



I Am Devloper
@iamdevloper

 Follow

I think I've had milk last longer than some JavaScript frameworks.

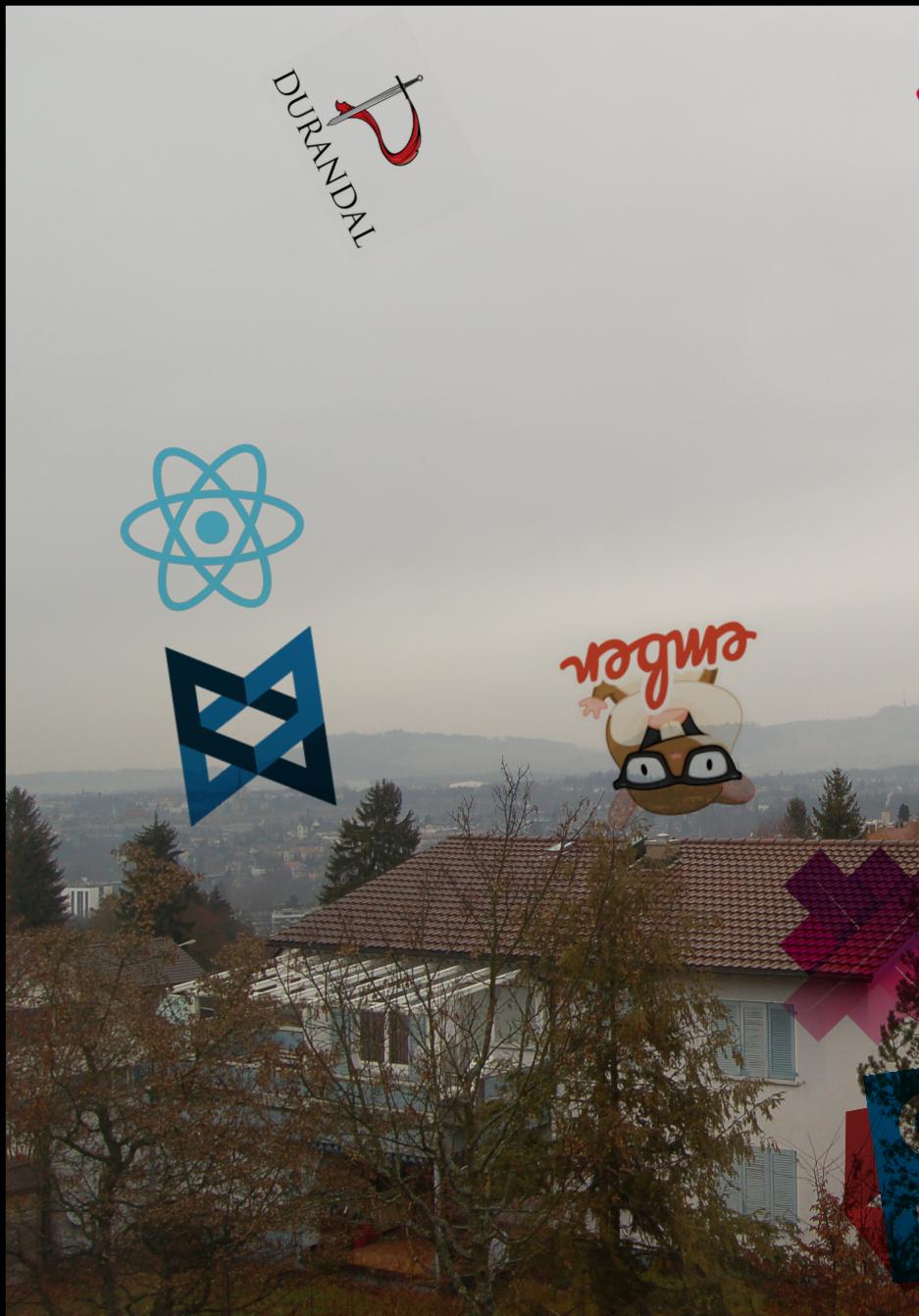
1:22 PM - 4 Dec 2014



1,760



1,122





JavaScript knowledge that lasts longer
than milk

Pure JavaScript

ES6, Web Components and other APIs

Stefan Jäger

Berne /  steivan /  stefanjaeger



Christian Kohler

Zurich /  KohlerChristian /  christiankohler



zühlke
empowering ideas



We came here by train watching a movie



http://movie-finder/final

MOVIE on the TRAIN

Enter your train route and we tell you which movie to watch

from

to

MOVETIME!

Let's start with a component



The Matrix

8.7 Rating

136 min

```
<movie-tile>
```

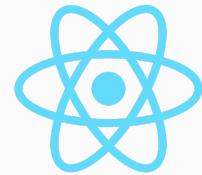
```
  title="The Matrix"  
  image="matrix.png"  
  rating="8.7"  
  runtime="136 min">
```

```
</movie-tile>
```

