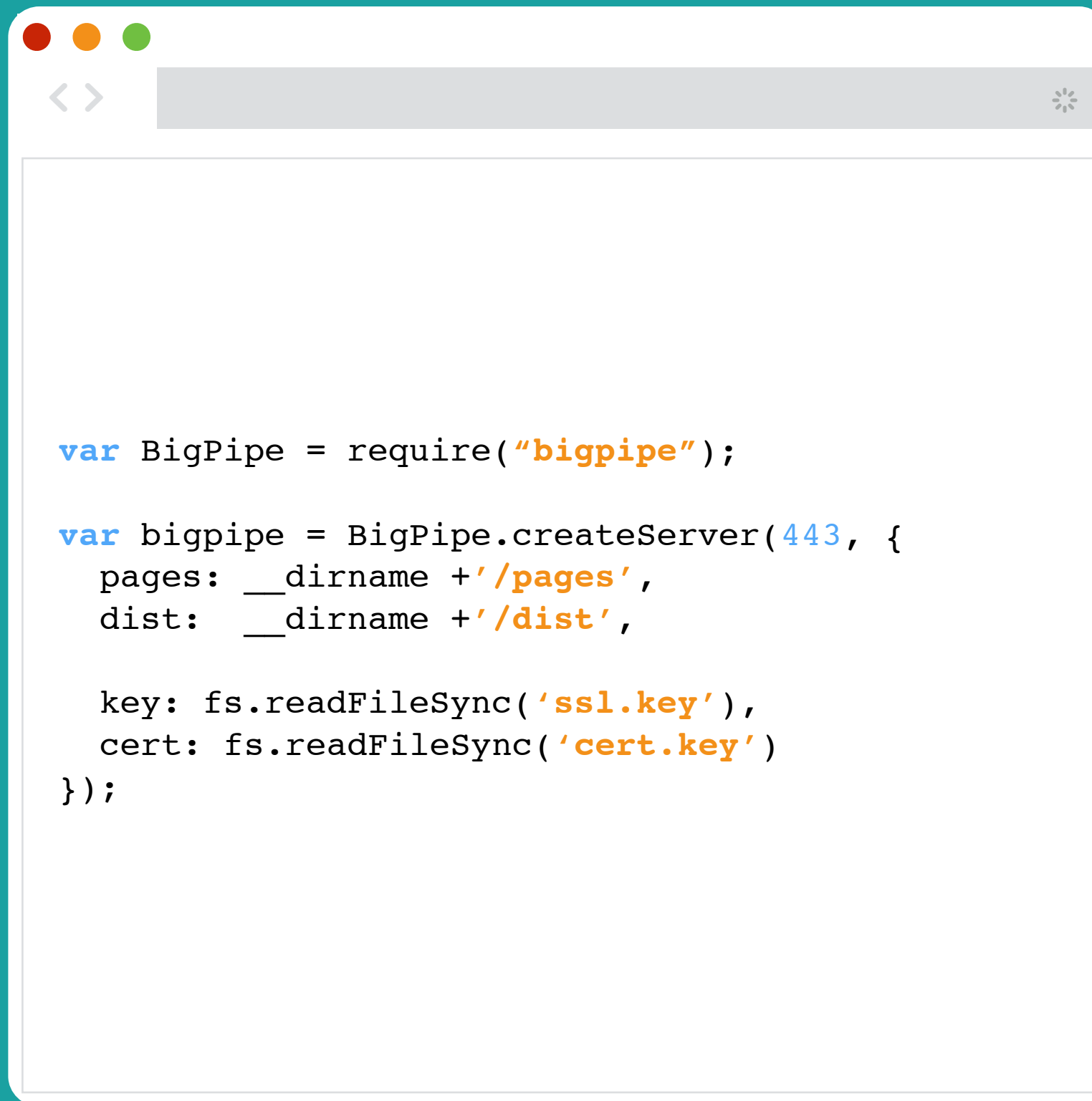




```
var BigPipe = require("bigpipe");

var bigpipe = BigPipe.createServer(8080, {
  pages: __dirname + '/pages',
  dist: __dirname + '/dist'
});

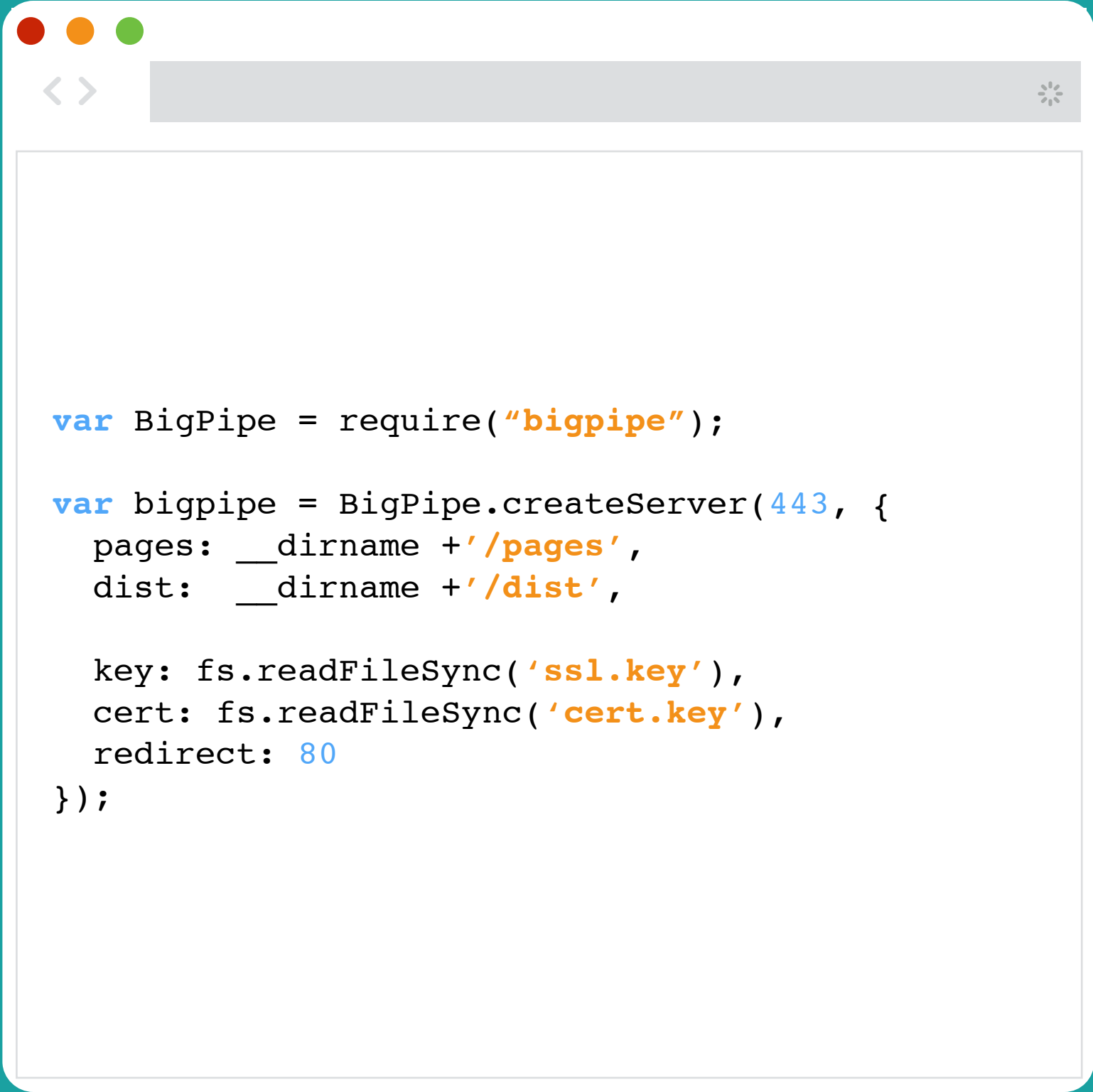
bigpipe.define([Page, Page, Page]);
bigpipe.define(__dirname + '/pages');
```



```
var BigPipe = require("bigpipe");

var bigpipe = BigPipe.createServer(443, {
