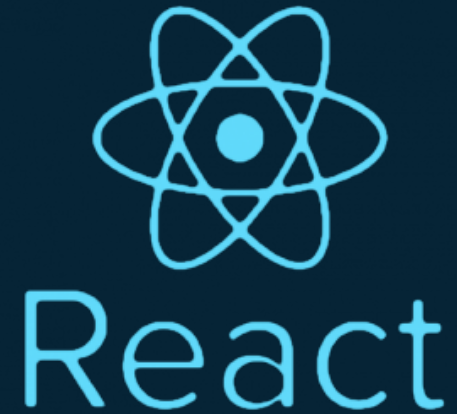


# Building SPA's using

ASP. NET Core +



by Jose Barbosa | @kidchenko

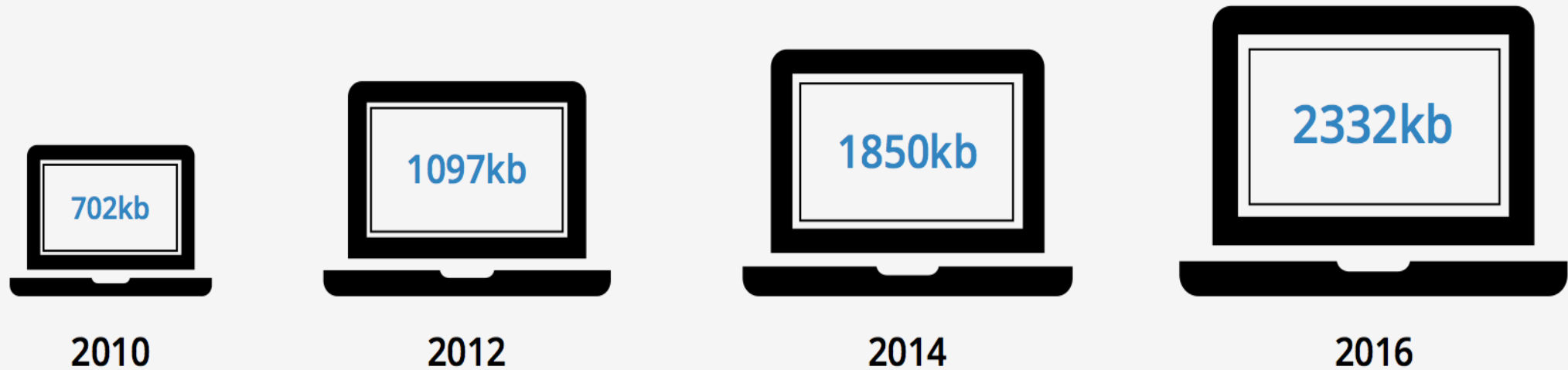
# Single Page Application

“...In an SPA, either all necessary code – HTML, JavaScript, and CSS – is retrieved with a single page load, or the appropriate resources are dynamically loaded and added to the page as necessary, usually in response to user actions. The page does not reload at any point in the process...”

# Benefits!?

## Contents [\[hide\]](#)

- 1 History
- 2 Technical approaches
  - 2.1 JavaScript frameworks
  - 2.2 Ajax
  - 2.3 Websockets
  - 2.4 Server-sent events
  - 2.5 Browser plugins
  - 2.6 Data transport (XML, JSON and Ajax)
  - 2.7 Server architecture
    - 2.7.1 Thin server architecture
    - 2.7.2 Thick stateful server architecture
    - 2.7.3 Thick stateless server architecture
- 3 Running locally
- 4 Challenges with the SPA model
  - 4.1 Search engine optimization
  - 4.2 Client/Server code partitioning
  - 4.3 Browser history
  - 4.4 Analytics
    - 4.4.1 Adding page loads to an SPA
  - 4.5 Speed of initial load
    - 4.5.1 Speeding up the page load
- 5 Page lifecycle
- 6 References
- 7 External links



## The Growth of Web Page Size



**James Ward**

@\_JamesWard

 Follow

Discovered this sign at the W3C headquarters...



RETWEETS  
**2,035**

FAVORITES  
**1,000**



8:17 AM - 11 Feb 2015



**updated daily!**  
Next update in 10 hours

0

if you spot an unlikely mistake on this website, get in touch: [javascriptisa@veryfast.biz](mailto:javascriptisa@veryfast.biz)

<https://dayssincelastjavascriptframework.com/>



**DHH**

@dhh

Creator of Ruby on Rails, Founder & CTO at Basecamp (formerly 37signals), NYT Best-selling author of REWORK and REMOTE, and Le Mans class-winning racing driver.

Chicago, USA

[david.heinemeierhansson.com](http://david.heinemeierhansson.com)

Participa desde abril de 2008

**DHH**

@dhh

Seguindo



Stimulus 1.0: A modest JavaScript framework for the HTML you already have. Make your majestic monolith sparkle with the best in JavaScript sprinkles ✨❤️🚀

Traduzir do inglês

**Sprinkle your HTML with controller, target, and action attributes:**

```
<!--HTML from anywhere-->
<div data-controller="hello">
  <input data-target="hello.name" type="text">
  <button data-action="click->hello#greet">
    Greet
  </button>
  <span data-target="hello.output">
  </span>
</div>
```

**Write a compatible controller and watch Stimulus bring it to life:**

```
// hello_controller.js
import { Controller } from "stimulus"

export default class extends Controller {
  static targets = [ "name", "output" ]

  greet() {
    this.outputTarget.textContent =
      `hello, ${this.nameTarget.value}!`
  }
}
```

**Stimulus 1.0: A modest JavaScript framework for the HTML you already have**

We write a lot of JavaScript at Basecamp, but we don't use it to create "JavaScript applications" in the contemporary sense. All our...

[m.signalvnoise.com](http://m.signalvnoise.com)