We don't want to use



Angular Directives



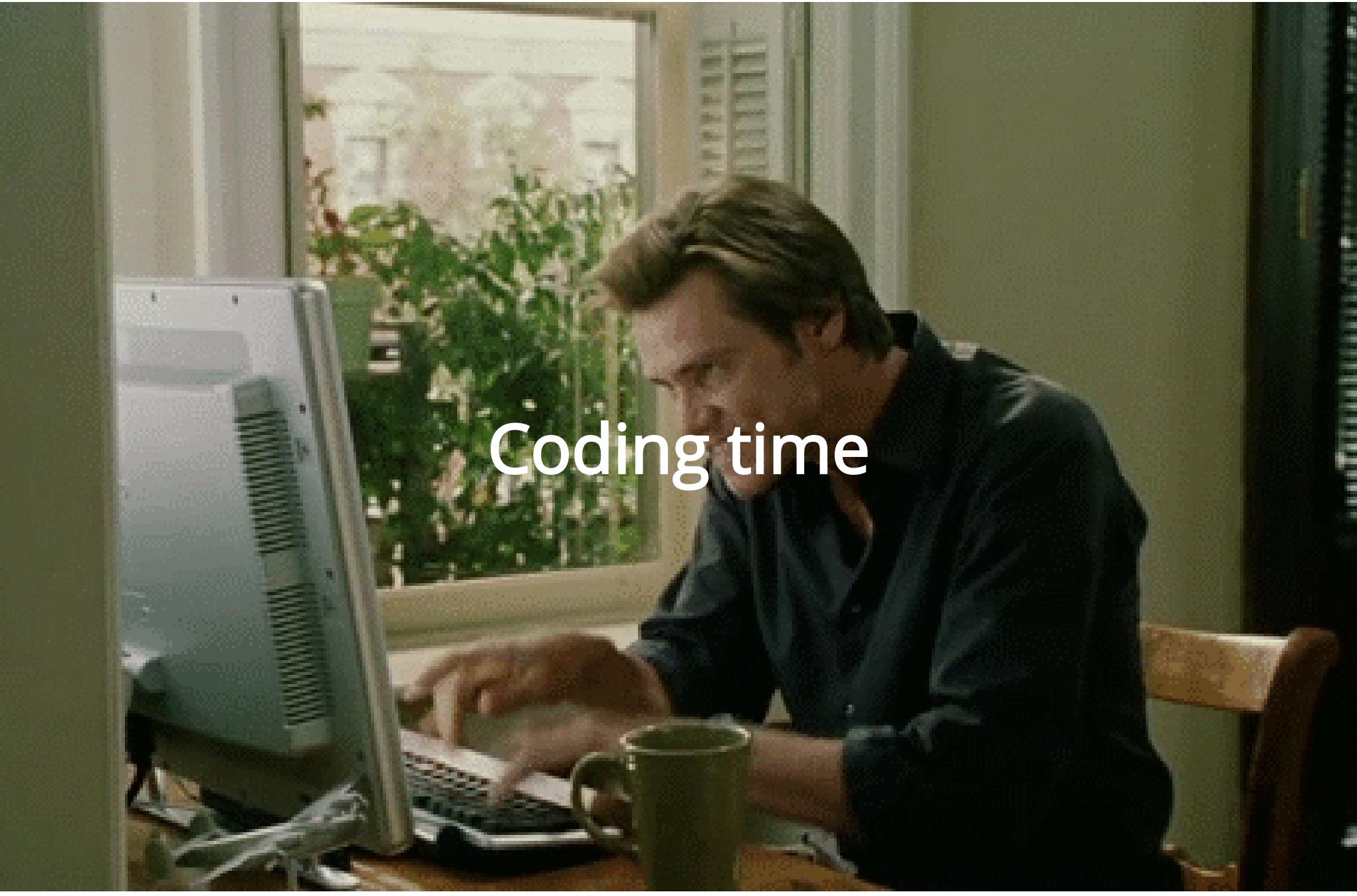
React Components

Custom Elements

<road-bike></road-bike>

```
class RoadBike extends HTMLElement {
  createdCallback() {
    this.innerHTML = `
      <div>
        <div class="frame">
          ...
        </div>
        <div class="chain">
          ...
        </div>
        <div class="saddle">
          ...
        </div>
      </div>`;
  }
}

document.registerElement('road-bike', {
  prototype: RoadBike.prototype
});
```



Coding time



app.js

1



index.html app.js

1



index.html app.js

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Train Movie</title>
6   <meta name=viewport content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="app/styles.css" type="text/css">
8   <script src="system.js"></script>
9 </head>
10 <body>
11
12   <movie-tile></movie-tile>
13
14   <script>
15     System.import('app');
16   </script>
17 </body>
18 </html>
```

```
1 class MovieTile extends HTMLElement {
2     createdCallback() {
3         render();
4     }
5
6     render() {
7         this.innerHTML =
8             
9             <p class="title">Batman Begins</p>
10            <p>9.9 Rating</p>
11            <p>49 min</p>
12        ;
13    }
14}
15
16 document.registerElement('movie-tile', {
17     prototype: MovieTile.prototype
18});
```

Demo

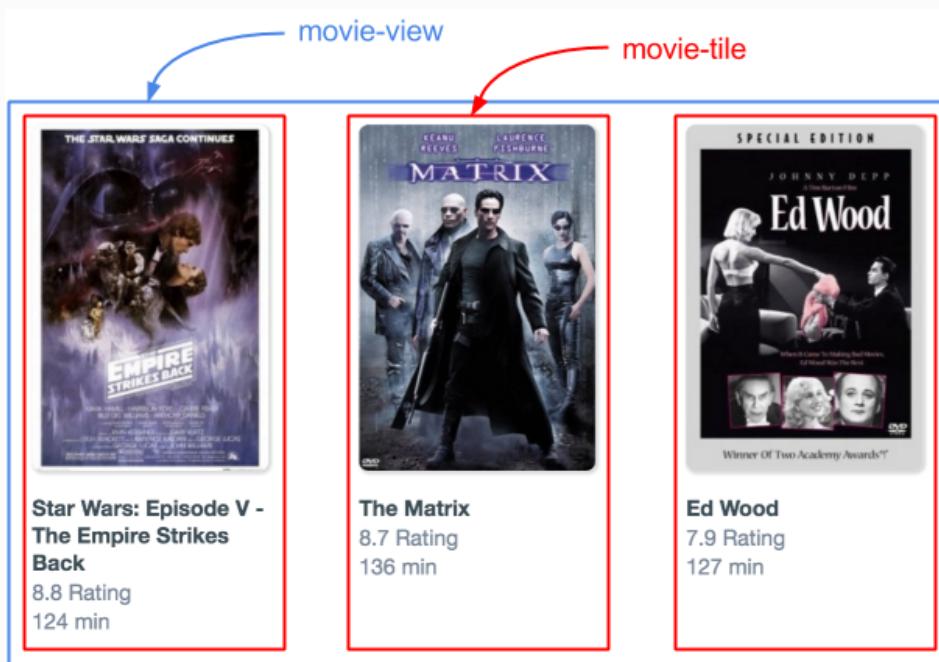
http://movie-finder/v1



The Matrix

The Matrix
8.7 Rating
136 min

Display multiple movies in a view

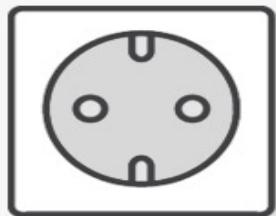