  pages: __dirname + '/pages',
  dist:  __dirname + '/dist',

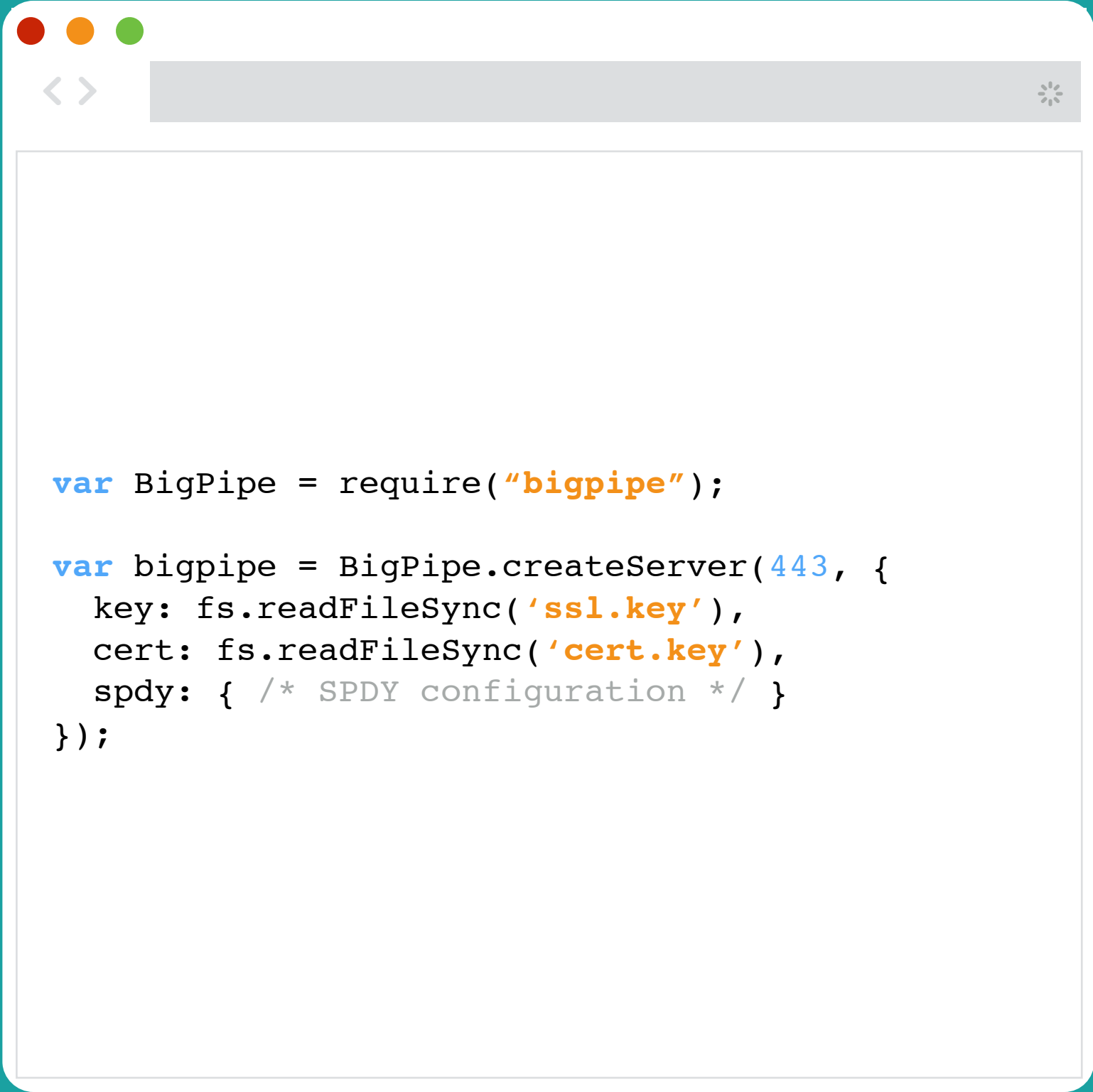
  key: fs.readFileSync('ssl.key'),
  cert: fs.readFileSync('cert.key')
});
```



```
var BigPipe = require("bigpipe");

var bigpipe = BigPipe.createServer(443, {
  pages: __dirname + '/pages',
  dist:  __dirname + '/dist',

  key: fs.readFileSync('ssl.key'),
  cert: fs.readFileSync('cert.key'),
  redirect: 80
});
```



```
var BigPipe = require("bigpipe");

var bigpipe = BigPipe.createServer(443, {
  key: fs.readFileSync('ssl.key'),
  cert: fs.readFileSync('cert.key'),
  spdy: { /* SPDY configuration */ }
});
```

```
var BigPipe = require("bigpipe");

var bigpipe = BigPipe.createServer(8080, {
  pages: __dirname + '/pages',
  dist: __dirname + '/dist',
  transformer: 'engine.io',
  parser: 'JSON'
});
```

