

REACT+ WORDPRESS

a match made in heaven



JOSH BROTON
@joshbroton || joshbroton.com



KANYE WEST @kanyewest · 1h

You have distracted from my creative process



30K

29K

...

HilariousGifts.com





**OWNER/FOUNDER:
BROTON DIGITAL CONSULTING**



**FOUNDER:
PRESTIGE CONFERENCE**



**CTO/JAVASCRIPT/FRONTEND:
VIDSCRIP, BASIN COMMERCE,
11 OTHER STARTUPS**

If I had a dollar for everytime I got distracted, I wish I had some ice cream.

someecards
user card





I don't have attention
deficit disorder...

I have this isn't
interesting enough to
pay attention to disorder.



som~~e~~ecards
user card

**BROTY
+ DECONGESTANTS
+ COFFEE
+ NERVES**



WHAT THIS TALK SHOULD BE

“

CONFERENCE SESSIONS SHOULD NEVER
AIM TO TEACH THE AUDIENCE
SOMETHING, BUT RATHER INSPIRE THEM
TO WANT TO LEARN.

- TED NEWARD

WHAT WE'RE COVERING

(I hear Jennifer
and Ann have
awesome
sessions)

01

WTF IS A SPA? WHY A SPA?

02

WHY / WHY NOT REACT?

03

REACT + WORDPRESS

01

WHAT IS A SPA?

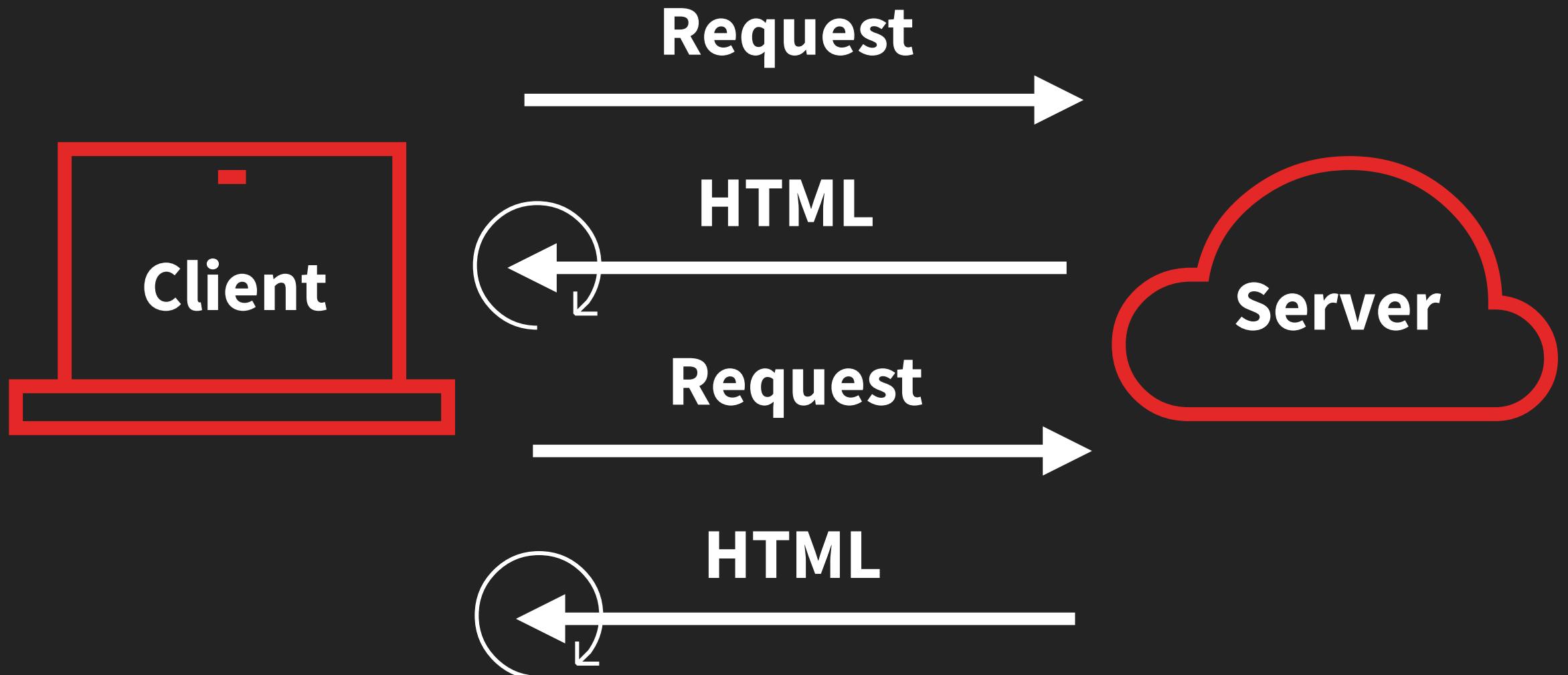


01

WHAT IS A SPA?