18:50 - 30 de jan de 2018

352 Retweets 983 Curtidas



26

352

983







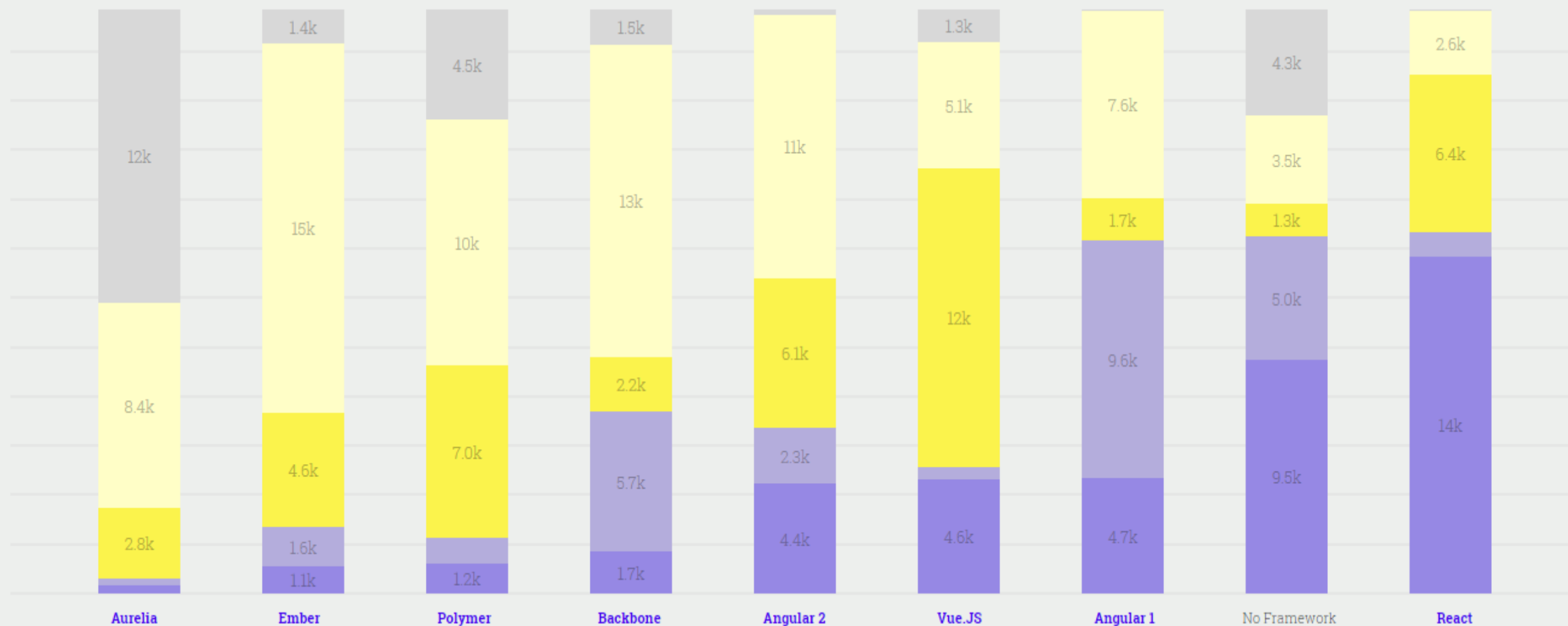
A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES

# Libraries Results

Per-library survey results.

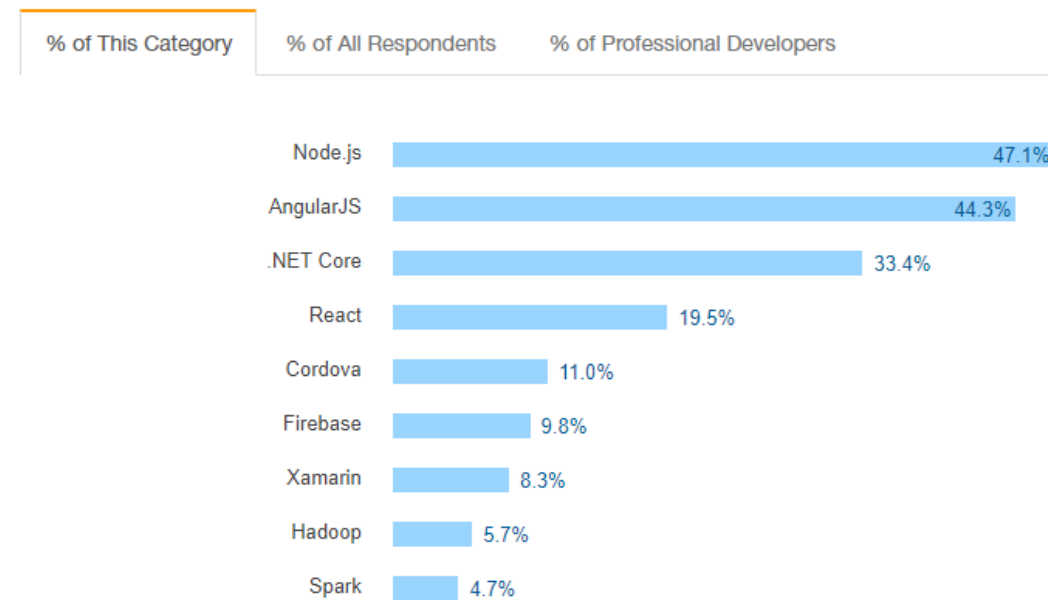
<https://stateofjs.com/2017/front-end/results/>

■ I've never heard of it    ■ I've HEARD of it, and am NOT interested    ■ I've HEARD of it, and WOULD like to learn it    ■ I've USED it before, and would NOT use it again    ■ I've USED it before, and WOULD use it again



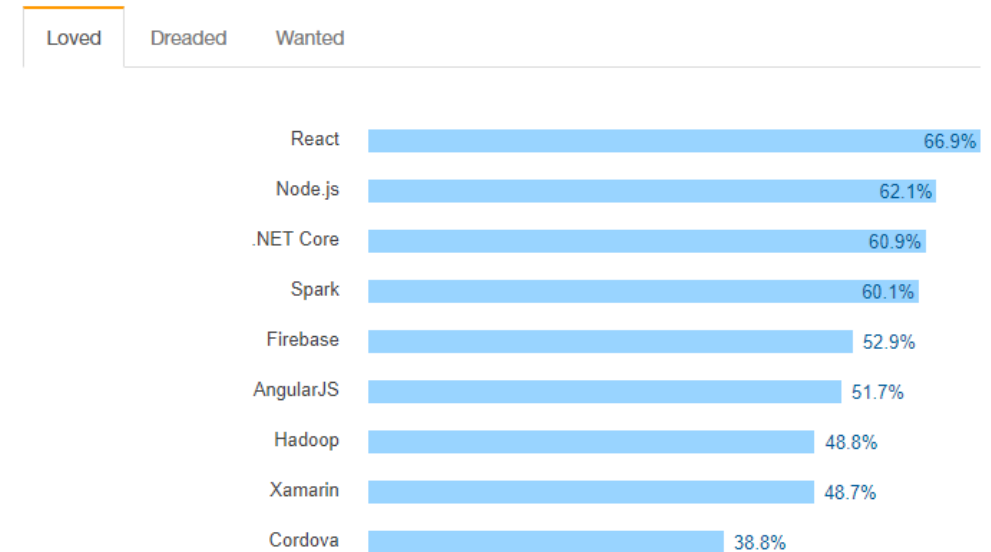
<https://insights.stackoverflow.com/survey/2017#technology-frameworks-libraries-and-other-technologies>

### Frameworks, Libraries, and Other Technologies



<https://insights.stackoverflow.com/survey/2017#technology-most-loved-dreaded-and-wanted-frameworks-libraries-and-other-technologies>