```
<movie-view>
  <movie-tile></movie-tile>
  <movie-tile></movie-tile>
  <movie-tile></movie-tile>
</movie-view>
```

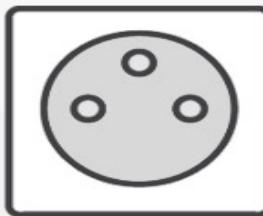
```
1 class MovieTile extends HTMLElement {
2     createdCallback() {
3         render(this.attributes);
4     }
5
6     render(attrs) {
7         this.innerHTML =
8             `
```

How can we organize our application?

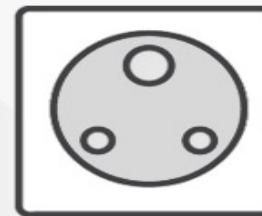
ECMAScript 6 Modules



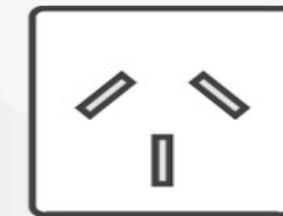
TYPE F
EUROPE, RUSSIA



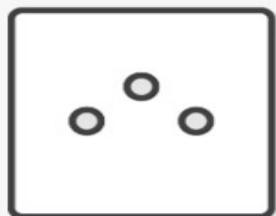
TYPE E
FRA, BEL, POL,
SVK & CZE



TYPE D
INDIA



TYPE I
AUS, ARG, CHN
& NZL



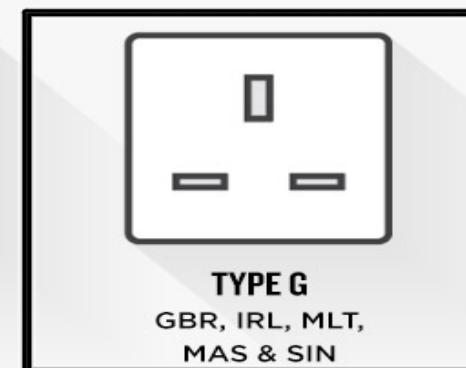
TYPE J
SUI, LIE & RWA



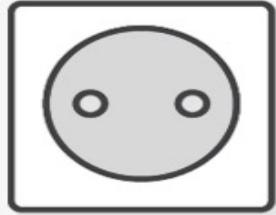
TYPE A
USA, MEX, CAN
& JPN



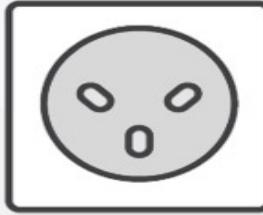
TYPE B
USA, MEX, CAN
& JPN



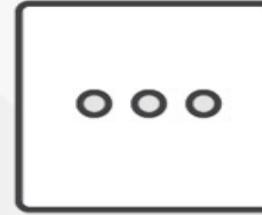
TYPE G
GBR, IRL, MLT,
MAS & SIN



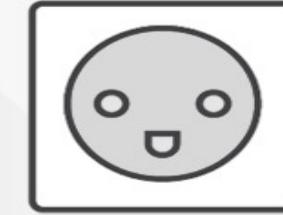
TYPE C
EUROPE, SOUTH AMERICA
& ASIA



TYPE H
ISRAEL



TYPE L
ITALY & CHILE



TYPE K
DENMARK
& GREENLAND

Modules

```
// calc.js
export function multiplyBySeven(a) {
  return a * 7;
}
```

```
// app.js
import { multiplyBySeven } from 'calc';
console.log(multiplyBySeven(3));
```

```
1 class MovieTile extends HTMLElement {
2     createdCallback() {
3         render(this.attributes);
4     }
5
6     render(attrs) {
7         this.innerHTML = `
8             
9             <p class="title">${attrs.title.value}</p>
10            <p>${attrs.rating.value} Rating</p>
11            <p>${attrs.runtime.value}</p>
12        `;
13    }
14 }
15
16 class MoviesView extends HTMLElement {
17     createdCallback() {
18         const movies = [ {}, {}, {} ];
19         this.render(movies);
20     }
21
22     render(movies) {
23         this.innerHTML = `
24             <div class="movies-container">
25                 ${movies.map(this.renderMovie)}
26             </div>
27         `;
28     }
29
30     renderMovie(movie) {
```



index.html

app.js

movies.js

1



index.html app.js

movie-tile.js

movies.js

1



index.html

app.js

movie-tile.js

movies.js

1



index.html app.js movie-tile.js movies.js

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Train Movie</title>
6   <meta name=viewport content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="app/styles.css" type="text/css">
8   <script src="system.js"></script>
9 </head>
10 <body>
11
12   <movie-tile image="matrix.jpg"
13     title="The Matrix"
14     rating="9.9"
15     runtime="49 min">
16   </movie-tile>
17
18   <script>
19     System.import('app');
20   </script>
21 </body>
22 </html>
```

Demo

http://movie-finder/v2

The image shows the front cover of the Star Wars: Episode V - The Empire Strikes Back DVD. It features the main characters Luke Skywalker, Han Solo, and Leia Organa in a dark, atmospheric setting. The title "THE STAR WARS SAGA CONTINUES" is at the top, and "THE EMPIRE STRIKES BACK" is prominently displayed in the center.

**Star Wars: Episode V -
The Empire Strikes
Back**

8.8 Rating
124 min

The image shows the front cover of the The Matrix DVD. It features the main cast members Neo, Trinity, and Agent Smith standing in a dark, urban environment. The title "THE MATRIX" is at the top, with the names "KEANU REEVES" and "LAURENCE FISHBURNE" above it.

The Matrix

8.7 Rating
136 min

The image shows the front cover of the Ed Wood Special Edition DVD. It features Johnny Depp as Ed Wood in a black suit, with a woman in a white dress standing behind him. The title "Ed Wood" is at the top, with "JOHNNY DEPP" and "A Tim Burton Film" above it. The cover also includes a quote: "When it Come To Making Bad Movies, Ed Wood Has The Best." and a small image of the three lead actors.

Ed Wood

7.9 Rating
127 min

Let's use real data

Can we use \$http?

