

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

links and form

submissions by

page.

Opting out:

Add data-

or forms.

rendering only the

turbo="false" to links

What it does: Turbo Drive accelerates

No setup required. Works automatically for all links and forms intercepting them and unless opted out. changed parts of the Example:

Stimulus

How to use:

Home

Turbo Frames

What it does:

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

Example:

id="messages">

Wrap the content in a <turbo-frame> tag with a unique ID.

How to use:

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

Advanced Hotwire

Turbo Streams

What it does:

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

Actions:

- append .
- prepend .
- replace
- update
- remove

Example:

r

<div id="message_1">New message</div>

se to	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:	HTML Usage:
	<pre>import { Controller } from</pre>	<div data-<="" td=""></div>
	"@hotwired/stimulus"	<pre>controller="greeter"></pre>
		<input data-greeter-<="" td=""/>
	export default class extends	target="name"
	Controller {	<pre>type="text"></pre>
	<pre>static targets = ["name"]</pre>	<button data-<="" td=""></button>
		action="click-
	greet() {	>greeter#greet">Greet
	<pre>console.log(`Hello,</pre>	tton>
	<pre>\${this.nameTarget.value}!`)</pre>	
	}	
	}	

Hotwire in Ruby on Rails

Turbo Frames Turbo Drive Turbo Drive accelerates links and form submissions by avoiding full page Turbo Frames allow you to update parts of a page without a full reload. reloads. How to use: How to use: Wrap the content you want to update in a <turbo-frame> tag. Simply include Turbo in your application.js file. Example: Example: Benefits: Benefits: Faster page loads . • Partial updates Reduced server load . Better performance . Seamless user experience Simplified code Limitations: Limitations: • Not suitable for all types of navigation • Limited to frame content Requires careful handling of JavaScript events . Requires server-side support **Turbo Streams** Stimulus Turbo Streams deliver page changes over WebSocket, SSE, or in response to Stimulus is a modest JavaScript framework for adding behavior to your

form submissions.	HTML.
How to use: Use turbo_stream helpers in your controllers.	How to use: Create controllers and connect them to your HTML with data attributes.
Example:	Example:
def create	// app/javascript/controllers/hello_controller.js
<pre>@message = Message.new(message_params)</pre>	<pre>import { Controller } from "@hotwired/stimulus"</pre>
if @message.save	
turbo_stream	<pre>export default class extends Controller {</pre>
else	connect() {
render :new	<pre>console.log("Hello, Stimulus!")</pre>
end	}
end	}
Benefits:	Benefits:
Real-time updates	• Lightweight
• Efficient	• Easy to learn
Easy to implement	Integrates well with Turbo
Limitations:	Limitations:
Requires WebSocket or SSE	Not a full-fledged framework
More complex setup	Limited functionality