Single Page App

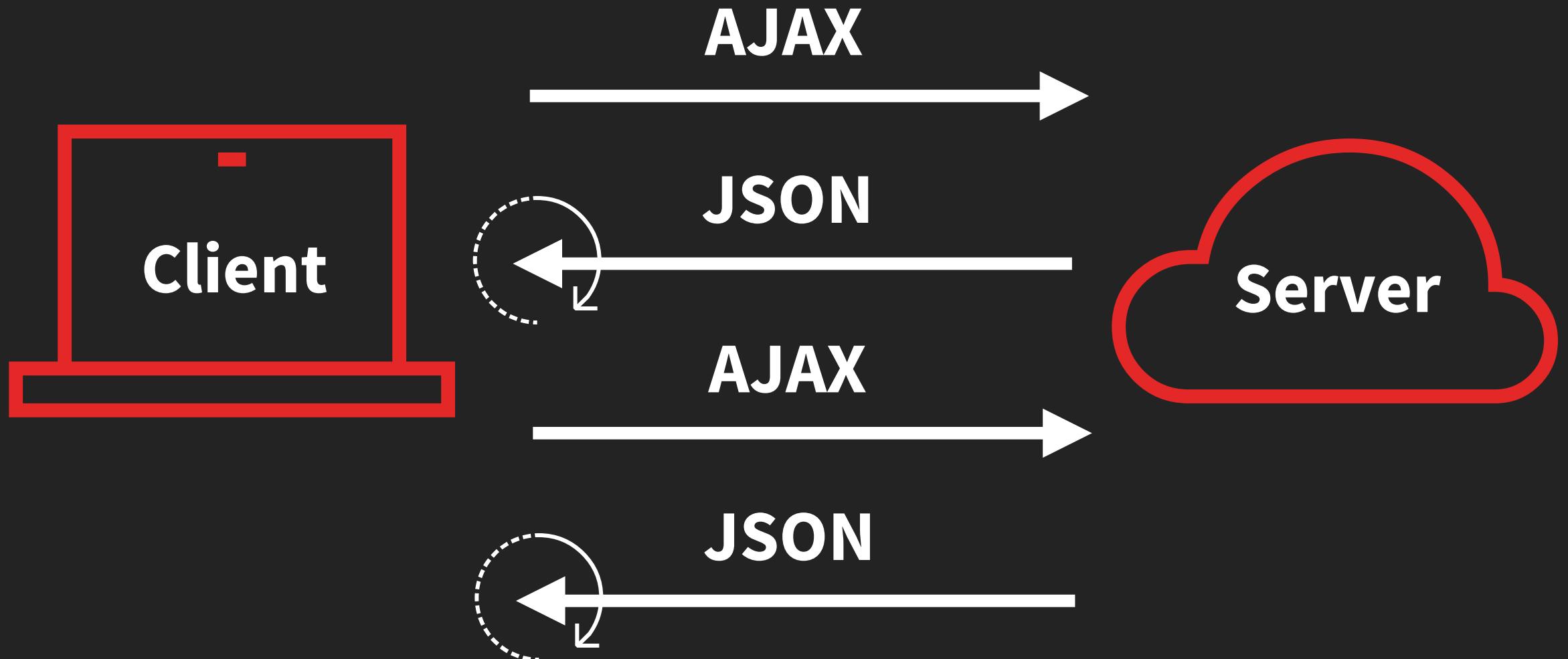
TYPICAL WEB SITE:



SINGLE PAGE APP:



SINGLE PAGE APP:



4GIFs.com



THE SPA ADVANTAGE



SCALES WITH LARGE
DATA SETS



FASTER ON SUBSEQUENT
PAGE LOADS



FRONT-END CODE IS
EASIER TO TEST



OFFLOADS PROCESSING TO
CLIENTS



02

WHY REACT?

#yoloGangs

02

WHY REACT?



02

WHY REACT?