Fetch API

It is a living standard

<http://fetch.spec.whatwg.org>

There is a polyfill

<https://github.com/github/fetch>

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari (WebKit)
Basic support	42.0	39 (39) 34 behind pref	Not supported	29 28 behind pref	Not supported
Streaming response body	43.0	?	?	?	?



```
function reqListener() {  
  var data = JSON.parse(this.responseText);  
  console.log(data);  
}  
  
function reqError(err) {  
  console.log('Fetch Error :-S', err);  
}  
  
var oReq = new XMLHttpRequest();  
oReq.onload = reqListener;  
oReq.onerror = reqError;  
oReq.open('get', './api/some.json', true);  
oReq.send();
```



```
fetch('./api/some.json')
  .then(result => result.json())
  .then(data => console.log(data))
  .catch(err => console.log('Fetch Error :-S', err));
```



index.html app.js movie-tile.js movies.js

movies-api.js

1

```
1 import MovieTile from './movie-tile';
2
3 class MoviesView extends HTMLElement {
4     createdCallback() {
5         const movies = [ {}, {}, {} ];
6         this.render(movies);
7     }
8
9     render(movies) {
10         this.innerHTML =
11             <div class="movies-container">
12                 ${movies.map(this.renderMovie)}
13             </div>
14         ;
15     }
16
17     renderMovie(movie) {
18         return
19             <movie-tile image="${movie.image}"
20                         title="${movie.title}"
21                         runtime="${movie.runtime}"
22                         rating="${movie.rating}">
23             </movie-tile>
24         ;
25     }
26 }
27
28 document.registerElement('movies-view', {
29     prototype: MoviesView.prototype
30 }).;
```

Demo

http://movie-finder/v3

The interface displays movie cards with the following details:

Movie Title	Rating	Length
Blue Suede Shoes	8.9 Rating	50 min
The Legend of Robin Hood	7.7 Rating	50 min
The Wonderful World of Dogs	7.6 Rating	52 min
The Cola Conquest	7.5 Rating	50 min
MICHAEL JORDAN in MAX		
A Tale of Two Critters		
ALAN DAVIES		
RONNIE BARKER		

Blue Suede Shoes
8.9 Rating
50 min

The Legend of Robin Hood
7.7 Rating
50 min

The Wonderful World of Dogs
7.6 Rating
52 min

The Cola Conquest
7.5 Rating
50 min

MICHAEL JORDAN in MAX

A Tale of Two Critters

ALAN DAVIES

RONNIE BARKER

Introducing search view

Use Google Maps API to get the duration



index.html app.js movie-tile.js movies.js movies-api.js google-api.js

1



index.html app.js movie-tile.js movies.js movies-api.js google-api.js search.js

1



index.html

app.js

movie-tile.js

movies.js

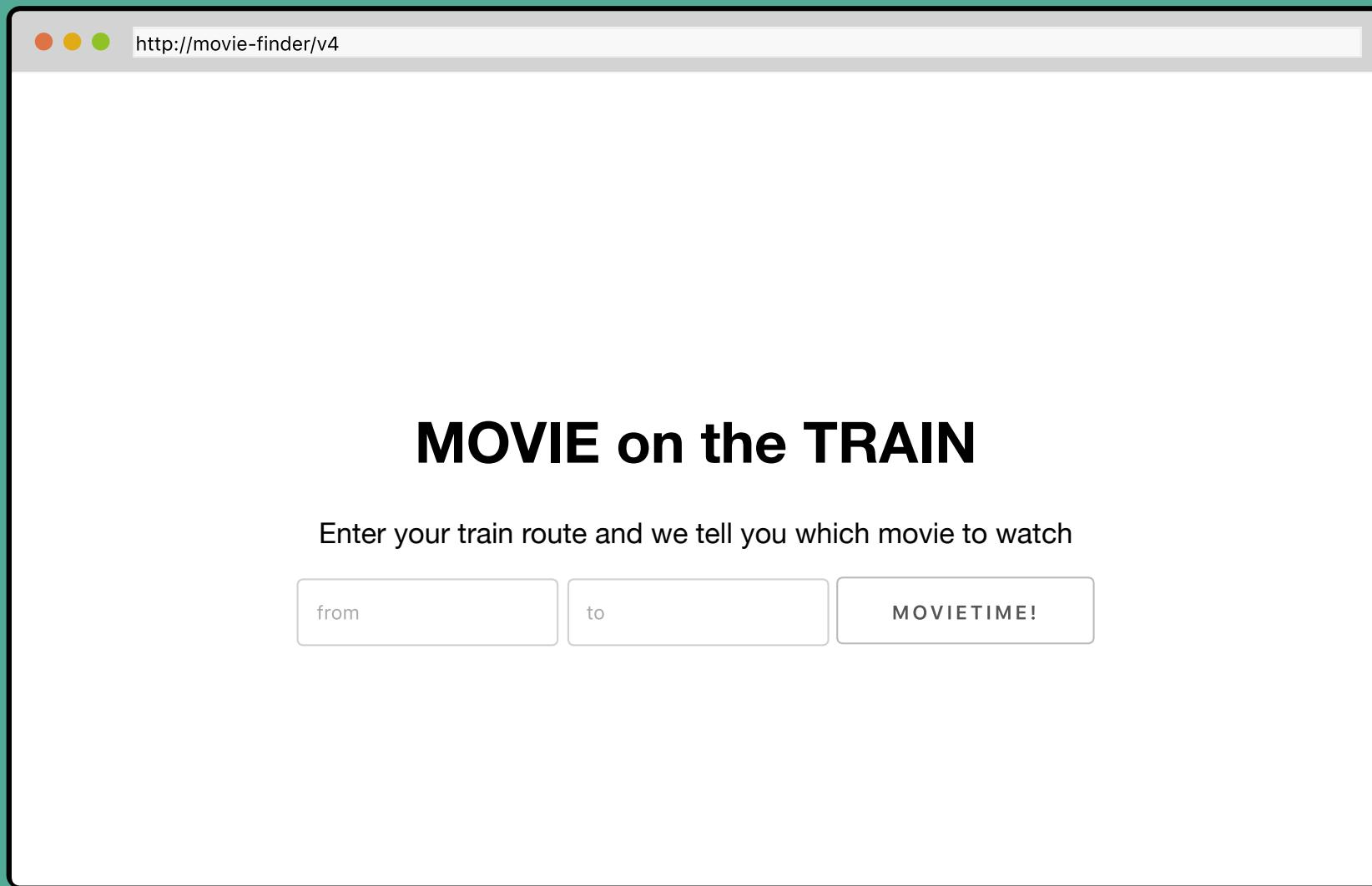
movies-api.js

google-api.js

search.js

```
1 import MoviesView from './movies';
2
3 class AppRoot extends HTMLElement {
4     createdCallback() {
5         this.render();
6     }
7
8     render() {
9         this.innerHTML =
10            `<movies-view></movies-view>
11            `;
12     }
13 }
14
15 document.registerElement('app-root', {
16     prototype: AppRoot.prototype
17});
```