### Most Loved, Dreaded, and Wanted Frameworks, Libraries and Other Technologies



% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

ASP.NET

@re



# OLÁ! 🇧🇷

Software Engineer at Jetabroad;  
Agile Advocate (Scrum/Lean);  
Digital Entrepreneur;  
JavaScript Lover;  
C# Lover;  
[Ex-Lambda3](#) from Brazil;  
Crazy `Farang`



[@kidchenko](#)



[github.com/kidchenko](https://github.com/kidchenko)





## Book Cheap Flights Online

From anywhere in the world to anywhere else

☒ Return ☐ One way ☐ Multi-city & Stopover

Depart from

 Sydney, NSW, Australia (SYD)

Arrive into

 Singapore Changi (SIN)

Depart date

 06 Mar 2018

Return date

 13 Mar 2018

Passengers

1 Adult 0 Child 0 Infant

[+ Show options \(cabin class, airlines & currency\)](#)

**Search flights**

PayPal VISA

MasterCard

American Express

Debit Card

PAYMENT OPTIONS WILL VARY DEPENDING ON ITINERARY



Accredited Agent



## Explore with Jetabroad

We are an award winning online travel agent based in Sydney, Australia. We specialise in providing you with only the most competitive international airfares through our superior knowledge, expertise and technology. At Jetabroad we help you easily and affordably book flights that lead to unforgettable journeys.

## Sourced from over 500 airlines.



We offer domestic and international flight options with the widest possible range choice. This means you can mix and match to create a perfect itinerary that offers value and convenience.



**Over 500,000 travelers**  
from all over the world have  
booked with us.



**Jetabroad  
Customer  
Support**

[Click here for help  
with your booking](#)



[Like us on facebook](#)

*Diversity is the mix, inclusion is respecting the difference*

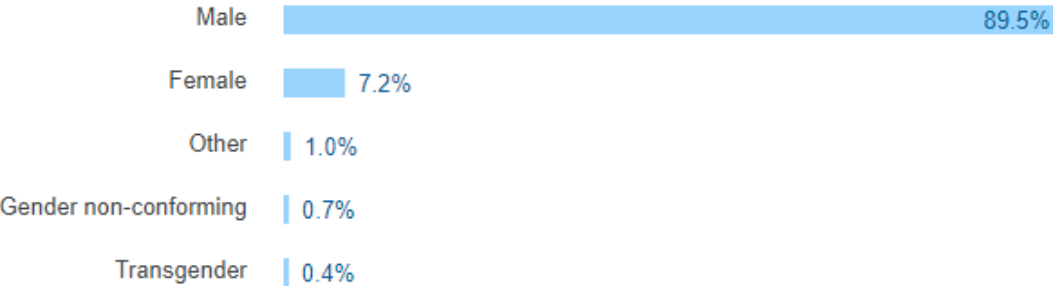




# Demographics

## Gender

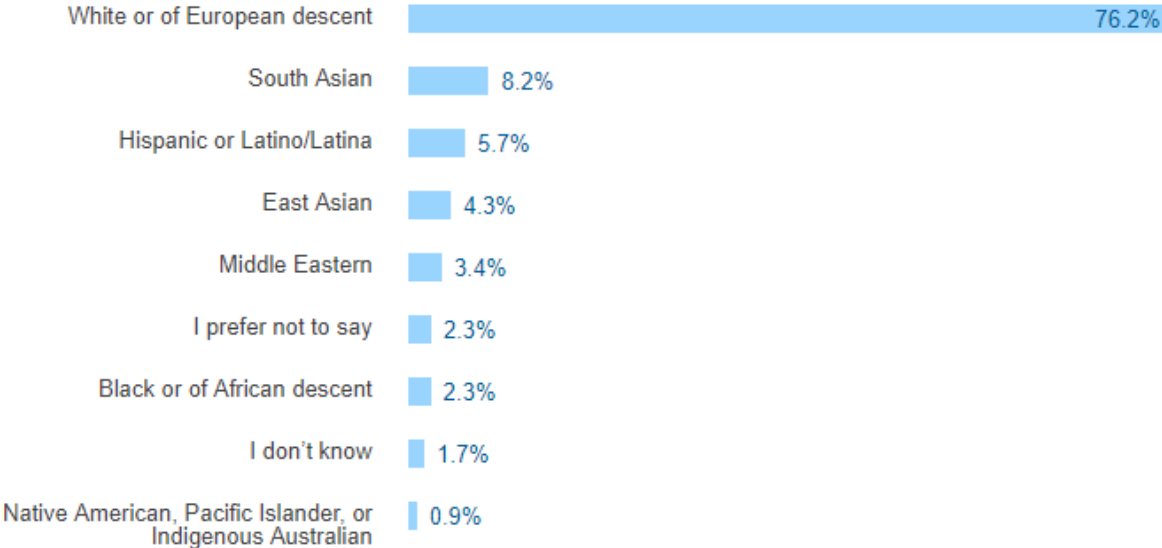
All Respondents Professional Developers Students



26,762 responses

## Ethnicity

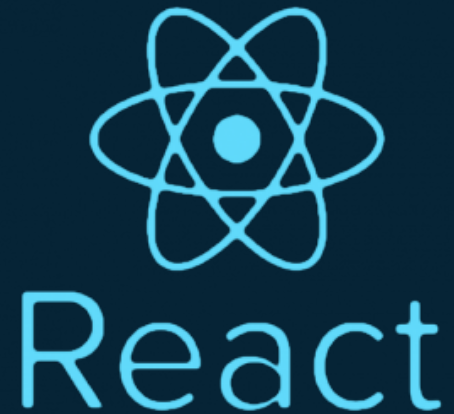
All Respondents Professional Developers Students



24,652 responses; select all that apply

# Building SPA's using

ASP. NET Core +





# ReactJS.NET

REACT ♥ C# AND ASP.NET MVC

# Nops!

ReactJS.NET makes it easier to use Facebook's [React](#) and [JSX](#) in C# and other .NET languages, focusing specifically on ASP.NET MVC (although it also works in other environments). It supports both ASP.NET 4 (with MVC 4 or 5), and ASP.NET Core MVC. It is cross-platform and can run on Linux via Mono or .NET Core. Take a look at the [tutorial](#) to see how easy it is to get started with React and ReactJS.NET.

*Latest news: [ReactJS.NET 3.0 - .NET Core and its small tweaks](#) (October 9, 2016)*

## On-the-fly JSX to JavaScript compilation

Simply name your file with a `.jsx` extension and link to the file via a `script` tag.

