

History and Philosophy

- Extracted from Basecamp codebase
- One of the preset JS/webpacker configurations for Rails 5.2+ (along with Vue, React, Elm, Ember, AngularJS)
- Uses data-attributes
- Convention over configuration

Install

```
yarn add stimulus
```

The Basics

- Controllers
- Actions
- Targets

Controllers

```
<div data-controller="demo"></div>
```

- the controller is scoped to the element its declared on and children
- autoloaded when using `@stimulus/webpack-helpers`
- convention over configuration: code expected in `controllers/demo_controller.js`
- controllers use "kebab-case"!!



Controllers - Data Map

If you name a data attribute data-CONTROLLER-field-name, you acquire magically generated getter/setters.

```
<div data-controller="demo" data-demo-some-value="foo"></div>
```

produces

```
this.data.get("someValue") # => "foo"  
this.data.has("someValue") # => true  
this.data.set("someValue", bar) # => data-demo-some-value === "bar"s  
this.data.delete("someValue") # => value is destroyed
```

Strongly Typed

- All values are strings...

Controllers - Lifecycle

- `initialize()`
- `connect()`
- `disconnect()`

These work when and how you'd expect

Actions

```
<div data-controller="demo">  
  <button data-action="click->demo#doThing">...</button>  
</div>
```

- handles a DOM event
- *action descriptor* links event (click) to the controller (demo) with the function to run (doThing)
- special global event notation: `foo@window` `bar@document`

Targets - Syntax

```
<div data-controller="demo">  
  <button data-action="click->demo#getMessage">...</button>  
  <div data-target="demo.message"></div>  
</div>
```

```
// controllers/demo_controller.js  
export default class extends Controller {  
  static targets = [ "message" ]  
  ...  
}
```

Targets - Properties

- `this.messageTarget`
- `this.messageTargets`
- `this.hasMessageTarget`

Controller generates a singular, plural, and existential property for each target, `message` in this example

Webscale

- multiple controllers per element are supported (left-to-right execution)
- multiple instances of controllers on different elements are supported

Demo - Carousel in Stimulus

index.html

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <link rel="stylesheet" href="main.css">
  <script src="bundle.js" async></script>
</head>
<body>
  <div data-controller="carousel" data-carousel-index="1">
    <figure>
      <img data-target="carousel.image" />
      <figcaption data-target="carousel.caption">
        </figcaption>
      </figure>
      <button data-action="carousel#previous">Prev</button>
      <button data-action="carousel#next">Next</button>
    </div>
  </body>
</html>
```

index.js

```
import { Application } from "stimulus"
import { definitionsFromContext } from "stimulus/webpack-helpers"

const application = Application.start()
const context = require.context("./controllers", true, /\.js$/)
application.load(definitionsFromContext(context))
```

carousel_controller.js

```
import { Controller } from "stimulus"
import slides from '../slides'

export default class extends Controller {
  static targets = [ "slide", "image", "caption" ]

  initialize() {
    if (!this.data.has("index")) {
      this.index = 0
    }
    this.showCurrentSlide()
  }

  next() {
    const upperBound = slides.length
    this.index = upperBound ? (this.index + 1) % upperBound : this.index + 1
  }

  previous() {
    const upperBound = slides.length
    this.index = upperBound ? (this.index + upperBound - 1) % upperBound : this.index - 1;
  }

  showCurrentSlide() {
    const img = this.imageTarget
    const caption = this.captionTarget
    const slide = slides[this.index]

    img.src = slide.imageUrl
    img.height = 500

    caption.innerHTML = this.generateCaption(slide)
  }

  get index() {
    return parseInt(this.data.get("index"))
  }

  set index(value) {
    this.data.set("index", value)
    this.showCurrentSlide()
  }

  generateCaption(slide) {
    const referralParams = 'utm_source=test-driven-carousel&utm_medium=referral'
    return `${slide.description}</strong> Photo by <a href="https://unsplash.com/@${slide.attribution.username}?${referralParams}">${slide.attribution.name}</a> on <a href="https://unsplash.com/?${referralParams}">Unsplash</a>`
  }
}
```


Carousel in React

THERE IS NO HTML PAGE

```
import React from 'react';
import ReactDOM from 'react-dom';
import Carousel from '../src/Carousel';
import slides from './slides';

const container = document.createElement('div');
document.body.appendChild(container);
ReactDOM.render(<Carousel slides={slides} />, container);
```

Carousel

```
import React from 'react';
import PropTypes from 'prop-types';
import CarouselButton from './CarouselButton';
import CarouselSlide from './CarouselSlide';

class Carousel extends React.PureComponent {
  static propTypes = {
    defaultImg: CarouselSlide.propTypes.Img,
    defaultImgHeight: CarouselSlide.propTypes.imgHeight,
    slides: PropTypes.arrayOf(PropTypes.shape(CarouselSlide.propTypes))
      .isRequired,
  };
  static defaultProps = {
    defaultImg: CarouselSlide.defaultProps.Img,
    defaultImgHeight: CarouselSlide.defaultProps.imgHeight,
  };

  state = {
    slideIndex: 0,
  };
  handleNextClick = () => {
    const { slides } = this.props;
    this.setState(({ slideIndex }) => ({
      slideIndex: (slideIndex + 1) % slides.length,
    }));
  };
  handlePrevClick = () => {
    const { slides } = this.props;
    this.setState(({ slideIndex }) => ({
      slideIndex: (slideIndex + slides.length - 1) % slides.length,
    }));
  };

  render() {
    const { defaultImg, defaultImgHeight, slides, ...rest } = this.props;
    return (
      <div {...rest}>
        <CarouselSlide
          Img={defaultImg}
          imgHeight={defaultImgHeight}
          {...slides[this.state.slideIndex]}
        />
        <CarouselButton data-action="prev" onClick={this.handlePrevClick}>
          Prev
        </CarouselButton>
        <CarouselButton data-action="next" onClick={this.handleNextClick}>
          Next
        </CarouselButton>
      </div>
    );
  }
}
export default Carousel;
```

CarouselButton

```
import React from 'react';
import PropTypes from 'prop-types';

const CarouselButton = props => <button {...props} />;

CarouselButton.propTypes = {
  children: PropTypes.node.isRequired,
};
export default CarouselButton;
```

CarouselSlide

```
import React from 'react';
import PropTypes from 'prop-types';
import styled from 'styled-components';

const DefaultImg = styled.img`
  object-fit: cover;
  width: 100%;
  height: ${props =>
    typeof props.imgHeight === 'number'
      ? `${props.imgHeight}px`
      : props.imgHeight};
`;

const CarouselSlide = ({
  Img,
  imgUrl,
  imgHeight,
  description,
  attribution,
  ...rest
}) => (
  <figure {...rest}>
    <Img src={imgUrl} imgHeight={imgHeight} />
    <figcaption>
      <strong>{description}</strong> {attribution}
    </figcaption>
  </figure>
);

CarouselSlide.propTypes = {
  Img: PropTypes.elementType,
  imgHeight: PropTypes.oneOfType([PropTypes.number, PropTypes.string]),
  imgUrl: PropTypes.string.isRequired,
  description: PropTypes.node.isRequired,
  attribution: PropTypes.node,
};

CarouselSlide.defaultProps = {
  Img: DefaultImg,
  imgHeight: 500,
};

export default CarouselSlide;
```