Demo



Router

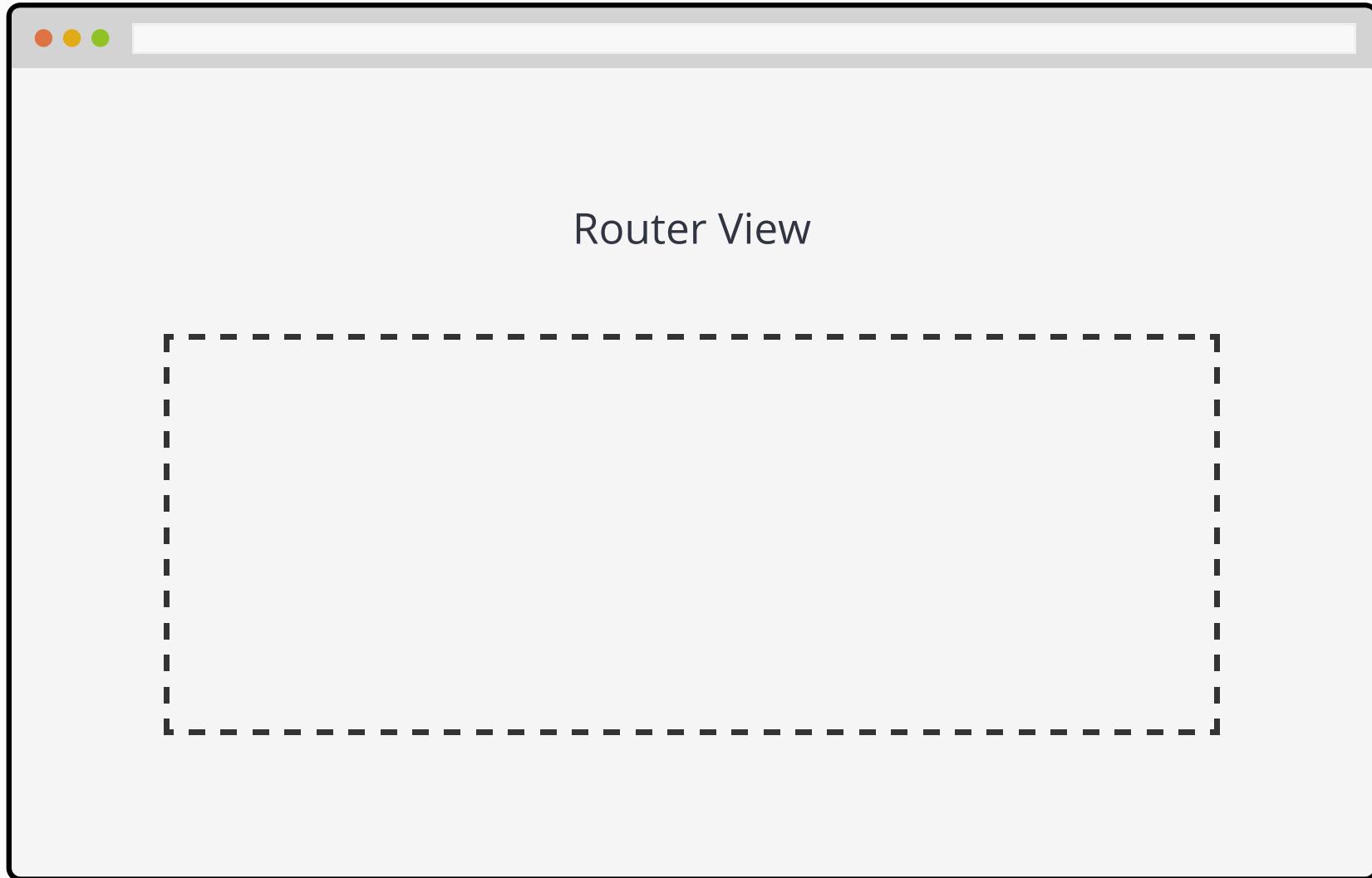
Can we use ng-route?