primus


The transforming real-time
engine

primus

Why is so damn _awesome_

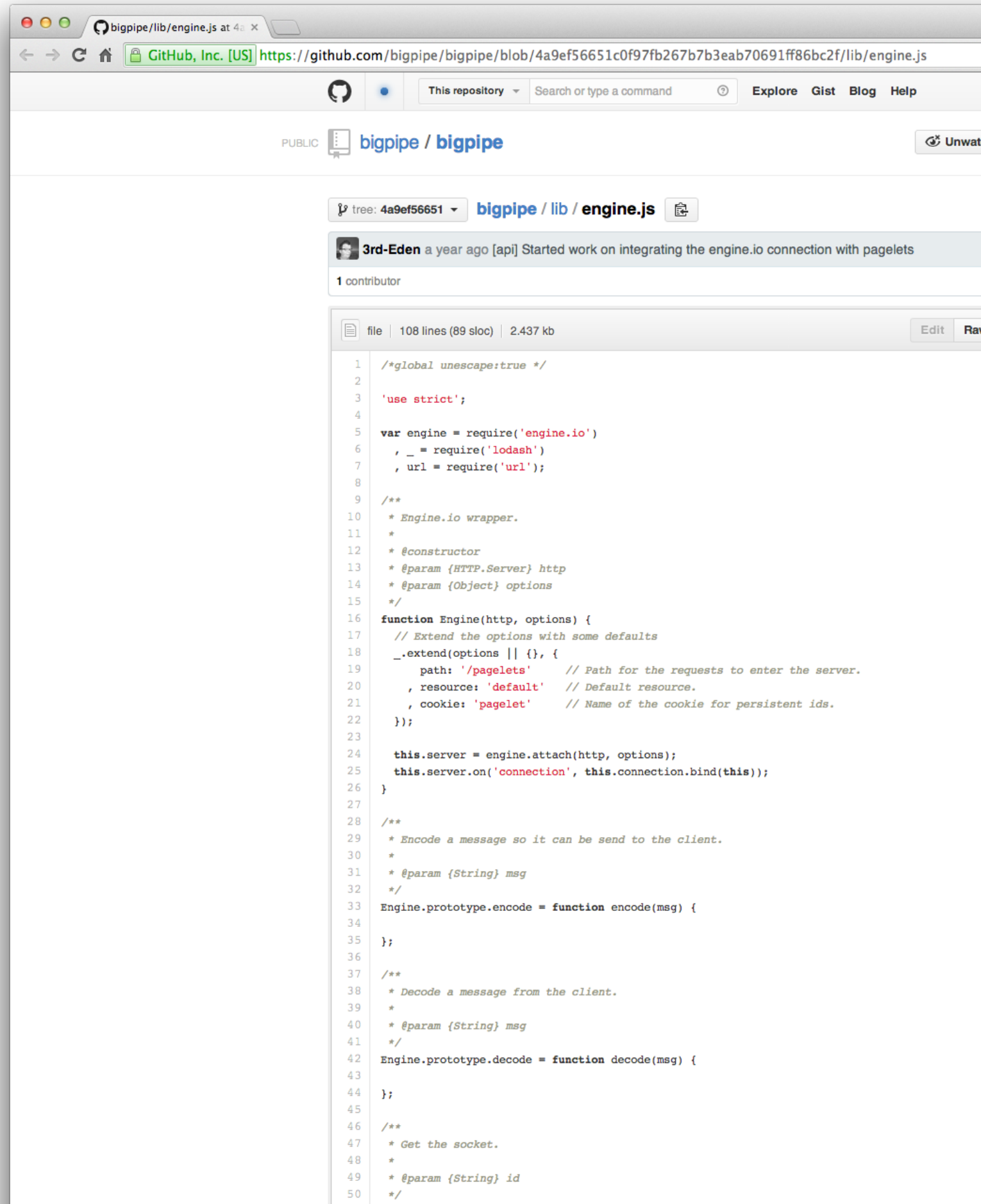
- seamless switching

engine.io, sockjs, socket.io, websockets
and browserchannel

- additional fixes
- plugin support
- custom encoders
- reconnect that works
- offline detection
- build with 

