Intro to Hotwire

Alexey Avdeev

https://github.com/alexx-avdeev/introduction-to-hotwire

What Hotwire consists of?

TURBO

STIMULUS

STRADA

Useful resources

- Hotwire official website (https://hotwired.dev)
- Pragmatic Studio (https://pragmaticstudio.com/courses/hotwire-rails)
- Hotwire Tutorial (https://www.hotrails.dev)
- The Rails Hot Way (https://railshotway.com/)
- GoRails (https://gorails.com/series/hotwire-rails)
- Broadcasts with Turbo Stream (https://blog.corsego.com/turbo-hotwire-broadcasts)

Turbo

- Turbo Drive (responsible for cache, rendering application visits, etc…)
- Turbo Frames (scope part of page to be updated as a whole)
- Turbo Streams (update different parts of page)
- Turbo Native (will not be covered today)

Turbo Frames

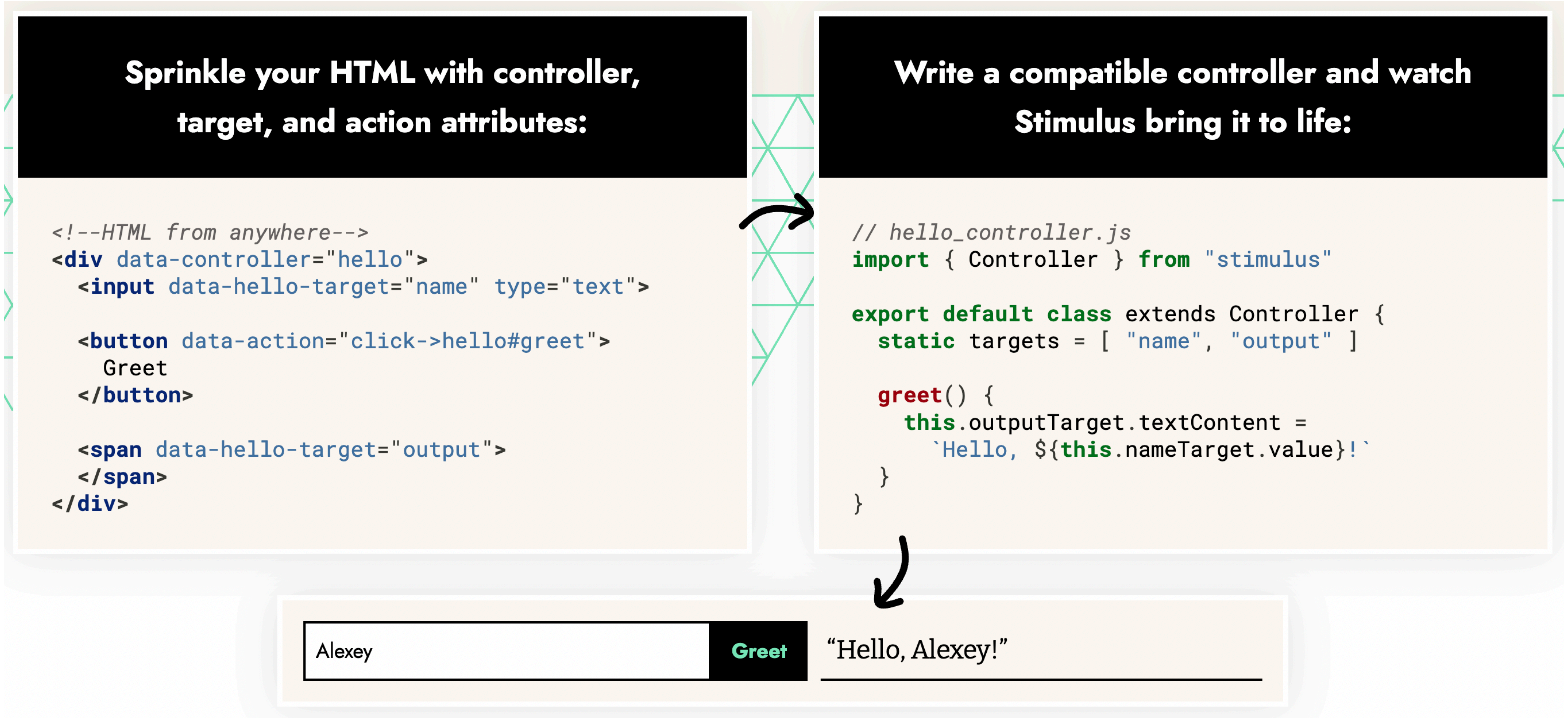
- by default all link clicks inside the frame are scoped to this frame
- when you response is received, turbo will extract target turbo frame and replace existing turbo frame with the new one
- you can target turbo frame from outside the frame of you can target a different frame from current
- turbo frames can be eager- and lazy-loaded
- turbo frames can be reloaded

Turbo Frames Demo

Stimulus

building blocks of Stimulus

- controllers
- actions
- targets



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

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

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

  <span data-hello-target="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}!`
  }
}
```

Alexey

“Hello, Alexey!”

Targets and values

- target elements have to be nested inside the element with controller attached

- to announce target use data-<controller name>-target=“targetName” on the target element

- to specify a value use data-<controller name>-<value-name>-value=“value”

- value callback <name>ValueChanged(value, previousValue)

Actions

- actions is an alternative to setting event listener
- to register an action use data-action=“<eventType>**->**<controller-name>#<method in controller>”
- eventType can be omitted to use default event
- to use global events use <eventType>@window (@document)
- also can use :once, :capture, :passive, :!passive options
  click->gallery#showDialog:once

Controller Lifecycle Callbacks

- initialize - invoked once when controller is initialized
- <targetName>TargetConnected - every time a target with name targetName appears in the document
- connect - every time an element with controller attached appears in the document
- <targetName>TargetDisonnected - every time a target with name targetName disappears from the document
- disconnect - every time the element with controller attached disappears from the document

Stimulus Demo

Turbo Streams

- DO NOT REQUIRE TURBO FRAMES TO WORK!
- turbo streams are designed to deliver fragments of html to target elements on a page
- each turbo stream element is specifies target element by id
- there are 7 possible turbo stream actions

Turbo Stream Actions

- append — appends contents of targeted element
- prepend — prepends contents of targeted element
- replace — replaces targeted element with contents of turbo stream
- update — updates contents of targeted element with contents of turbo stream
- remove — removes targeted element
- before — adds contents of turbo stream before targeted element
- after — adds contents of turbo stream after targeted element

Turbo Streams Demo

https://github.com/alexx-avdeev/introduction-to-hotwire