2014: THE YEAR OF VANILLA JAVASCRIPT



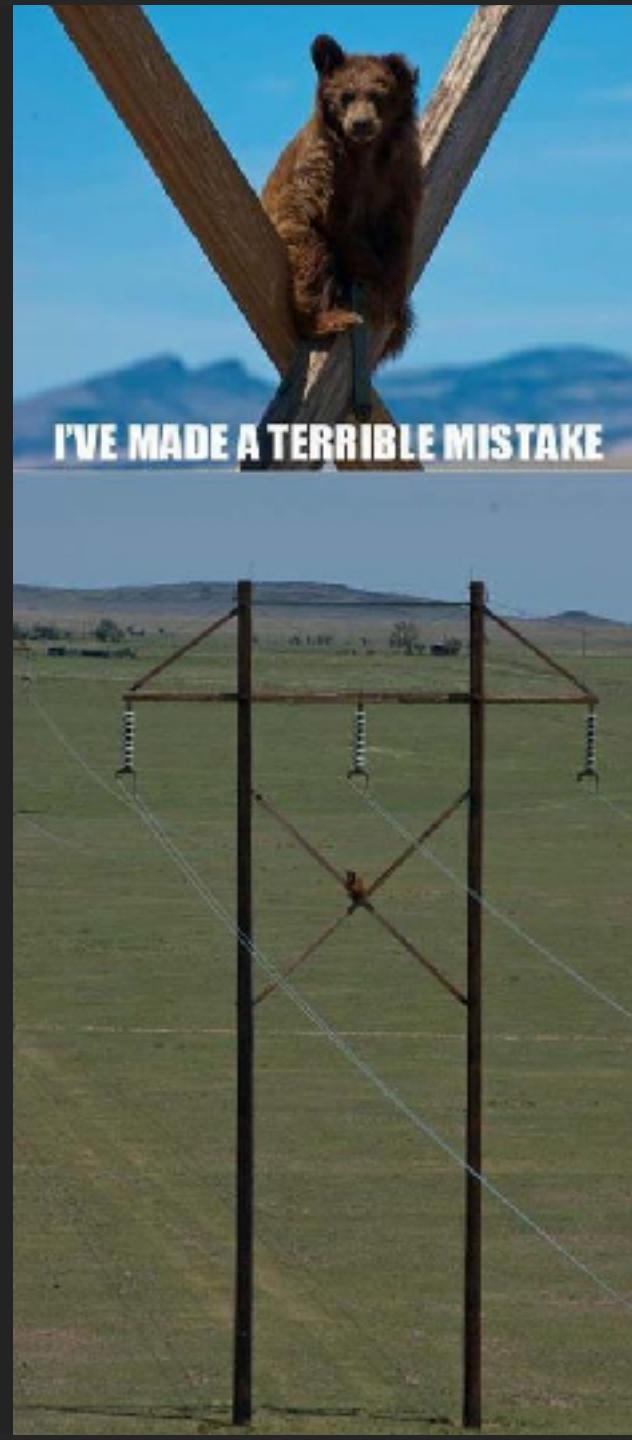
02

WHY REACT?

2017: THE YEAR OF THE FRAMEWORK

02

WHY REACT?



02

WHY REACT?



02

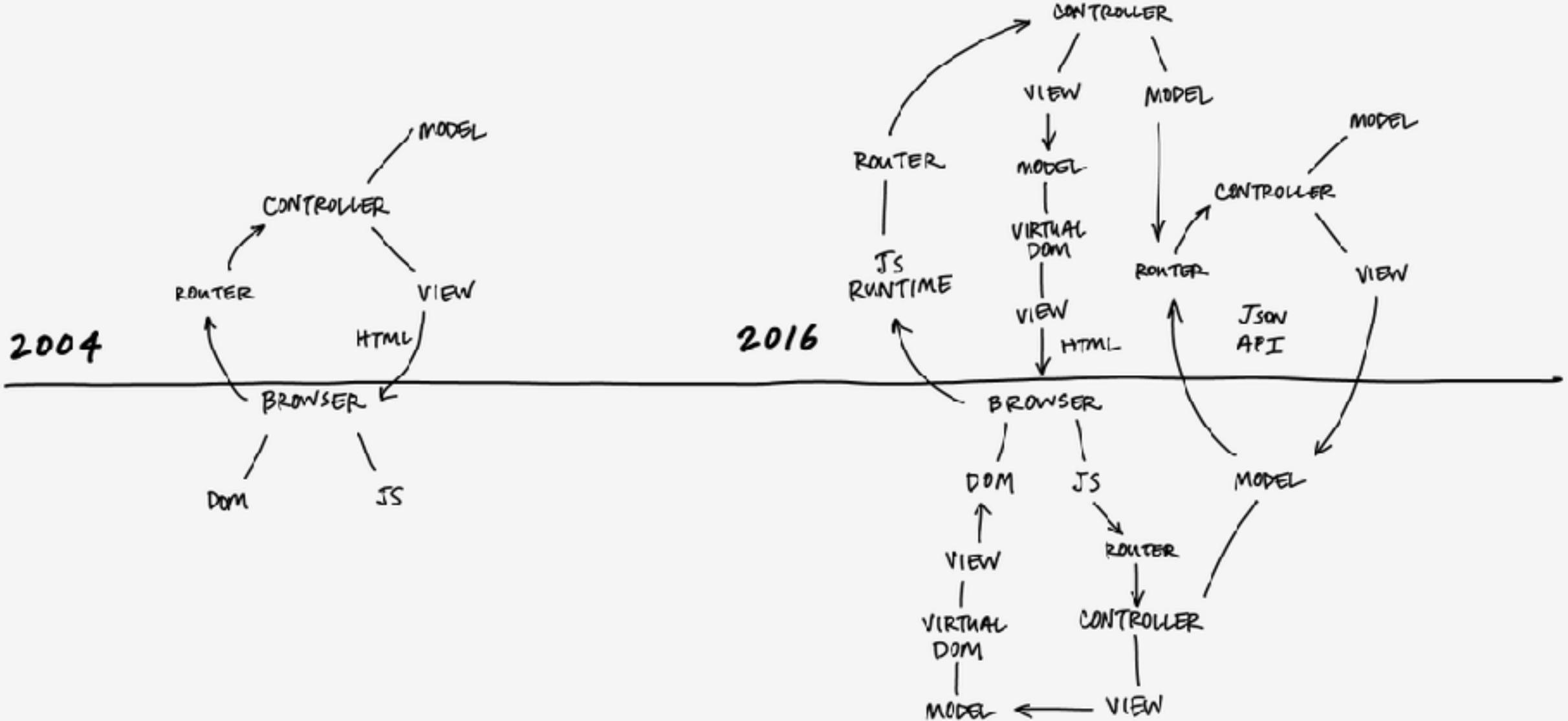
WHY REACT?