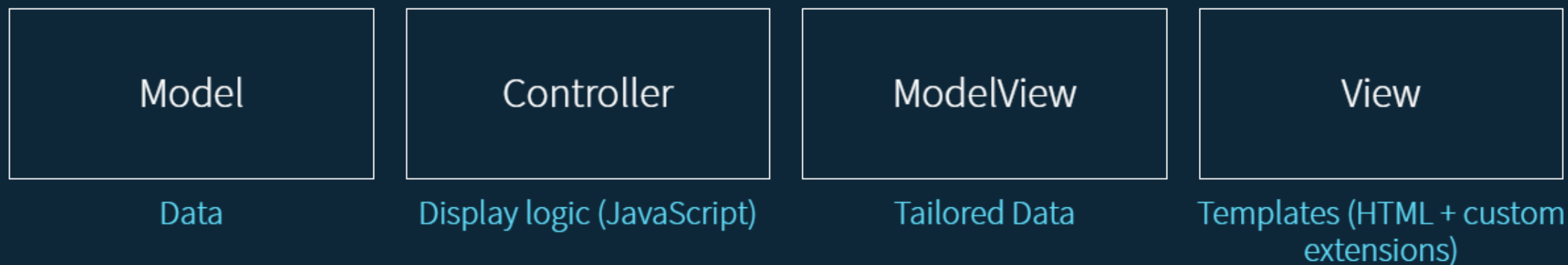
The files will automatically be

```
// /Scripts/HelloWorld.jsx
var HelloWorld = React.createClass({
  render: function() {
    return <div>Hello world!</div>;
  }
});
```



# Traditional MV\* Frameworks

## Separation of Concerns



Controller, ModelView, and View **are** coupled:  
when you change one, you often have to change the others

# Angular JS - MV\*

## Model

```
var myDetails = {  
  "firstname": "Srinivas",  
  "lastname": "Nagaram",  
  "title": "UI Developer"  
};
```

Data

## View

```
<tr>  
  <td>{{myDetails.firstname}}</td>  
  <td>{{myDetails.lastname}}</td>  
  <td>{{myDetails.title}}</td>  
</tr>
```

HTML

## \* Whatever

Angular JS does not follow any one of below methods . For this reason we will refer this framework as MV\* implementation.

Controller  
ViewModel  
Presenter

Logic

# React

A JavaScript library for building user interfaces

[Get Started](#)[Take the Tutorial >](#)

## Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes.

Declarative views make your code more predictable and easier to debug.

## Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs.

Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

## Learn Once, Write Anywhere

We don't make assumptions about the rest of your technology stack, so you can develop new features in React without rewriting existing code.

React can also render on the server using Node and power mobile apps using [React Native](#).

<https://reactjs.org/>



# ASP.NET CORE 2.0



```
C:\kidchenko\asp-net-core-react ^ master ≡/⚡x  
▶ dotnet --version  
2.1.4
```

<https://dot.net>

C:\kidchenko\asp-net-core-react\EmptyWeb ^ master ↑1 ⚡

► dotnet new

Usage: new [options]

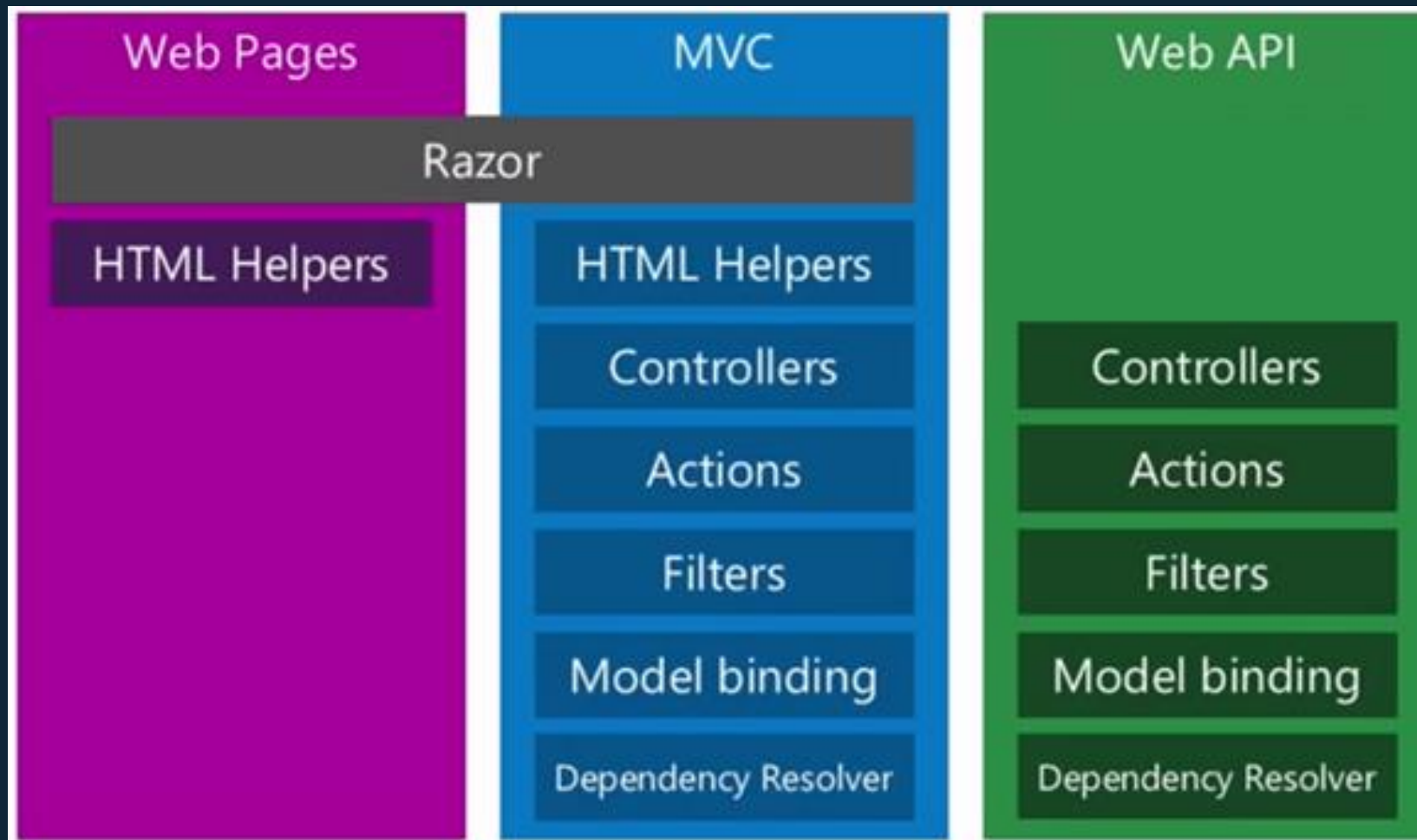
#### Options:

- h, --help Displays help for this command.
- l, --list Lists templates containing the specified name. If no name is specified, lists all templates.
- n, --name The name for the output being created. If no name is specified, the name of the current directory is used.
- o, --output Location to place the generated output.
- i, --install Installs a source or a template pack.
- u, --uninstall Uninstalls a source or a template pack.
- type Filters templates based on available types. Predefined values are "project", "item" or "other".
- force Forces content to be generated even if it would change existing files.
- lang, --language Specifies the language of the template to create.

