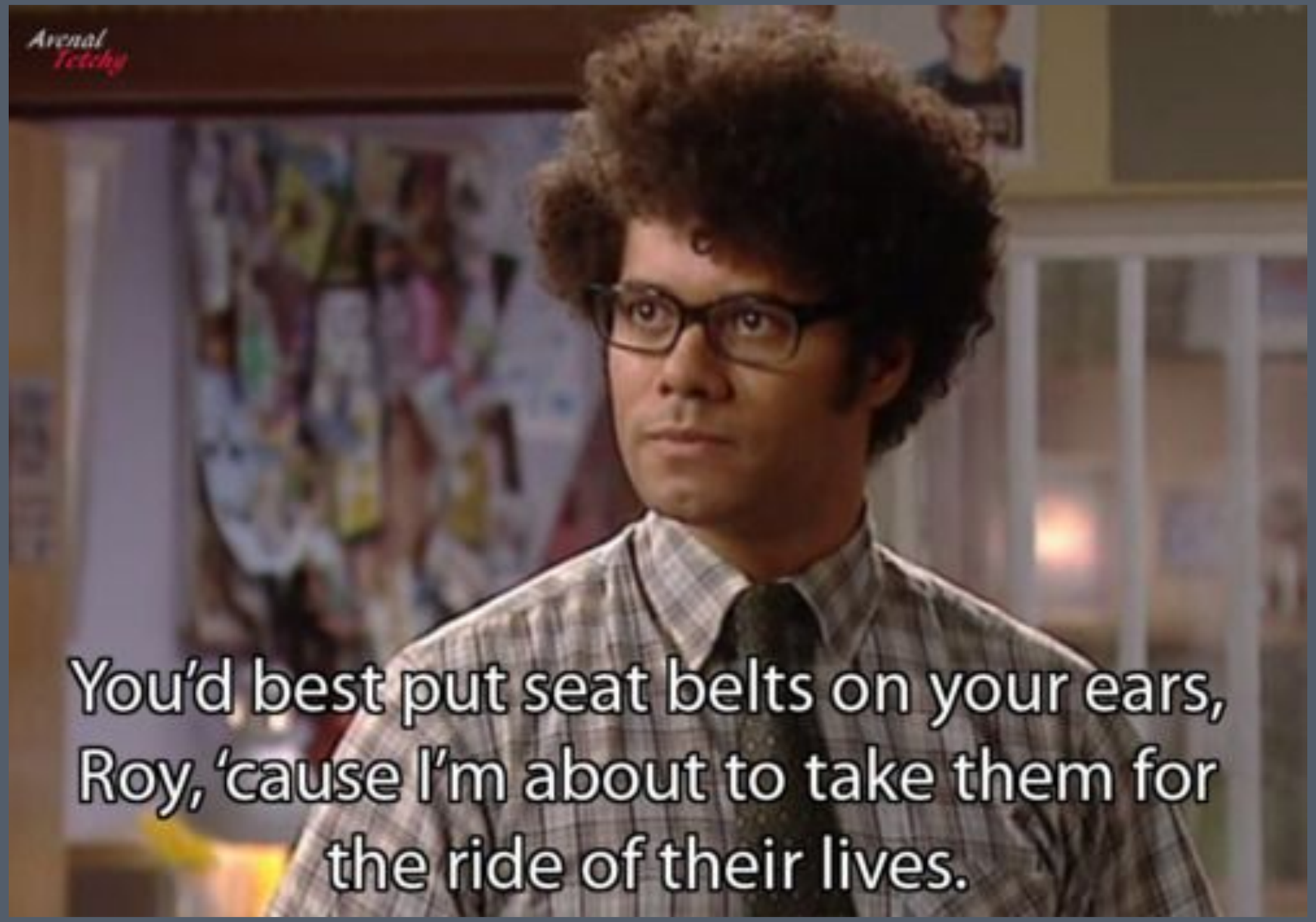
Stimulus Reflex

StimulusJS + Websockets (and friends)


You'd best put seat belts on your ears,
Roy, 'cause I'm about to take them for
the ride of their lives.