02

WHY REACT?







Ada Rose Edwards

@Lady_Ada_King



 Follow

The nice thing about the web is the diagram on the left hand side still applies if you want it to.



Sam Stephenson @sstephenson

12 years of progress?

RETWEETS

20

LIKES

28



10:55 AM - 10 May 2016



...



02

WHY REACT?



JSX IS OUTSTANDING

02

WHY REACT?

(

```
<article className="some-blog-post">
  <header className="post_header">
    <h1>
      {post.title.replace(' ', '')}
    </h1>
    <aside className="byline">
      {generatePostAuthorLink(post.author)}
    </aside>
  </header>
  <div className="post_content">
    {post.content}
  </div>
  ...
</article>
```

)

A photograph of a brown dog lying on a paved surface. A white paper coffee cup with a textured sleeve is balanced precariously on the dog's head. In the background, a person's legs in jeans and sneakers are visible.

Get down from there coffee

you do not
belong on
that dog

you are a
beverage

EPICPICSOFWIN.COM

02

WHY REACT?

(

```
<article className="some-blog-post">
  <header className="post_header">
    <h1>
      {post.title.replace(' ', '')}
    </h1>
    <aside className="byline">
      {generatePostAuthorLink(post.author)}
    </aside>
  </header>
  <div className="post_content">
    {post.content}
  </div>
  ...
</article>
```

)

02

WHY REACT?

```
(  
  <article className="some-blog-post">  
    <PostHeader title={post.title} author={post.author} />  
    <PostContent content={post.content} />  
    <PostFooter categories={post.categories} />  
  </article>  
)
```

02

WHY REACT?

```
(  
  <Post />  
)
```

02

WHY REACT?

```
(  
  this.state.posts.map((singlePost) => {  
    return <Post post={singlePost} />  
  }  
)
```

02

WHY REACT?

```
render() {  
  let { singlePost } = this.props;  
  return (  
    <article className="some-blog-post">  
      <PostHeader title={singlePost.title} author={singlePost.author} />  
      <PostContent content={singlePost.content} />  
      <PostFooter categories={singlePost.categories} />  
    </article>  
  )  
}
```

02

WHY REACT?

```
(  
  {this.state.showPost === true ? (  
    <article className="some-blog-post">  
      <PostHeader title={post.title} author={post.author} />  
      <PostContent content={post.content} />  
      <PostFooter categories={post.categories} />  
    </article>  
  ) : null }  
)
```



a pile of moss
@whale_eat_squid

 Follow



I really wish react people would stop saying you can use it without jsx. You can, but you can also stir boiling soup with your bare hand

RETWEETS

3

LIKES

11



2:46 PM - 24 Apr 2017

 1

  3

 11

02

WHY REACT?