Templates	Short Name	Language	Tags
Console Application	console	[C#], F#, VB	Common/Console
Class library	classlib	[C#], F#, VB	Common/Library
Unit Test Project	mstest	[C#], F#, VB	Test/MSTest
xUnit Test Project	xunit	[C#], F#, VB	Test/xUnit
ASP.NET Core Empty	web	[C#], F#	Web/Empty
ASP.NET Core Web App (Model-View-Controller)	mvc	[C#], F#	Web/MVC
ASP.NET Core Web App	razor	[C#]	Web/MVC/Razor Pages
ASP.NET Core with Angular	angular	[C#]	Web/MVC/SPA
ASP.NET Core with React.js	react	[C#]	Web/MVC/SPA
ASP.NET Core with React.js and Redux	reactredux	[C#]	Web/MVC/SPA
ASP.NET Core Web API	webapi	[C#], F#	Web/WebAPI
global.json file	globaljson		Config
NuGet Config	nugetconfig		Config
Web Config	webconfig		Config
Solution File	sln		Solution
Razor Page	page		Web/ASP.NET
MVC ViewImports	viewimports		Web/ASP.NET
MVC ViewStart	viewstart		Web/ASP.NET

#### Examples:

```
dotnet new mvc --auth Individual
dotnet new nugetconfig
dotnet new --help
```



SPA

EXPLORER

▲ OPEN EDITORS

📄 Welcome

📄 *C# Program.cs* A

▲ EMPTYWEB

▸ bin ●

▸ obj ●

▸ wwwroot

📄 EmptyWeb.csproj A

📄 C# Program.cs A

📄 C# Startup.cs A

web root directory ←

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

0 references

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    if (env.IsDevelopment())
    {
        app.UseDeveloperExceptionPage();
    }
    app.UseStaticFiles();

    // app.Run(async (context) =>
    // {
    //     await context.Response.WriteAsync("Hello World!");
    // });
}
```

EXPLORER

OPEN EDITORS

Startup.cs

EMPTYWEB

bin

obj

wwwroot

index.html

EmptyWeb.csproj

Program.cs

Startup.cs

Static files:  
HTML, CSS,  
JavaScript

```
<!DOCTYPE html>
<html>
<head>
<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
</head>
<body>
<div id="root"></div>
<script type="text/babel">

  ReactDOM.render(
    <h1>Hello, world!</h1>,
    document.getElementById('root')
  );

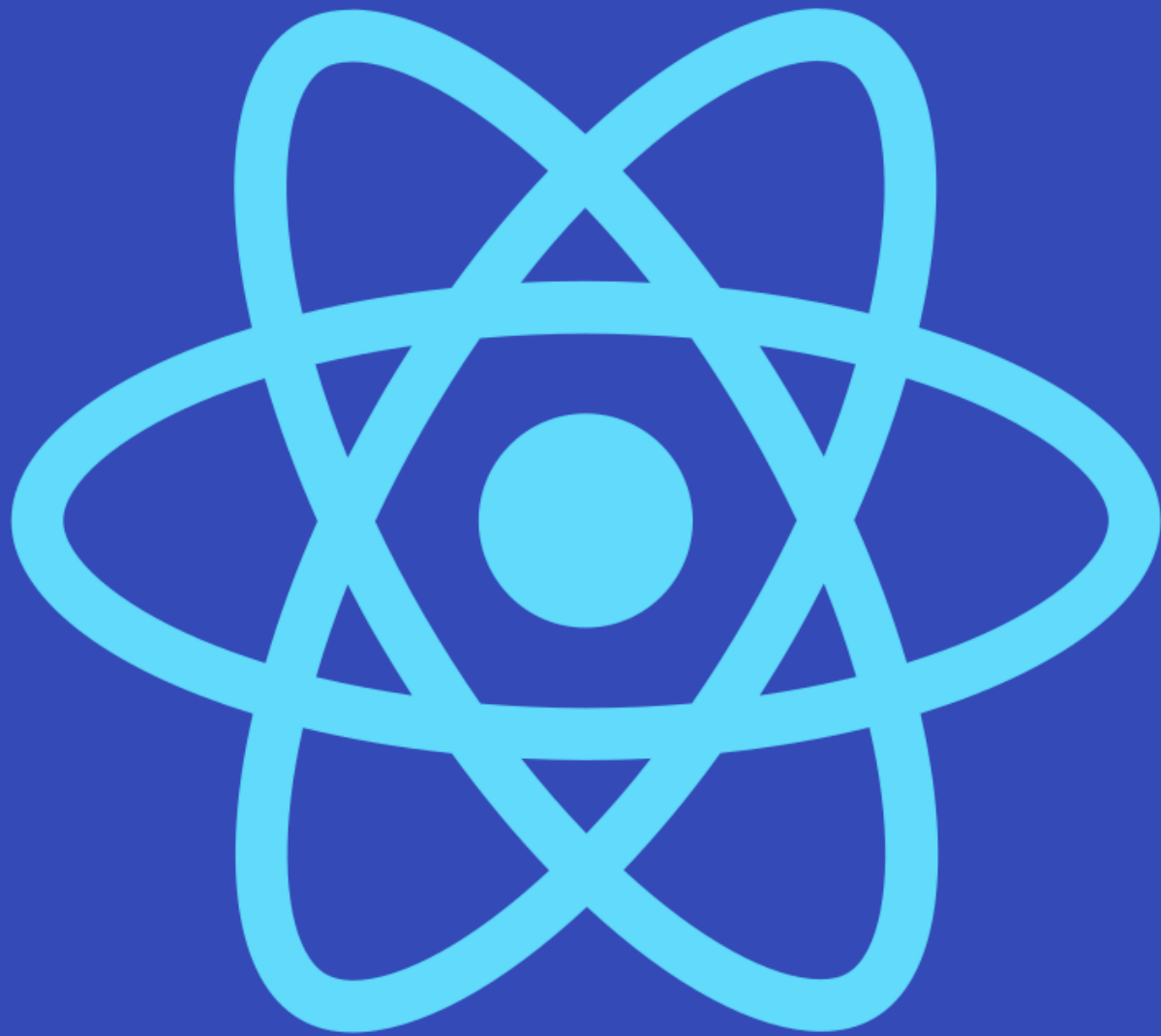
</script>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<script src="https://unpkg.com/react@16/umd/react.development.js"></script>
<script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js"></script>
<script src="https://unpkg.com/babel-standalone@6.15.0/babel.min.js"></script>
</head>
<body>
<div id="root"></div>
<script type="text/babel">

