



# Introduction to Stimulus JS

Hoang

Growth Session #26 - July 25-26 2019



A modest JavaScript framework for the HTML you  
already have

- Stimulus is a JavaScript framework with modest ambitions.
- It doesn't seek to take over your entire front-end—in fact, it's not concerned with rendering HTML at all.
- Instead, it's designed to augment your HTML with just enough behavior to make it shine.
- Stimulus pairs beautifully with *Turbolinks* to provide a complete solution for fast, compelling applications with a minimal amount of effort.

# Example

**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}!`
  }
}
```

Hoang

Greet


"Hello, Hoang!"

## The DOM: A Place for State?

- The core concepts of many modern frameworks lie in separating “state” from the “DOM”. In this way, the DOM acts only as a client to the data it is representing.
- Stimulus flips this concept on its head, and instead relies on the DOM to hold state
- With Stimulus, you won’t see HTML templates. Instead, you will see data-\* attributes that connect the HTML to the JavaScript application.
- So Stimulus doesn’t render templates. Rather it is built to connect actions and events you take on the front end of an application to controllers on the back end.
- Stimulus uses three concepts to accomplish this: controllers, actions, and targets.

# Controllers

- A controller is the basic organizational unit of a Stimulus application.
- Controllers are instances of JavaScript classes that you define in your application. Each controller class inherits from the Controller base class exported by the stimulus module



```
import { Controller } from "stimulus"

export default class extends Controller {
  // ...
}
```

# Actions

- Actions are how you handle DOM events in your controllers.
- An action is a connection between:
  - a controller method
  - the controller's element
  - a DOM event listener

```
<div data-controller="gallery">
  <button data-action="click->gallery#next">...</button>
</div>
```

```
// controllers/gallery_controller.js
import { Controller } from "stimulus"

export default class extends Controller {
  next(event) {
    // ...
  }
}
```

# Actions

## Event Shorthand

Stimulus lets you shorten the action descriptors for some common element/event pairs, such as the button/click pair above, by omitting the event name:



```
<button data-action="gallery#next">...</button>
```




```
<div data-controller="gallery"  
      data-action="resize@window->gallery#layout">  
  ...  
</div>
```



# Targets

- Targets let you reference important elements by name.
- The data-target value `search.query` is called a target descriptor. This descriptor says:
  - search is the scope's controller identifier
  - query is the target name, which can be anything you choose



```
<div data-controller="search">  
  <input type="text" data-target="search.query">  
  <div data-target="search.errorMessage"></div>  
  <div data-target="search.results"></div>  
</div>
```

# Targets

```
// controllers/search_controller.js
import { Controller } from "stimulus"

export default class extends Controller {
  static targets = [ "query", "errorMessage", "results" ]
  // ...
}
```

```
if (this.hasResultsTarget) {
  this.resultsTarget.innerHTML = "..."
}
```

For each target name defined in the static targets array, Stimulus adds the following properties to your controller, where [name] corresponds to the target's name:

Type	Name	Value
Singular	this.[name]Target	The first matching target in scope
Plural	this.[name]Targets	An array of all matching targets in scope
Existential	this.has[Name]Target	A boolean indicating whether there is a matching target in scope


# Lifecycle Callbacks

Special methods called lifecycle callbacks allow you to respond whenever a controller connects to and disconnects from the document:

Method	Invoked by Stimulus...
<code>initialize()</code>	Once, when the controller is first instantiated
<code>connect()</code>	Anytime the controller is connected to the DOM
<code>disconnect()</code>	Anytime the controller is disconnected from the DOM

# Data Maps

Each Stimulus controller has a data map which lets you access special data attributes on the controller's element.

```
  
<div data-controller="content-loader"  
      data-content-loader-url="/messages">  
</div>
```


```
  
// controllers/content_loader_controller.js  
import { Controller } from "stimulus"  
  
export default class extends Controller {  
  connect() {  
    fetch(this.data.get("url")).then(/* ... */)   
  }  
}
```

# Data Maps

Use the following methods in a controller to access data attributes by key:

Method	Result
<code>this.data.get(key)</code>	Returns the string value of the mapped data attribute
<code>this.data.has(key)</code>	Returns true if the mapped data attribute exists
<code>this.data.set(key, value)</code>	Sets the string value of the mapped data attribute
<code>this.data.delete(key)</code>	Deletes the mapped data attribute

- Connecting HTML to JavaScript
- Separation of Concerns
- A Readable Document



Let's try a modest JavaScript  
framework for the HTML **we**  
already have

# Thanks!

## Contact Nimble

[nimblehq.co](https://nimblehq.co)

[hello@nimblehq.co](mailto:hello@nimblehq.co)

## Bangkok

399 Interchange 21 Sukhumvit Road, Unit  
#2402-03, Klong Toei, Wattana, Bangkok  
10110, Thailand

## Singapore

28C Stanley St, Singapore 068737

## Hong Kong

20th Floor, Central Tower  
28 Queen's Road, Central, Hong Kong