JSX IS OUTSTANDING



REACT SYNTAX IS SIMPLE

02

WHY REACT?

```
(  
  this.state.posts.map((singlePost) => {  
    return <Post post={singlePost} />  
  })  
)  
(  
  <article className="some-blog-post">  
    <PostHeader title={post.title} author={post.author} />  
    <PostContent content={post.content} />  
    <PostFooter categories={post.categories} />  
  </article>  
)
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  componentDidMount() { }  
  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  componentDidMount() { }  
  componentWillReceiveProps(nextProps) { }  
  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  componentDidMount() { }  
  componentWillReceiveProps(nextProps) { }  
  shouldComponentUpdate(nextProps,nextState) { //can return false }  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  componentDidMount() { }  
  componentWillReceiveProps(nextProps) { }  
  shouldComponentUpdate(nextProps, nextState) { //can return false }  
  componentWillUpdate(nextProps, nextState) { }  
}  
  
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  componentDidMount() { }  
  componentWillReceiveProps(nextProps) { }  
  shouldComponentUpdate(nextProps, nextState) { //can return false }  
  componentWillUpdate(nextProps, nextState) { }  
  componentDidUpdate(prevProps, prevState)  
  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  componentDidMount() { }  
  componentWillReceiveProps(nextProps) { }  
  shouldComponentUpdate(nextProps, nextState) { //can return false }  
  componentWillUpdate(nextProps, nextState) { }  
  componentDidUpdate(prevProps, prevState)  
  componentWillUnmount() { }  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  componentWillMount() { }  
  componentDidMount() { }  
  componentWillReceiveProps(nextProps) { }  
  shouldComponentUpdate(nextProps, nextState) { //can return false }  
  componentWillUpdate(nextProps, nextState) { }  
  componentDidUpdate(prevProps, prevState) { }  
  componentWillUnmount() { }  
  
  render() { }  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    return (  
      <header className="post_header">  
        <a href={this.props.permalink} target="_blank">  
          <h1>  
            {this.props.title}  
          </h1>  
        </a>  
        <aside className="byline">  
          {this.props.author.name}  
        </aside>  
      </header>  
    )  
  }  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    return (  
      <header className="post_header">  
        <a href={this.props.permalink} target="_blank">  
          <h1>  
            {this.props.title}  
          </h1>  
        </a>  
        <aside className="byline"  
              onClick={this.showAuthorModal.bind(this)}>  
          {this.props.author.name}  
        </aside>  
      </header>  
    )  
  }  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    return (  
      <header className="post_header">  
        <a href={this.props.permalink} target="_blank">  
          <h1>  
            {this.props.title}  
          </h1>  
        </a>  
        {this.props.author.id !== this.state.user.id ? (  
          <aside className="byline"  
            onClick={this.showAuthorModal.bind(this)}>  
            {this.props.author.name}  
          </aside>  
        ) : null }  
      </header>  
    )
```

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    return (  
      <header className="post_header">  
        <a onClick={this.props.openPostInIframe.bind(this)}>  
          <h1>  
            {this.props.title}  
          </h1>  
        </a>  
        {this.props.author.id !== this.state.user.id ? (  
          <aside className="byline"  
            onClick={this.showAuthorModal.bind(this)}>  
            {this.props.author.name}  
          </aside>  
        ) : null }  
      </header>  
    )
```

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    return (  
      <header className="post__header">  
        <h1 role="button"  
          onClick={this.props.openPostIniFrame.bind(this)}>  
          {this.props.title}  
        </h1>  
        {this.props.author.id !== this.state.user.id ? (  
          <aside className="byline"  
            onClick={this.showAuthorModal.bind(this)}>  
            {this.props.author.name}  
          </aside>  
        ) : null }  
      </header>  
    )  
  }  
}
```

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    return (  
      <header className="post__header">  
        <h1 role="button"  
          onClick={this.props.openPostIniFrame.bind(this)}  
          className={classNames({'read-title': this.hasRead()})}>  
          {this.props.title}  
        </h1>  
        {this.props.author.id !== this.state.user.id ? (  
          <aside className="byline"  
            onClick={this.showAuthorModal.bind(this)}>  
            {this.props.author.name}  
          </aside>  
        ) : null }  
      </header>  
    )
```

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    if(this.props.isFetchingPosts === true) {  
      return <LoadingSpinner />  
    } else {  
      return (  
        <header className="post__header">  
          <h1 role="button"  
            onClick={this.props.openPostIniFrame.bind(this)}  
            className={classNames({'read-title': this.hasRead()})}>  
            {this.props.title}  
          </h1>  
          {this.props.author.id !== this.state.user.id ? (  
            <aside className="byline"  
              onClick={this.showAuthorModal.bind(this)}>  
              {this.props.author.name}  
            </aside>
```

02

WHY REACT?

```
class Post extends Component {  
  render() {  
    if(this.props.isFetchingPosts === true) {  
      return <LoadingSpinner />  
    } else {  
      return (  
        <article className="some-blog-post">  
          <PostHeader title={post.title} author={post.author} />  
          <PostContent content={post.content} />  
          <PostFooter categories={post.categories} />  
        </article>  
      )  
    }  
  }  
}
```

02

WHY REACT?

```
class Post extends Component {  
  render() {  
    if(this.props.isFetchingPosts === true) {  
      return <LoadingSpinner />  
    } else {  
      return (  
        <Post post={singlePost} />  
      )  
    }  
  }  
}
```

02

WHY REACT?