The routing standard

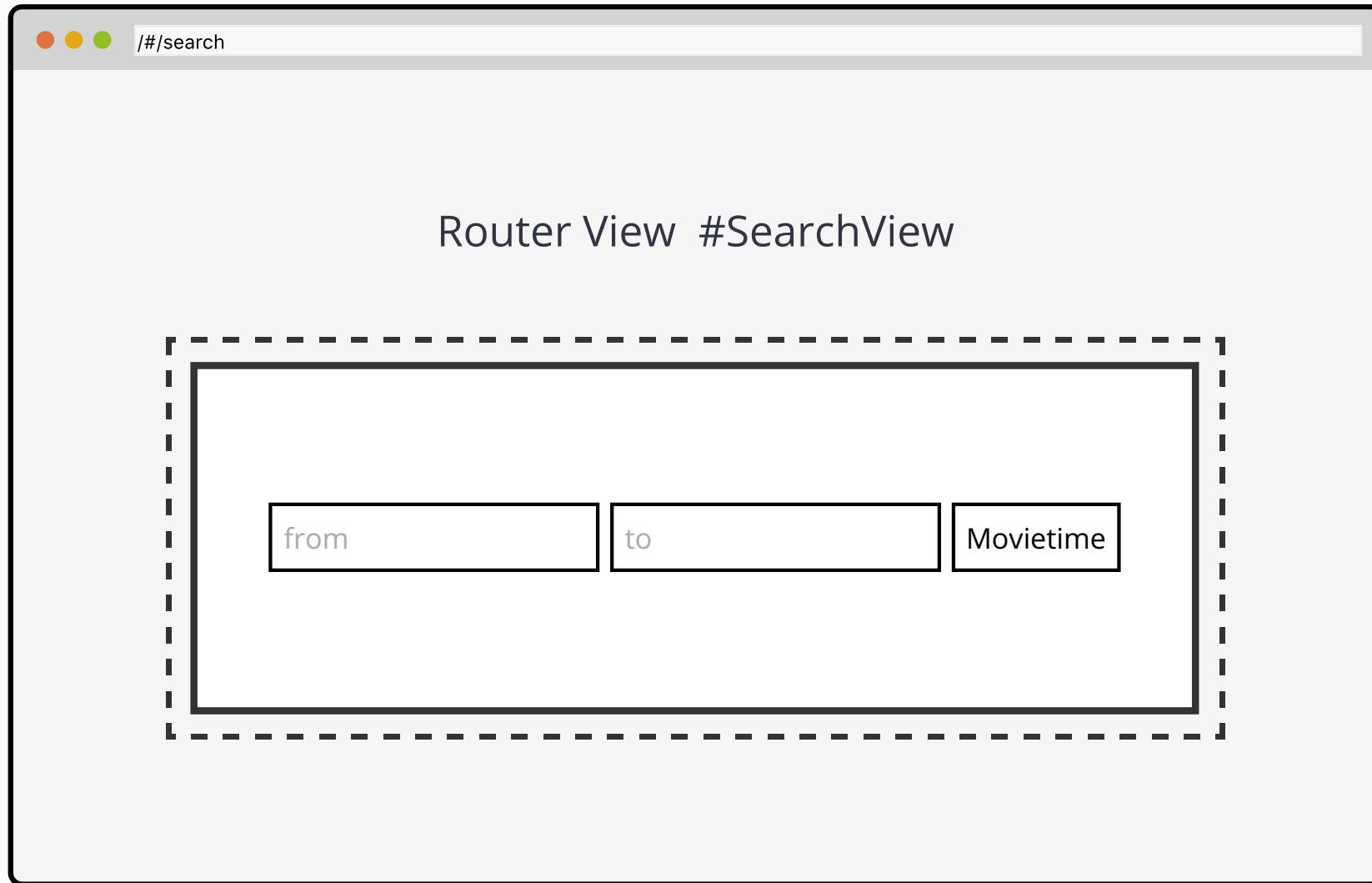


There is no standard yet

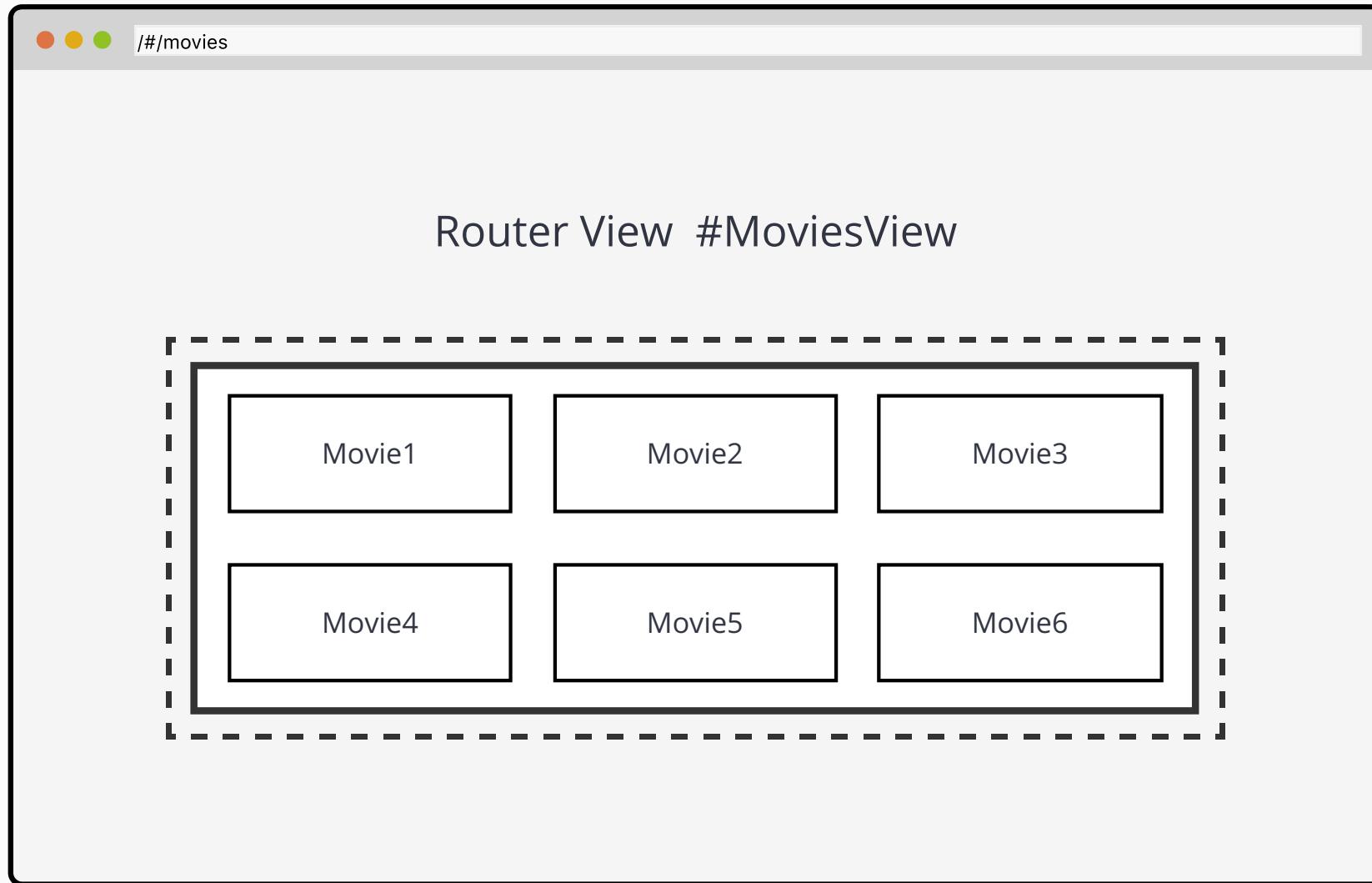
How does a router work?



How does a router work?



