Ruby

A beginner-friendly yet detailed guide to understanding and implementing Hotwire in Ruby on Rails applications. Covers Turbo Drive, Turbo Frames, Turbo Streams, and Stimulus.

Hotwire Basics

What is Hotwire?

Hotwire stands for **HTML Over The Wire**. It's a collection of frameworks designed to build modern web applications without much JavaScript by sending HTML instead of JSON over the wire.

Components of Hotwire:

- Turbo Drive
- Turbo Frames
- Turbo Streams
- Stimulus

Turbo Drive

What it does:
Turbo Drive accelerates links and form submissions by intercepting them and rendering only the changed parts of the page.

How to use:
No setup required. Works automatically for all links and forms unless opted out.

Opting out:
Add `data-turbo="false"` to links or forms.

Example:
`<a href="/home" data-turbo="false">Home</a>`

Turbo Frames

What it does:
Allows you to update only a part of the page (a frame) without a full page reload.

How to use:
Wrap the content in a `<turbo-frame>` tag with a unique ID.

Example:

```
<turbo-frame id="messages">
  <!-- Content to be updated -->
</turbo-frame>
```

Updating a frame:
The server responds with HTML that includes a `<turbo-frame>` with the same ID.

Advanced Hotwire

Turbo Streams

What it does:
Turbo Streams deliver page changes over WebSocket, SSE, or in response to form submissions using CRUD-like actions.

Actions:

- `append`
- `prepend`
- `replace`
- `update`
- `remove`

Example:

```
<turbo-stream action="append" target="messages">
  <template>
    <div id="message_1">New message</div>
  </template>
</turbo-stream>
```

Stimulus

What it does:
Stimulus is a modest JavaScript framework that augments your HTML with just enough behavior to make it shine.

Core concepts:

- Controllers
- Actions
- Targets

Example Controller:

```
import { Controller } from
"@hotwired/stimulus"

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

  greet() {
    console.log(`Hello,
${this.nameTarget.value}!`)
  }
}
```

HTML Usage:

```
<div data-
controller="greeter">
  <input data-greeter-
target="name"
type="text">
  <button data-
action="click-
>greeter#greet">Greet</bu
tton>
</div>
```

r

Hotwire in Ruby on Rails

Turbo Drive

Turbo Drive accelerates links and form submissions by avoiding full page reloads.

How to use:
Simply include Turbo in your application.js file.

Example:

```
# app/assets/javascripts/application.js
import "@hotwired/turbo-rails"
```

Benefits:

- Faster page loads
- Reduced server load
- Seamless user experience

Limitations:

- Not suitable for all types of navigation
- Requires careful handling of JavaScript events

Turbo Frames

Turbo Frames allow you to update parts of a page without a full reload.

How to use:
Wrap the content you want to update in a `<turbo-frame>` tag.

Example:

```
<turbo-frame id="messages">
  <!-- Content to be updated -->
</turbo-frame>
```

Benefits:

- Partial updates
- Better performance
- Simplified code

Limitations:

- Limited to frame content
- Requires server-side support

Turbo Streams

Turbo Streams deliver page changes over WebSocket, SSE, or in response to form submissions.

How to use:
Use `turbo_stream` helpers in your controllers.

Example:

```
def create
  @message = Message.new(message_params)
  if @message.save
    turbo_stream
  else
    render :new
  end
end
```

Benefits:

- Real-time updates
- Efficient
- Easy to implement

Limitations:

- Requires WebSocket or SSE
- More complex setup

Stimulus

Stimulus is a modest JavaScript framework for adding behavior to your HTML.

How to use:
Create controllers and connect them to your HTML with data attributes.

Example:

```
// app/javascript/controllers/hello_controller.js
import { Controller } from "@hotwired/stimulus"

export default class extends Controller {
  connect() {
    console.log("Hello, Stimulus!")
  }
}
```

Benefits:

- Lightweight
- Easy to learn
- Integrates well with Turbo

Limitations:

- Not a full-fledged framework
- Limited functionality