JSX IS OUTSTANDING



REACT SYNTAX IS SIMPLE



REACT FAILS FAST AT COMPILE TIME

02

WHY REACT?

```
class PostHeader extends Component {  
  render() {  
    return (  
      <header className="post_header">  
        <a href={this.props.permalink} target="_blank">  
          <h1>  
            {this.props.title}  
          </h1>  
        </a>  
        <aside className="byline">  
          {this.props.author}  
        </aside>  
      </header>  
    )  
  }  
}
```

02

WHY REACT?

```
ERROR in ./src/components/content/PostHeader/PostHeader.jsx
Module build failed: SyntaxError: /Users/joshbroton/dev/git/vidscrip/site/src/components/content/PostHeader/PostHeader.jsx: Unexpected token (13:46)
 11 |         return (
 12 |             <header className="post__header">
> 13 |                 <a href={this.props.permalink} target="_blank">
   |                                         ^
 14 |                     <h1>
 15 |                         {this.props.title}
 16 |                     </h1>
```

02

WHY REACT?

```
● ▶ ./src/components/content/PostHeader/PostHeader.jsx      client:47
Module build failed: SyntaxError:
/Users/joshbroton/dev/git/vidscript/site/src/components/content/PostHeade
r/PostHeader.jsx: Unexpected token (13:46)
  11 |         return (
  12 |             <header className="post__header">
> 13 |                 <a href={this.props.permalink} target="_blank">
   14 |                     <h1>
   15 |                         {this.props.title}
   16 |                     </h1>
at Parser.pp.raise
```

02

WHY REACT?



02

WHY REACT?



HilariousGifs.com

02

WHY REACT?



JSX IS OUTSTANDING



REACT SYNTAX IS SIMPLE



REACT FAILS FAST AT COMPILE TIME



THE DEVELOPER TOOLS ARE INCREDIBLE

REACT DEVELOPER EXPERIENCE

Hot reloading, Chrome React Extension, Redux Dev Tools, JSX compile-time checks, verbose error messages

02

WHY REACT?



JSX IS OUTSTANDING



REACT SYNTAX IS SIMPLE



REACT FAILS FAST AT COMPILE TIME



THE DEVELOPER TOOLS ARE INCREDIBLE



SPEED. GLORIOUS, GLORIOUS SPEED

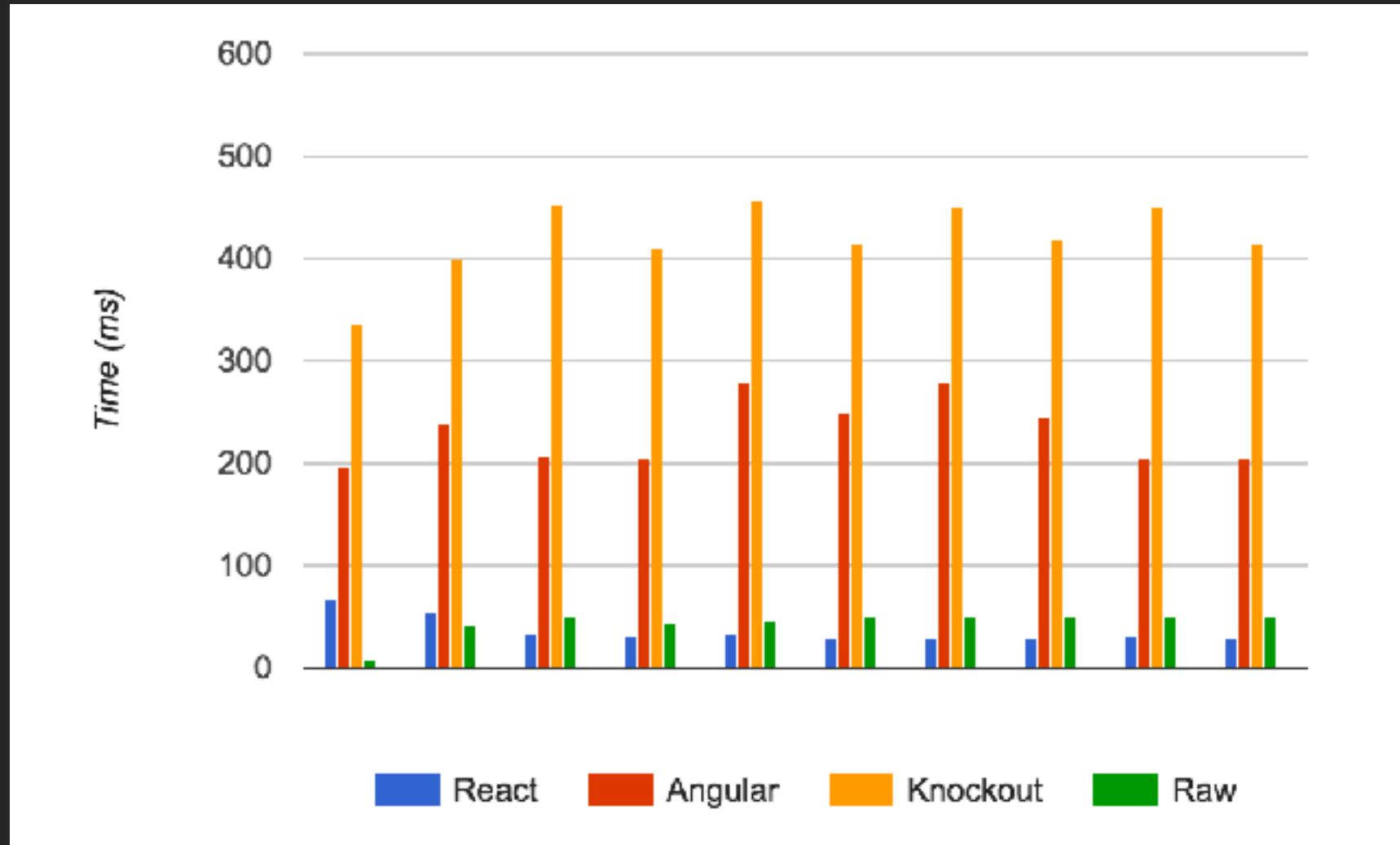
HOW FAST IS REACT?