Use StimulusReflex to stream page updates using websockets to the browser without writing an JS[1]

[1] Well almost

Break it down

Break it down - StimulusJS

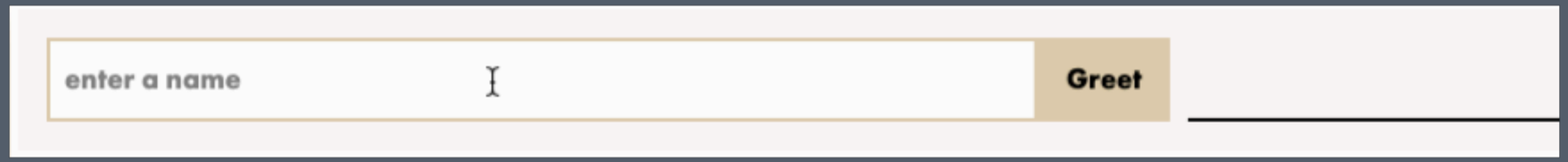
```
<!--HTML from anywhere-->
<div data-controller="hello">
  <input data-target="hello.name" type="text">

  <button data-action="click->hello#greet">
    Greet
  </button>

  <span data-target="hello.output">
  </span>
</div>
```

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

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

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

```
data-action="click->hello#greet"
    data-target="hello.name"
  data-target="hello.output"
```

```html
<!--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>
```

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

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

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



Break it down - Reflex

Websockets

Keeps the browser and the server connected allowing the server and browser to communicate over a full-duplex tcp connection

Break it down - Reflex

ActionCable

The plumbing that allows rails to communicate with websockets

Break it down - Reflex

CableReady

The gem used to trigger client-side DOM changes from server-side Ruby.

Put them all together

Yeah, but why?

Core Concepts

(Comes in 2 flavours )

Core Concepts

Look ma, no controllers

1. HTML data attributes
2. Server side Reflex Object

HTML data attributes

```
<!-- app/views/pages/index.html.erb -->
<a href="#"
  data-reflex="click->Counter#increment"
  data-step="1"
  data-count="<%= @count.to_i %>">
  Increment <%= @count.to_i %>
</a>
```

The `data-syntax` attribute format `[DOM-event]->[ReflexClass]#[action]`

Server side Reflex Object

[DOM-event]->[ReflexClass]#[action]

`click->Counter#increment`

```ruby
# app/reflexes/counter_reflex.rb
class CounterReflex < StimulusReflex::Reflex
  def increment
    @count = element.dataset[:count].to_i + element.dataset[:step].to_i
  end
end
```

Server side Reflex Object

```
data-step="1"
```

```
data-count="<%= @count.to_i %>
```

```ruby
# app/reflexes/counter_reflex.rb
class CounterReflex < StimulusReflex::Reflex
  def increment
    @count = element.dataset[:count].to_i + element.dataset[:step].to_i
  end
end
```

Core Concepts

Less Magic 🧙‍♂️

1. HTML data attributes
2. Client side StimulusReflex controller
3. Server side Reflex Object
4. Server side controller

HTML data attributes

```
<!-- app/views/pages/index.html.erb -->
<a href="#" data-controller="counter" data-action="click->counter#increment">
  Increment <%= @count %>
</a>
```

Client side StimulusReflex controller

```
import { Controller } from 'stimulus';
import StimulusReflex from 'stimulus_reflex';

export default class extends Controller {
  connect() {
    StimulusReflex.register(this)
  }

  increment(event) {
    event.preventDefault()
    this.stimulate('Counter#increment', 1)
  }
```

[ServerSideClass]#[action]

Server side Reflex Object

```ruby
# app/reflexes/counter_reflex.rb
class CounterReflex < StimulusReflex::Reflex
  def increment(step = 1)
    session[:count] = session[:count].to_i + step
  end
end
```

Server side controller

```ruby
# app/controllers/pages_controller.rb
class PagesController < ApplicationController
  def index
    @count = session[:count].to_i
  end
end
```

Diving in - Live coding time 🤞

Getting Setup

```
rails new todo-example --webpack=stimulus
cd todo-example
bundle add stimulus_reflex
bundle exec rails stimulus_reflex:install
rails db:create
```

Handy Tools

Things I found on my adventure

Polacode extension - Code screenshots
https://newcss.net/ - Classless CSS

That is all folks