primus

started out as a wrapper
around engine.io

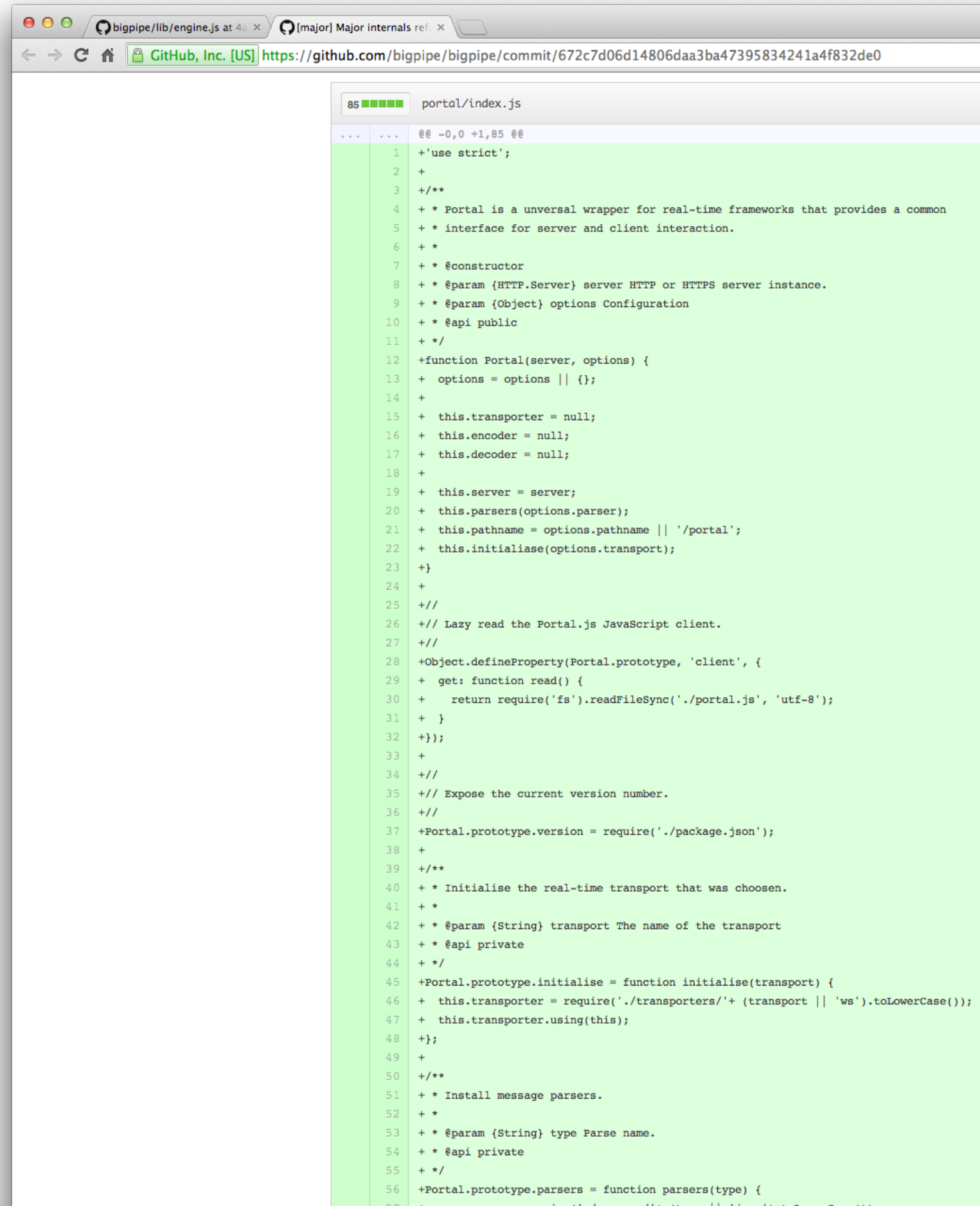


The screenshot shows a web browser displaying the GitHub repository for `bigpipe/lib/engine.js`. The browser's address bar shows the URL `https://github.com/bigpipe/bigpipe/blob/4a9ef56651c0f97fb267b7b3eab70691ff86bc2f/lib/engine.js`. The repository page includes a commit history section with a commit by `3rd-Eden` a year ago, titled `[api] Started work on integrating the engine.io connection with pagelets`. Below this, the file `engine.js` is shown with 108 lines (89 sloc) and a size of 2.437 kb. The code is a JavaScript file that defines the `Engine` constructor and its methods. It includes comments in Chinese and uses `/**` for documentation. The code is as follows:

```
1  /*global unescape:true */
2
3  'use strict';
4
5  var engine = require('engine.io')
6    , _ = require('lodash')
7    , url = require('url');
8
9  /**
10   * Engine.io wrapper.
11   *
12   * @constructor
13   * @param {HTTP.Server} http
14   * @param {Object} options
15   */
16  function Engine(http, options) {
17    // Extend the options with some defaults
18    _.extend(options || {}, {
19      path: '/pagelets' // Path for the requests to enter the server.
20      , resource: 'default' // Default resource.
21      , cookie: 'pagelet' // Name of the cookie for persistent ids.
22    });
23
24    this.server = engine.attach(http, options);
25    this.server.on('connection', this.connection.bind(this));
26  }
27
28  /**
29   * Encode a message so it can be send to the client.
30   *
31   * @param {String} msg
32   */
33  Engine.prototype.encode = function encode(msg) {
34
35  };
36
37  /**
38   * Decode a message from the client.
39   *
40   * @param {String} msg
41   */
42  Engine.prototype.decode = function decode(msg) {
43
44  };
45
46  /**
47   * Get the socket.
48   *
49   * @param {String} id
50   */
```

primus

eventually got refactored to
support more frameworks



```
85 portal/index.js
... @@ -0,0 +1,85 @@
1  + 'use strict';
2  +
3  + /**
4  +  * Portal is a universal wrapper for real-time frameworks that provides a common
5  +  * interface for server and client interaction.
6  +  *
7  +  * @constructor
8  +  * @param {HTTP.Server} server HTTP or HTTPS server instance.
9  +  * @param {Object} options Configuration