  ReactDOM.render(
    <h1>Hello, world!</h1>,
    document.getElementById('root')
  );

</script>
</body>
</html>
```



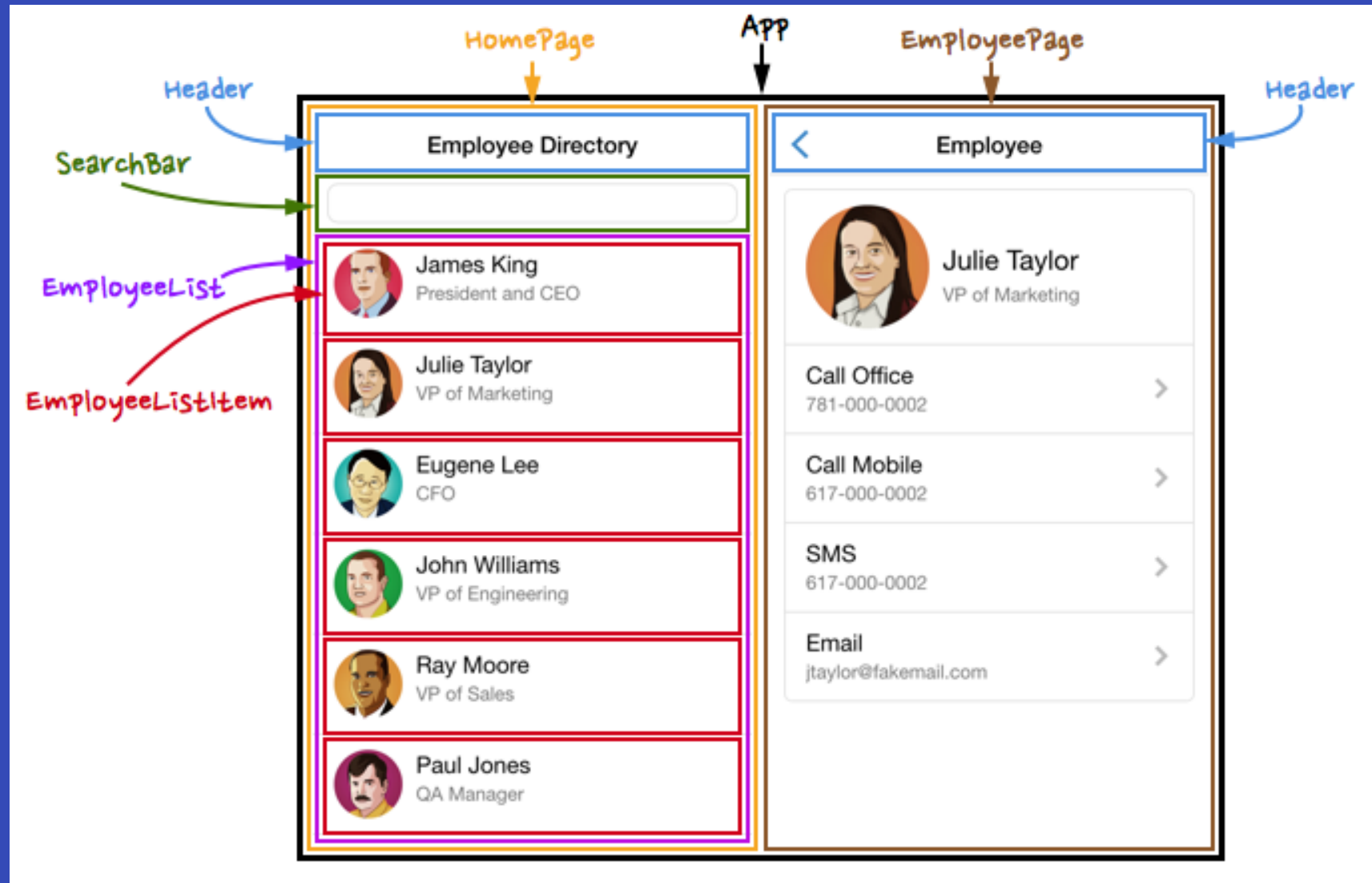


# A JavaScript Library

For creating  
User  
Interfaces

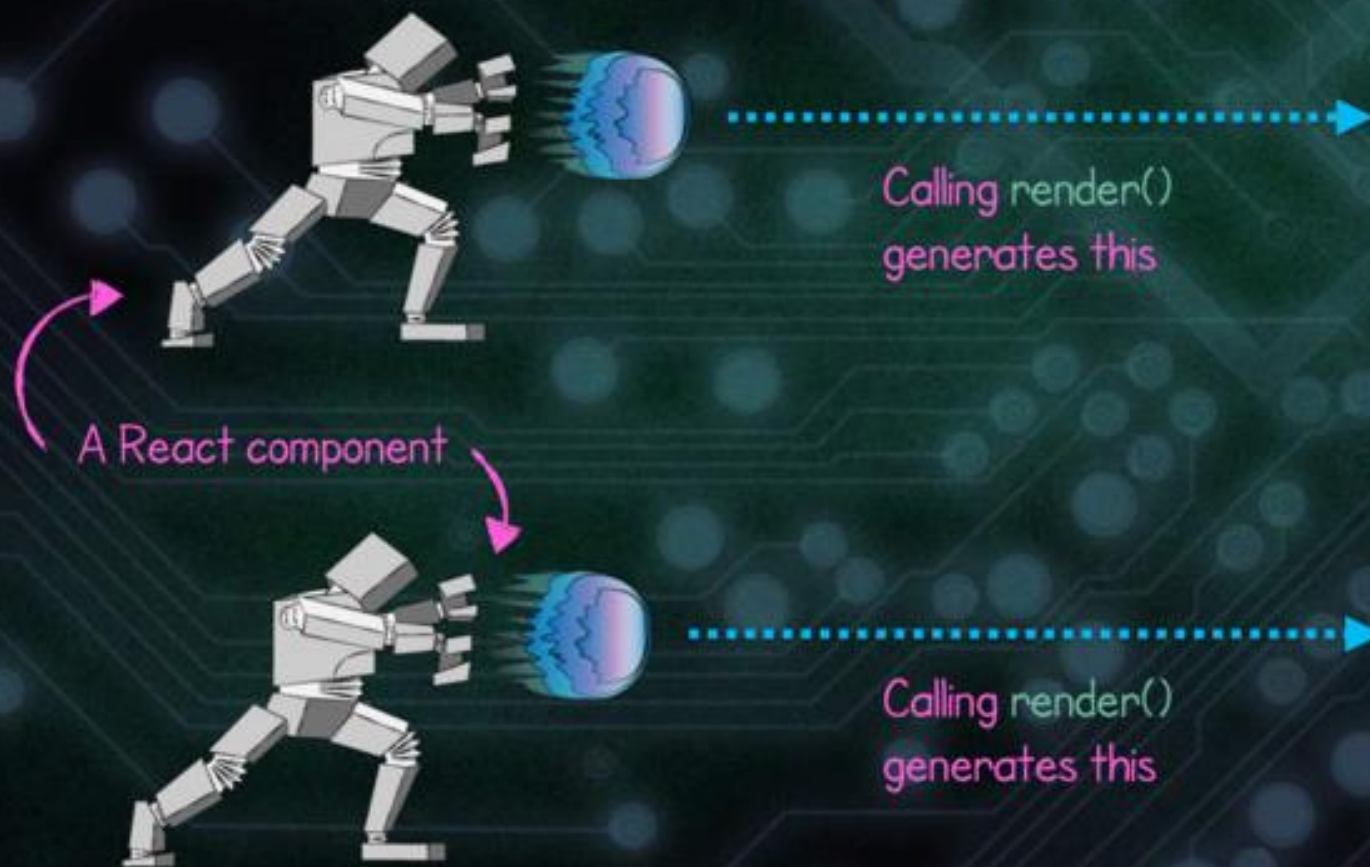
# Using Components

# Components



# What Is a React Component?

A component in React works similarly to JavaScript functions: It generates an output every time it is invoked.



## Output #1