HOW FAST IS REACT?

10x faster than Angular
18x faster than Knockout
3x faster than DOM
manipulation with jQuery

HOW FAST IS REACT?



HOW FAST IS REACT?

LIBRARY/FRAMEWORK SIZE:

Ember: 435k

Angular 1: 143k

Angular 4: 77k

React 0.15: 23k

02

WHY REACT?

The image shows a screenshot of a web page from CSS-Tricks. At the top left, there is a logo consisting of a grey asterisk (*) followed by the text "CSS-TRICKS" in a sans-serif font. To the right of the logo is a large, bold, white title: "Which Projects Need React? All Of Them!". Below the title, the author's name, "BY SACHA GREIF ON APRIL 26, 2017", is displayed in a smaller white font. Underneath the author's name is a small white icon of a person with a speech bubble, followed by the word "REACT". On the far left side of the page, there is a vertical black sidebar containing five white icons: a house (Home), a video camera (Video), a document (Document), a gear (Settings), and a photo (Image). The main content area has a light grey background, while the sidebar and some other sections have dark grey backgrounds.

03

REACT + WORDPRESS



03

REACT + WORDPRESS

THE WP REST API:
THE FUTURE OF WORDPRESS AND THE WEB

REACT + WORDPRESS

https://codex.wordpress.org/Version_4.7

REST API Content Endpoints

API endpoints for WordPress content. WordPress 4.7 comes with REST API endpoints for posts, comments, terms, users, meta, and settings. Content endpoints provide machine-readable external access to your WordPress site with a clear, standards-driven interface, paving the way for new and innovative methods of interacting with your site. [Check out the REST API reference.](#)





2000ish

```
[ - {  
    id: 1628,  
    date: "2016-04-01T02:32:21",  
    date_gmt: "2016-04-01T02:32:21",  
    - guid: {  
        rendered: "http://joshbroton.com/?p=1628"  
    },  
    modified: "2016-04-01T02:51:13",  
    modified_gmt: "2016-04-01T02:51:13",  
    slug: "so-you-want-to-change-the-gravity-forms-form-tag",  
    type: "post",  
    link: "http://joshbroton.com/so-you-want-to-change-the-gravity-forms-form-tag/",  
    - title: {  
        rendered: "So You Want to Change the Gravity Forms Form Tag#8230;"  
    },  
    - content: {  
        rendered: "<p>I had an interesting request from a client today. They use <a href='http://www.pardot.com/' onclick='__gaTracker('send', 'event', 'outbound-article', 'http://www.pardot.com/', 'Pardot');' target='_blank">Pardot</a> to integrate with the forms on their old website, and wanted to do the same with their new WordPress theme. We're using the always spectacular <a href='http://www.gravityforms.com/' onclick='__gaTracker('send', 'event', 'outbound-article', 'http://www.gravityforms.com/', 'Gravity Forms');'>Gravity Forms</a> for the forms on the site. </p> <p>I did the integrations, and we got an error.</p> <pre> The form enctype type appears to be incompatible with our pardot form handler according to this article: Wrong Enctype Pardot form handlers can only be integrated with forms using an empty enctype attribute or an enctype of application/x-www-form-urlencoded. Pardot doesn't accept an enctype of multipart/form-data. </pre> <p>I didn't realize (but now it makes perfect sense) that Gravity Forms defaults to enctype=&#8221;multipart/form-data&#8221;. I scoured the Gravity Forms settings, looking for a way to switch that to the correct enctype without any luck. I hit up a good friend of mine, <a href='https://twitter.com/projectgoboy' onclick='__gaTracker('send', 'event', 'outbound-article', 'https://twitter.com/projectgoboy', 'Rob Harrell');' target='_blank">Rob Harrell</a>, on Twitter, asking if there was a way to edit the form. Thankfully, there is. He sent me <a href='https://www.gravityhelp.com/documentation/article/gform_form_tag/' onclick='__gaTracker('send', 'event', 'outbound-article', 'https://www.gravityhelp.com/documentation/article/gform_form_tag/','this page about a filter a developer can use the edit the form tag output by Gravity Forms');' target='_blank">this page about a filter a developer can use the edit the form tag output by Gravity Forms</a>.</p> <p>I added this code to the functions.php file to change the enctype of the form:</p> <pre> add_filter( 'gform_form_tag', 'form_tag', 10, 2 ); function form_tag( $form_tag, $form ) { if ( $form['id'] != 1 ) { //not the form whose tag you want to change, return the unchanged tag return $form_tag; } $form_tag = preg_replace( "|enctype='(.*)'|", "enctype='application/x-www-form-urlencoded'", $form_tag ); return $form_tag; } </pre> <p>Breaking down that code:</p> <pre> //Enter the ID of the form you want to alter here: if ( $form['id'] != 1 ) { //Alter the <form> tag however you like here: $form_tag = preg_replace( "|enctype='(.*)'|", "enctype='application/x-www-form-urlencoded'", $form_tag ); //This can be just about any replacement you want, or you can just return a new string that is the new form tag return '<form class="gravity-form-or-some-other-class" action="post.php" enctype="application/x-www-form-urlencoded">' </pre> <p>Pardot is picky about the form they integrate with, and thankfully the Gravity Forms developers made it possible to alter the &lt;form&gt; tag however we need to.</p> "  
    },  
    - excerpt: {  
        rendered: "<p>I had an interesting request from a client today. They use Pardot to integrate with the forms on their old website, and wanted to do the same with their new WordPress theme. We're using the always spectacular Gravity Forms for &hellip; <a href='http://joshbroton.com/so-you-want-to-change-the-gravity-forms-form-tag/'>Continued</a></p> "  
    },
```