10 +  * @api public
11 +  */
12 + function Portal(server, options) {
13 +   options = options || {};
14 +
15 +   this.transporter = null;
16 +   this.encoder = null;
17 +   this.decoder = null;
18 +
19 +   this.server = server;
20 +   this.parsers(options.parser);
21 +   this.pathname = options.pathname || '/portal';
22 +   this.initialise(options.transport);
23 + }
24 +
25 + //
26 + // Lazy read the Portal.js JavaScript client.
27 + //
28 + Object.defineProperty(Portal.prototype, 'client', {
29 +   get: function read() {
30 +     return require('fs').readFileSync('./portal.js', 'utf-8');
31 +   }
32 + });
33 +
34 + //
35 + // Expose the current version number.
36 + //
37 + Portal.prototype.version = require('./package.json');
38 +
39 + /**
40 +  * Initialise the real-time transport that was chosen.
41 +  *
42 +  * @param {String} transport The name of the transport
43 +  * @api private
44 +  */
45 + Portal.prototype.initialise = function initialise(transport) {
46 +   this.transporter = require('./transporters/' + (transport || 'ws').toLowerCase());
47 +   this.transporter.using(this);
48 + };
49 +
50 + //
51 + * Install message parsers.
52 + *
53 + * @param {String} type Parse name.
54 + * @api private
55 + */
56 + Portal.prototype.parsers = function parsers(type) {
57 +   // ...
58 + }
```

primus

and finally it got named as
we know it

The screenshot shows a web browser window displaying a GitHub commit page for the repository 'primus / primus'. The commit message is '[major] Initial brain fart' by user '3rd-Eden' from 7 months ago. The commit hash is 6db821f9da7e5c9ab806f6f6bf469dd5d17c022c. The page shows two files changed: README.md and index.js. The README.md diff shows changes to the title, description, installation instructions, and license. The index.js diff shows the addition of a new file with a strict mode declaration, a require statement for 'spark', and a function definition for 'Primus'.

Browser tabs: [major] Initial brain fart · 6 x

GitHub, Inc. [US] <https://github.com/primus/primus/commit/6db821f9da7e5c9ab806f6f6bf469dd5d17c022c>

Public primus / primus Unwatch

[major] Initial brain fart
master 1.5.2 ... 0.1.0
3rd-Eden authored 7 months ago 0 parents commit 6db821f9da7e5c9ab806f6f6bf469dd5d17c022c

Showing 9 changed files with 779 additions and 0 deletions.