```
<div>
  <p>Good Morning</p>
  <p>10:45AM</p>
</div>
```

## Output #2

```
<div>
  <p>Good Morning</p>
  <p>10:55AM</p>
</div>
```

A Component  
do 3 things:

# A Component do 3 things:

- Render
- Receive Props
- Maintain State



JSX

# Controllers

[Route("api/[controller]")]

0 references

public class TodosController : Controller

{

// GET api/todos

[HttpGet]

0 references

public IEnumerable<Todo> Get()

{

return new Todo[] { new Todo { Id = 1, Text = "value1" } , new Todo { Id = 2, Text = "value2" } };

}

// GET api/todos/5

[HttpGet("{id}")]

0 references

public Todo Get(int id)

{

return new Todo{ Id = id, Text = "Todo" };

}

// POST api/todos

[HttpPost]

0 references

public IActionResult Post([FromBody]Todo todo)

{

return CreatedAtAction("Get", new { id = todo.Id }, todo);

}

Routing

Data coming  
from JSON

Return code 201  
and add  
location header

Status: 201 Created Time: 196 ms

content-type → application/json; charset=utf-8

date → Mon, 05 Feb 2018 14:29:06 GMT

location → http://localhost:5000/api/Todos/1

server → Kestrel

transfer-encoding → chunked

{ REST }

Method	Route	Request body (JSON)	Response body (JSON)	Success (HTTP Status)	Fail (HTTP Status)	Description
GET	/item/		<code>{"items":["Init item",...]}</code>	200		Get all items
GET	/item/{index}		<code>{"item":"Init item"}</code>	200	404	Get item by index
POST	/item/	<code>{"item":"New item"}</code>		201	400	Add item
PUT	/item/{index}	<code>{"item":"Update item"}</code>	<code>{"oldItem":"Old item"}</code>	200	400/404	Update item by index
DELETE	/item/{index}		<code>{"removeItem":"Remove item"}</code>	200	404	Delete item by index

## My API

### Todos

[Show/Hide](#) | [List Operations](#) | [Expand Operations](#)

GET	/api/Todos
POST	/api/Todos
DELETE	/api/Todos/{id}
GET	/api/Todos/{id}
PUT	/api/Todos/{id}

[ BASE URL: / , API VERSION: v1 ]