03

REACT + WORDPRESS

```
<article id="post-<?php the_ID(); ?>" <?php post_class('post'); ?>
    role="article" itemscope itemtype="http://schema.org/BlogPosting">
    <header class="content_header post_header">
        <h1 class="content_title post_title"
            title="<?php the_title_attribute(); ?>" itemprop="headline">
            <?php the_title(); ?>
        </h1>
        <aside class="byline vcard">
            <time class="byline_date updated" datetime="<?php echo
the_time('Y-m-j'); ?>" pubdate>
                <?php the_time(get_option('date_format')); ?>
            </time>
            by
            <span class="byline_author"><?php the_author_posts_link(); ?></
span>
        </aside>
    </header>
```

03

REACT + WORDPRESS

```
class Post extends Component {  
  componentWillMount() {  
    document.body.classList.add('post-list', 'blog');  
  }  
  componentWillUnmount() {  
    document.body.classList.remove('post-list', 'blog');  
  }  
  render() {  
    return (  
      //Your JSX goes here  
    )  
  }  
}
```

03

REACT + WORDPRESS

```
<article id={`post-${post.id}`} className="post"
  role="article" itemscope itemtype="http://schema.org/BlogPosting">
  <header class="content_header post_header">
    <h1 class="content_title post_title" itemprop="headline">
      {post.title}
    </h1>
    <aside class="byline vcard">
      <time class="byline_date updated">
        {moment(post.date).format()}
      </time>
      by <span class="byline_author">{post.author.name}</span>
    </aside>
  </header>
  <section class="post_content" itemprop="articleBody">
    <PostThumbnail post={post} />
    <div class="content--inner post_content--inner">(); ?>
      {post.excerpt ? post.excerpt : post.content}
    </div>
  </section>
</article>
```

03

REACT + WORDPRESS

```
<article id={`post-${post.id}`} className="post"
  role="article" itemscope itemtype="http://schema.org/BlogPosting">
  <PostHeaderView title={post.title}
    date={post.date}
    author={post.author} />
  <PostContentView thumbnail={post.thumbnail}
    content={post.content} />
  <PostFooterView categories={post.categories}
    tags={post.tags} />
</article>
```

03

REACT + WORDPRESS

```
<Post post={post} />
```

```
<ol className="posts-archive">
  {posts.map((post) => { return <Post post={post} /> })}
</ol>
```

03

HEADLESS WORDPRESS

1. index.php file

2. change

```
define('WP_USE_THEMES', true);
```

to

```
define('WP_USE_THEMES', false);
```

3. You're done!

03

HEADLESS WORDPRESS

1. front-page.php file
2. ©2016 api.joshbrotон.com
3. You're done!

MORE RESOURCES

- <https://facebook.github.io/react/docs/hello-world.html>
- <https://frontendmasters.com/courses/>
- <http://tri.be/blog/redux-react-and-the-wordpress-rest-api-v2/>
- <https://camjackson.net/post/9-things-every-reactjs-beginner-should-know>
- <https://github.com/reactjs/react-a11y>
- <https://github.com/joshbroton/react-wordpress-slides>

JOSH BROTON

@joshbroton || joshbroton.com