How does a router work?





index.html

app.js

movie-tile.js

movies.js

movies-api.js

google-api.js

search.js

```
1 import MoviesView from './movies';
2
3 class AppRoot extends HTMLElement {
4     createdCallback() {
5         this.render();
6     }
7
8     render() {
9         this.innerHTML =
10            `<movies-view></movies-view>
11            `;
12     }
13 }
14
15 document.registerElement('app-root', {
16     prototype: AppRoot.prototype
17});
```



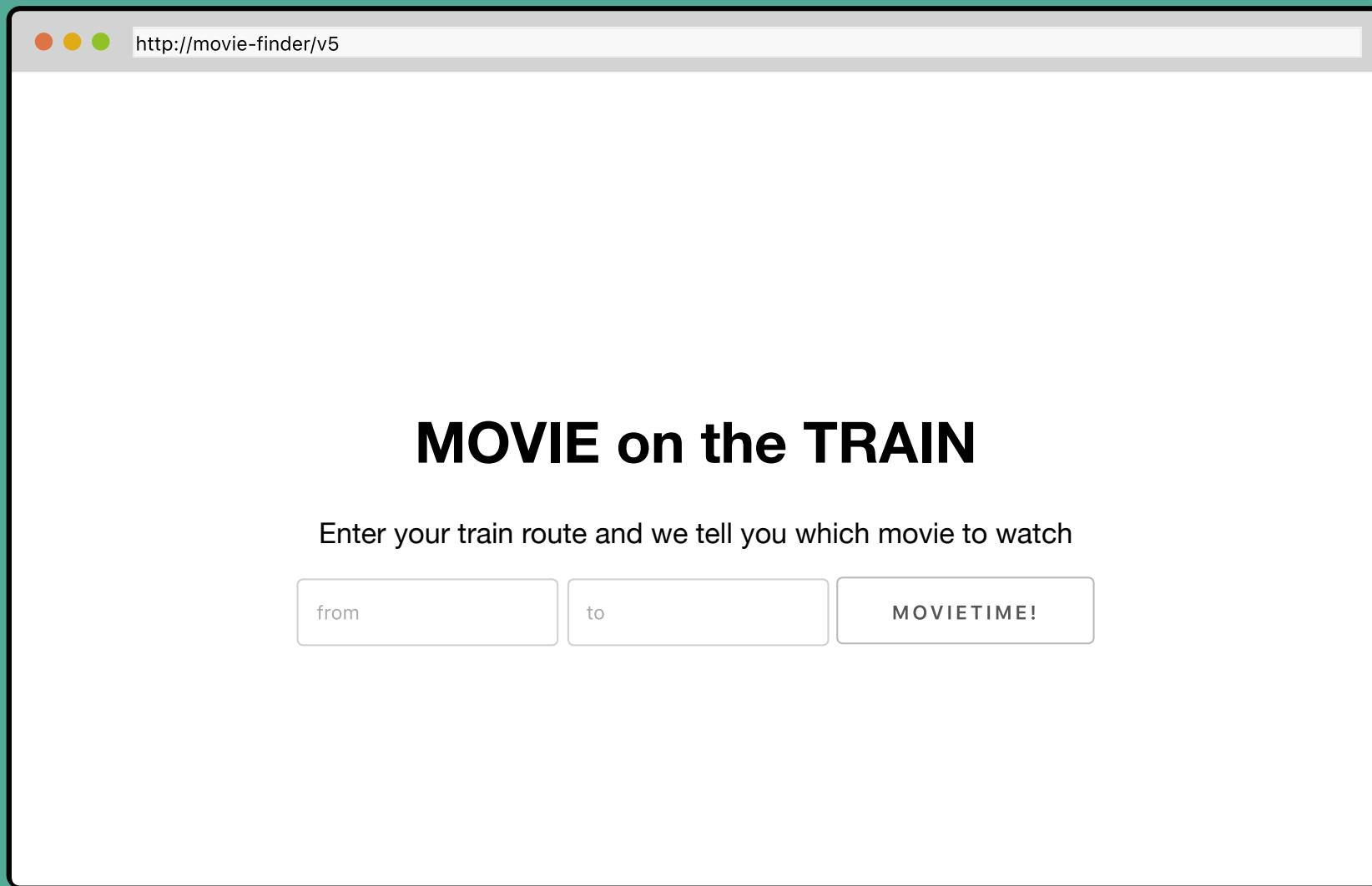
index.html app.js movie-tile.js movies.js movies-api.js google-api.js search.js router-view.js

```
1 class RouterView extends HTMLElement {
2     createdCallback() {
3     }
4
5     render() {
6     }
7 }
8
9 export default document.registerElement('router-view', {
10     prototype: RouterView.prototype
11});
```

```
1 class RouterConfig {
2     constructor() {
3         this.config = [ ];
4     }
5
6     map(routes) {
7         this.config = [this.config, ...routes];
8     }
9 }
10
11 class Router {
12
13     constructor() {
14         this.routerConfig = new RouterConfig();
15     }
16
17     configure(configCallback) {}
18
19     navigate(route) {}
20
21     registerRouterView(routerView) {}
22 }
23
24 export default new Router();
```

```
1 import GoogleApi from './google-api';
2
3 class SearchView extends HTMLElement {
4     createdCallback() {
5         this.render();
6         this.addEventListener('submit', this.search);
7     }
8
9     search(e) {
10        e.preventDefault();
11
12        const from = this.querySelector('input[name="from"]').value;
13        const to = this.querySelector('input[name="to"]').value;
14
15        GoogleApi.search(from, to).then(duration => {
16            // go to movies result page with this duration
17        });
18    }
19
20    render() {
21        this.innerHTML =
22            <div class="search">
23                <h1>MOVIE on the TRAIN</h1>
24                <p>Enter your train route and we tell you which movie to watch</p>
25                <form>
26                    <input name="from" type="text" placeholder="from" required>
27                    <input name="to" type="text" placeholder="to" required>
28                    <button type="submit">Movietime!</button>
29                </form>
30            </div>
```

Final App Demo



A close-up, black and white portrait of Clint Eastwood's face. He has a serious, intense expression, looking directly at the viewer. His skin is weathered, and he has a prominent mustache. The lighting is dramatic, casting deep shadows on one side of his face while highlighting the other. In the bottom left corner, a portion of a rifle's barrel and stock is visible, suggesting a Western setting.