```
dotnet add package Swashbuckle.AspNetCore --version 1.1.0
```

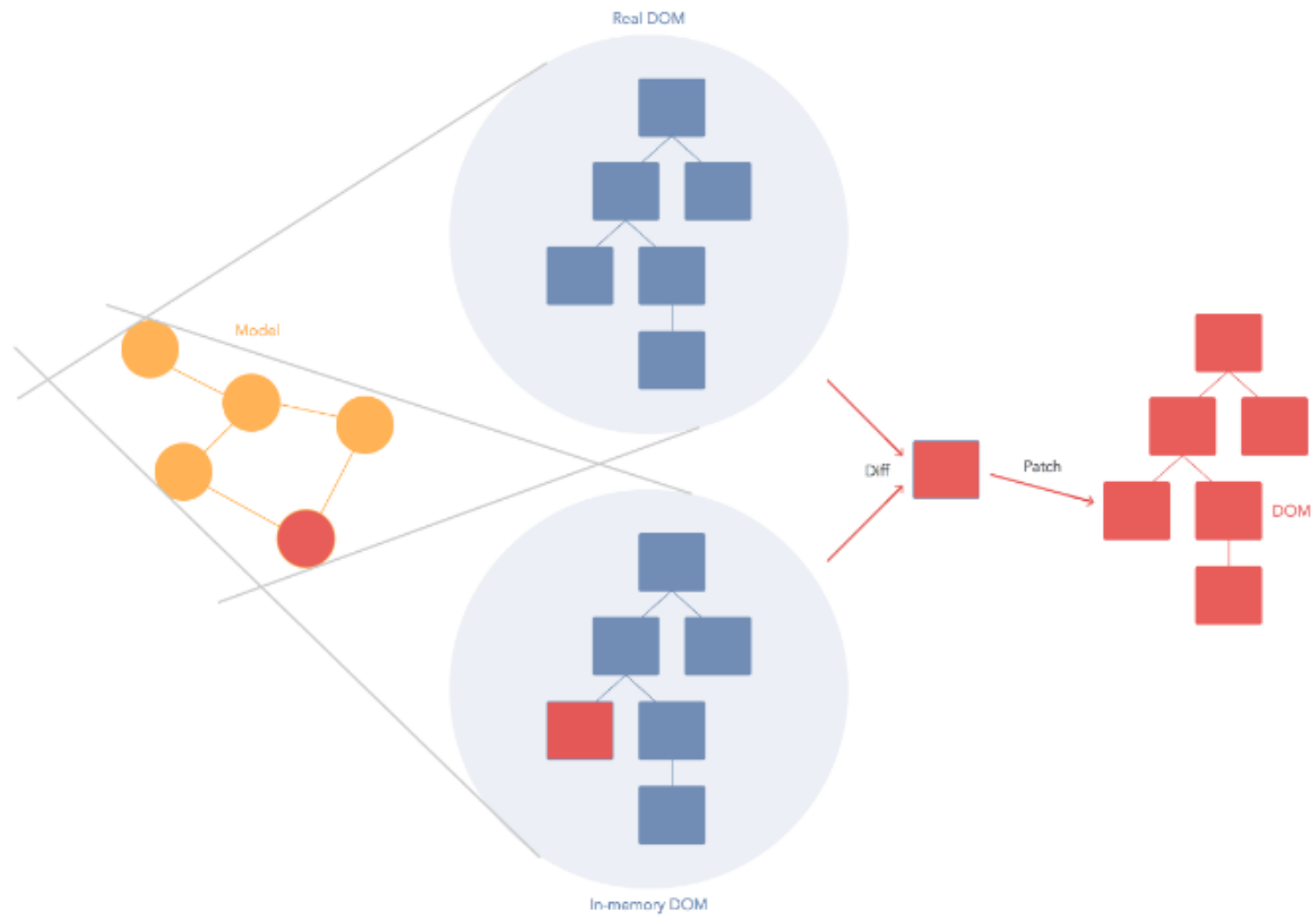


UNIDIRECTIONAL  
*data flow*

# Virtual DOM



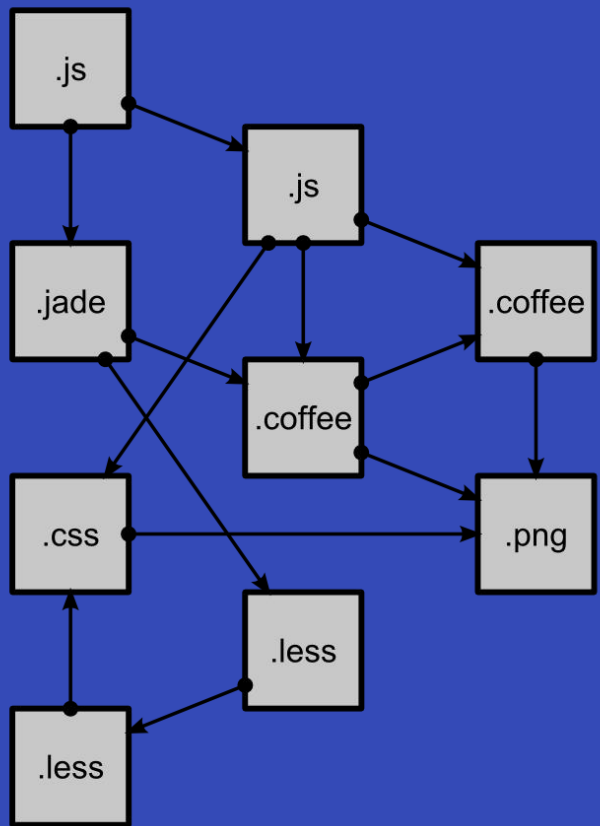




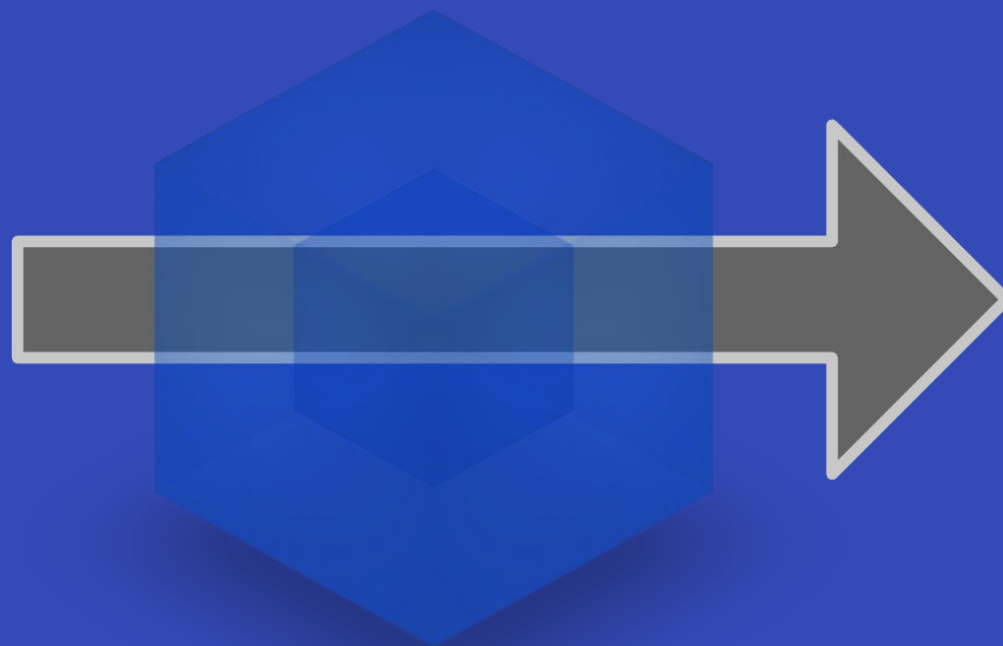
# Composition

```
var HomePage = React.createClass({  
  render: function () {  
    return (  
      <div>  
        <Header />  
        <SearchBar />  
        <EmployeeList />  
      </div>  
    );  
  }  
});
```

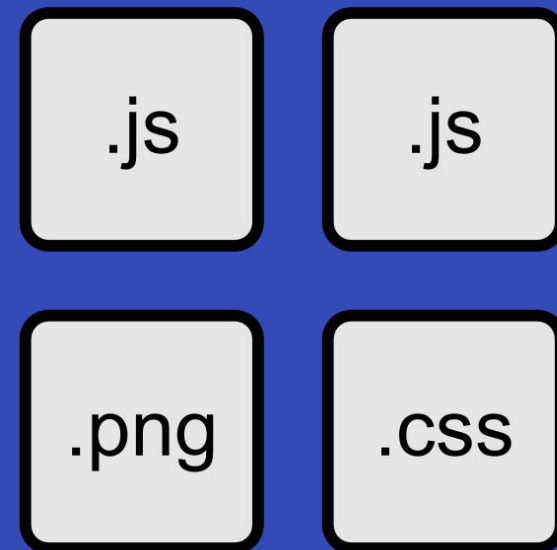
# Tooling



modules  
with dependencies



**webpack**  
MODULE BUNDLER



static  
assets



+



Webpack



JavaScript



ES6

React

*BABEL*

Babel

Final  
ES5

JavaScript  
File

```
~/Desktop — matt@Matthews-MacBook-Pro-2 — node /usr/local/bin/create-react-app my-s...
Last login: Wed Dec 20 08:04:38 on ttys000
[14:51:41] ~
[// ♥ cd Desktop/
[14:51:46] Desktop
[// ♥ create-react-app my-site

Creating a new React app in /Users/matt/Desktop/my-site.
|
```



# Thank you!

Obrigado!

Gracias!

Khob khun krab!

# Questions?



 [@kidchenko](https://twitter.com/kidchenko)

 [github.com/kidchenko](https://github.com/kidchenko)