19 README.md


```
@@ -0,0 +1,19 @@
1  + # Primus
2  +
3  + Primus, the creator god of transformers but now also known as universal wrapper
4  + for real-time frameworks. There are a lot of real-time frameworks available for
5  + Node.js and they all have different opinions on how real-time should be done.
6  + Primus provides a common low level interface to communicate in real-time using
7  + various of real-time frameworks.
8  +
9  + ### Installation
10 +
11 + Primus is released in `npm` and can be installed using:
12 +
13 + ```
14 + npm install primus --save
15 + ```
16 +
17 + ### License
18 +
19 + MIT
```

123 index.js

```
@@ -0,0 +1,123 @@
1  + 'use strict';
2  +
3  + var Spark = require('./spark');
4  +
5  + /**
6  + * Primus is a universal wrapper for real-time frameworks that provides a common
7  + * interface for server and client interaction.
8  + *
9  + * @constructor
10 + * @param {HTTP.Server} server HTTP or HTTPS server instance.
11 + * @param {Object} options Configuration
12 + * @api public
13 + */
14 + function Primus(server, options) {
15 +   options = options || {};
16 +
17 +   this.transporter = null;
18 +   this.encoder = null;
19 +   this.decoder = null;
20 + }
```

compiler

optimising dependencies



```
'use strict';

var Page = require('bigpipe').Page;

Page.extend({
  path: '/404',
  statusCode: 404,
  view: '404.ejs'
}).on(module);
```

```
'use strict';

var Page = require('bigpipe').Page;

Page.extend({
  path: '/500',
  statusCode: 500,
  view: '500.ejs'
}).on(module);
```

status page

custom and default error
pages




```
'use strict';
```

```
bigpipe.use('ack', {  
  // Only ran on the server.  
  server: function (bigpipe, options) {},  
  
  // Runs on the client, it's automatically  
  // bundled.  
  client: function (bigpipe, options) {},  
  
  // Optional library that needs to be  
  // bundled on the client  
  library: '',  
  
  // Optional plugin specific options, will  
  // be merged with Bigpipe.options  
  options: {}  
});
```

plugins

a great way to extend BigPipe




```
var BigPipe = require("bigpipe")
  , serve = require("static-serve");

var bigpipe = BigPipe.createServer(8080, {
  pages: __dirname + '/pages',
  dist:  __dirname + '/dist'
});

bigpipe.before('static', serve());
```

middleware

for the sake of compatibility




```
var BigPipe = require("bigpipe")
  , serve = require("static-serve");

var bigpipe = BigPipe.createServer(8080, {
  pages: __dirname + '/pages',
  dist: __dirname + '/dist'
});

bigpipe.before('static', serve(), {
  primus: true
});
```

middleware

can be proxied to Primus



```
var BigPipe = require("bigpipe")
  , serve = require("static-serve");

var bigpipe = BigPipe.createServer(8080, {
  pages: __dirname + '/pages',
  dist:  __dirname + '/dist'
});

bigpipe.use('static', serve());
bigpipe.disable('static');
bigpipe.enable('static');
bigpipe.remove('static');
bigpipe.before('static');
```