The good, the bad and the ugly

The good - No dependencies

No breaking changes

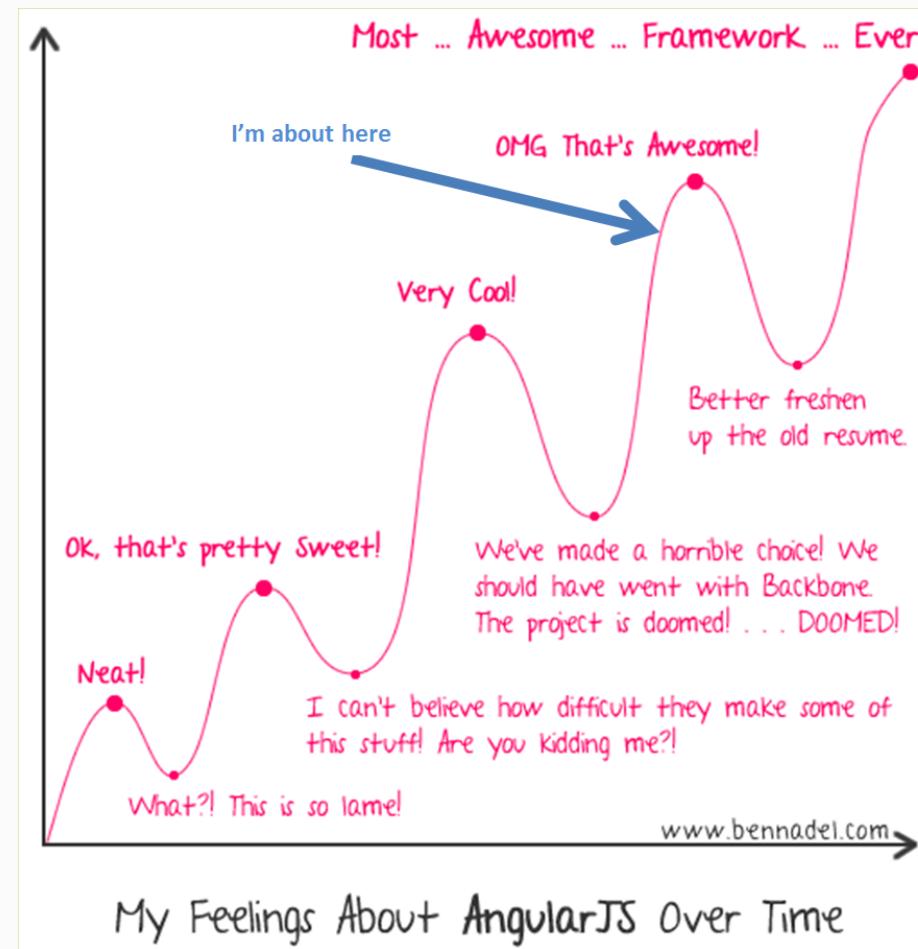
No waiting time for bug fixes

No aging libraries



remember left-pad?

The good - No need to learn a framework



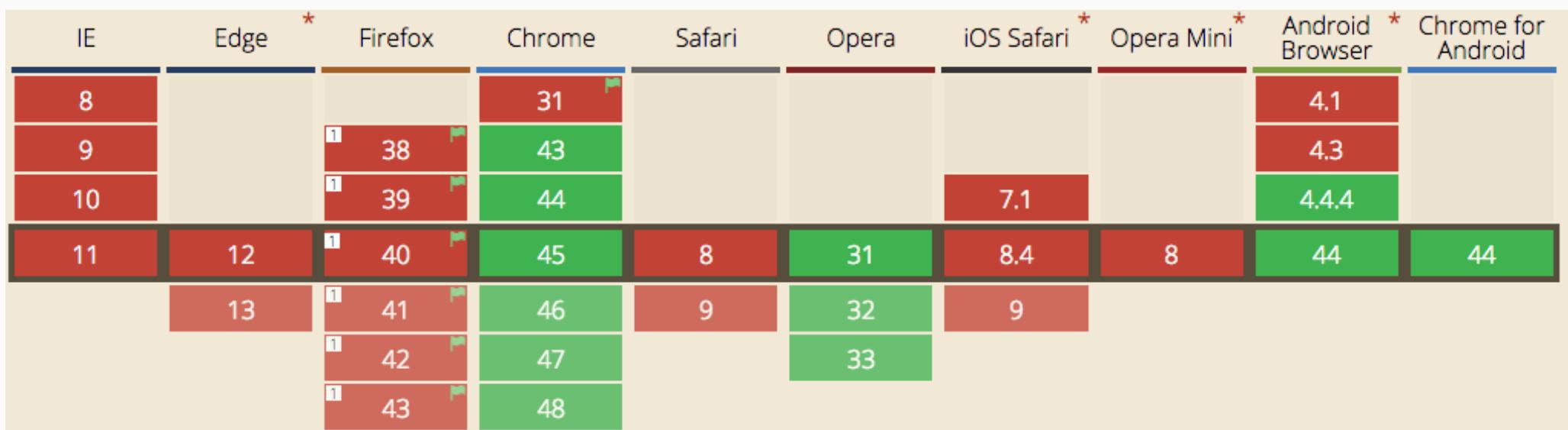
The bad - Difficult to test

No dependency injector, just ES6 Modules

```
import myservice from './services/myservice'
```

The bad - Browser support

Custom Elements (48%)



And the ugly
You end up writing your own framework

The router for example



Lessons learned

#1

learn concepts - not frameworks

Router

Movie App

```
config.map([
  { route: "search", component: SearchView },
  { route: "movies", component: MoviesView }
]);
```

Aurelia

```
config.map([
  { route: 'search', name: 'search', moduleId: './search' },
  { route: 'movies', name: 'movies', moduleId: './movies' }
]);
```

Angular 2

```
@RouteConfig([
  { path: '/search', name: 'Search', component: SearchViewComponent},
  { path: '/movies', name: 'Movies', component: MoviesViewComponent}
])
```

React

```
<Route path="/search" component={Search}>
<Route path="/movies" component={Movies}>
```

#2

use standards as much as possible

Reusable services

React

```
import MovieApi from './movie-api';

React.createClass({
  componentDidMount: () => {
    MovieApi.searchByDuration(props.duration).then(results => {
      this.setState({ results })
    });
  },
  render: () => {
    return (<div>{this.state.results.length} Results</div>);
  }
});
```

#3

learn to think in components

Components

Movie App

```
class HelloComponent extends HTMLElement {}
```

Aurelia

```
@customElement('hello-component')
export class HelloComponent {}
```

Angular 2

```
@Component({
  selector: 'hello-component',
  template: 'Hello {{name}}'
})
export class HelloComponent {}
```

React

```
class HelloComponent extends React.Component {
  render() {
    return Hello {this.props.name};
  }
}
```

#4

learn a standard once,
use it everywhere

ES6 Modules

Movie App

```
import MoviesView from './movies/movies';
```

Aurelia

```
import {inject} from 'aurelia-framework';
```

Angular 2

```
import {Component} from 'angular2/core';
```

React

```
import React from 'react';
```

Start learning standards today!



Stefan Jäger

 steivan



Christian Kohler

 KohlerChristian