middleware

can be enabled/disabled at will

live demo

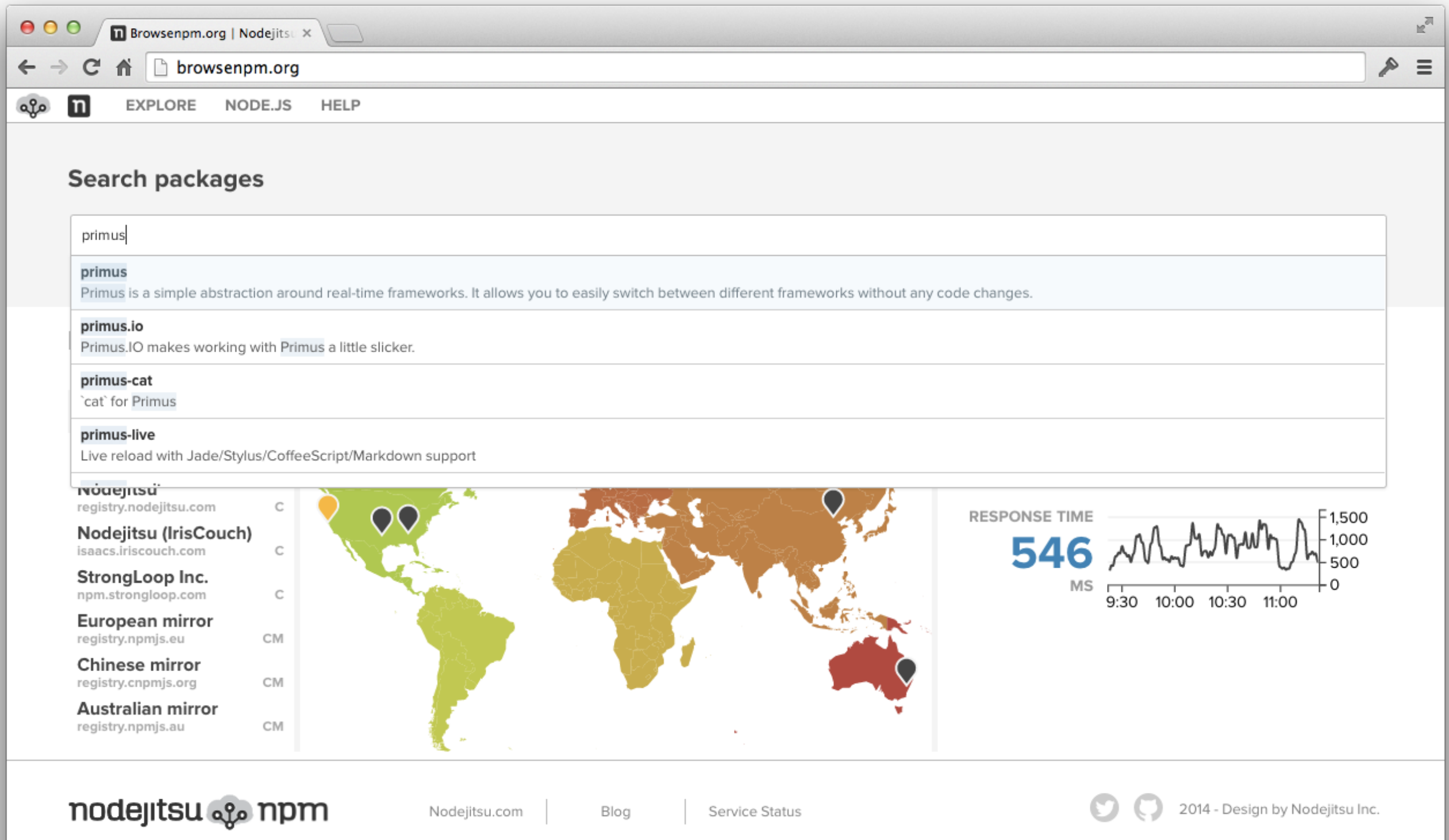


crosses fingers

browsenpm.org

Open Source example of a BigPipe powered site

public package page



The screenshot shows the browsenpm.org website. The browser address bar displays "browsenpm.org". The navigation bar includes "EXPLORE", "NODE.JS", and "HELP". The main content area features a "Search packages" section with a search bar containing "primus". Below the search bar, a list of search results is shown:

- primus**
Primus is a simple abstraction around real-time frameworks. It allows you to easily switch between different frameworks without any code changes.
- primus.io**
Primus.IO makes working with Primus a little slicker.
- primus-cat**
`cat` for Primus
- primus-live**
Live reload with Jade/Stylus/CoffeeScript/Markdown support

Below the search results, a world map is displayed with several location pins. To the right of the map, a "RESPONSE TIME" section shows a large blue number "546" followed by "MS". Below this, a line graph shows response time fluctuations over time, with the x-axis labeled from 9:30 to 11:00 and the y-axis labeled from 0 to 1,500.

At the bottom of the page, the footer includes the "nodejitsu npm" logo, the text "Nodejitsu.com | Blog | Service Status", and social media icons for Twitter and GitHub, followed by the text "2014 - Design by Nodejitsu Inc."

browsenpm.org

github.com/nodejitsu/browsenpm.org

github.com/nodejitsu/packages-pagelet

github.com/nodejitsu/npm-dependencies-pagelet

github.com/nodejitsu/npm-documentation-pagelet

github.com/nodejitsu/registry-status-pagelet

github.com/nodejitsu/npm-search-pagelet

future goals

Quicklining or “turbo links”, pagelet & page cache, recursive/nested pagelets, streaming/diff-able assets, more tests and benchmarking.

That's all I got!

Q/A: Tweet me @3rdEden or create github issues ;-)



3rdEden



3rd-Eden



arnout.kazemier

bigpipe.io

pagelet:	github.com/bigpipe/paglet
bigpipe:	github.com/bigpipe/bigpipe
pipe.js:	github.com/bigpipe/pipe.js
primus:	github.com/primus/primus
browsenpm:	browsenpm.org



3rdEden



3rd-Eden



arnout.kazemier