

Stimulus Reflex and Cable Ready

What are they?
How do I use them?

Can they help me create a better Reactive app
Than React!?





Beginning and Goals

- StimulusReflex was originally inspired by LiveView, but we are charting our own course.

Our goal has always been to make building modern apps with Rails the most productive and enjoyable option available.

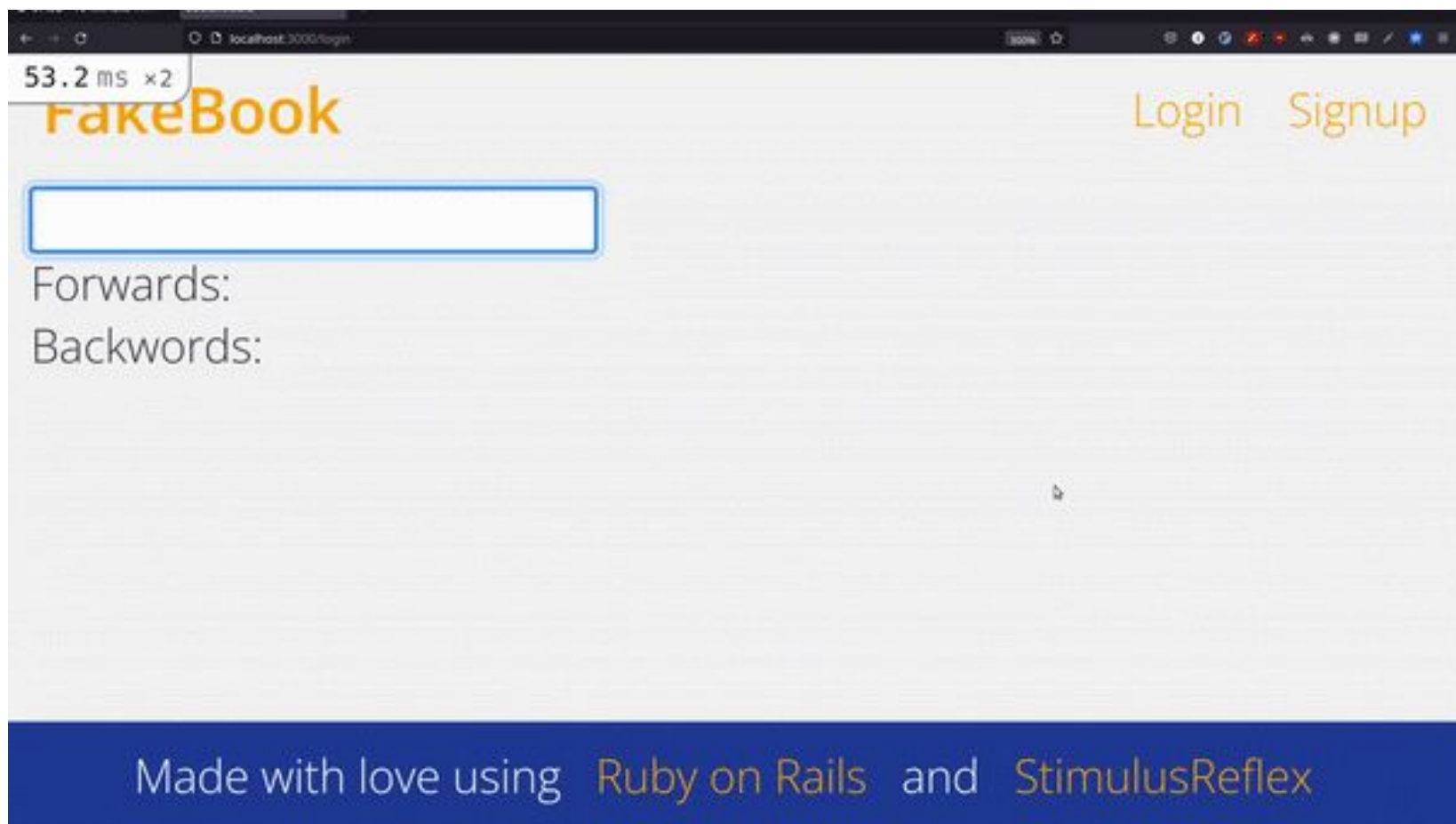


The magic sauce

```
</head>  
<body>  
  < ... data-reflex="click→Greet#say_hi" ... />  
</body>  
</html>
```

```
3  
4 class GreetReflex  
5   def say_hi  
6     ...  
7   end  
8 end
```





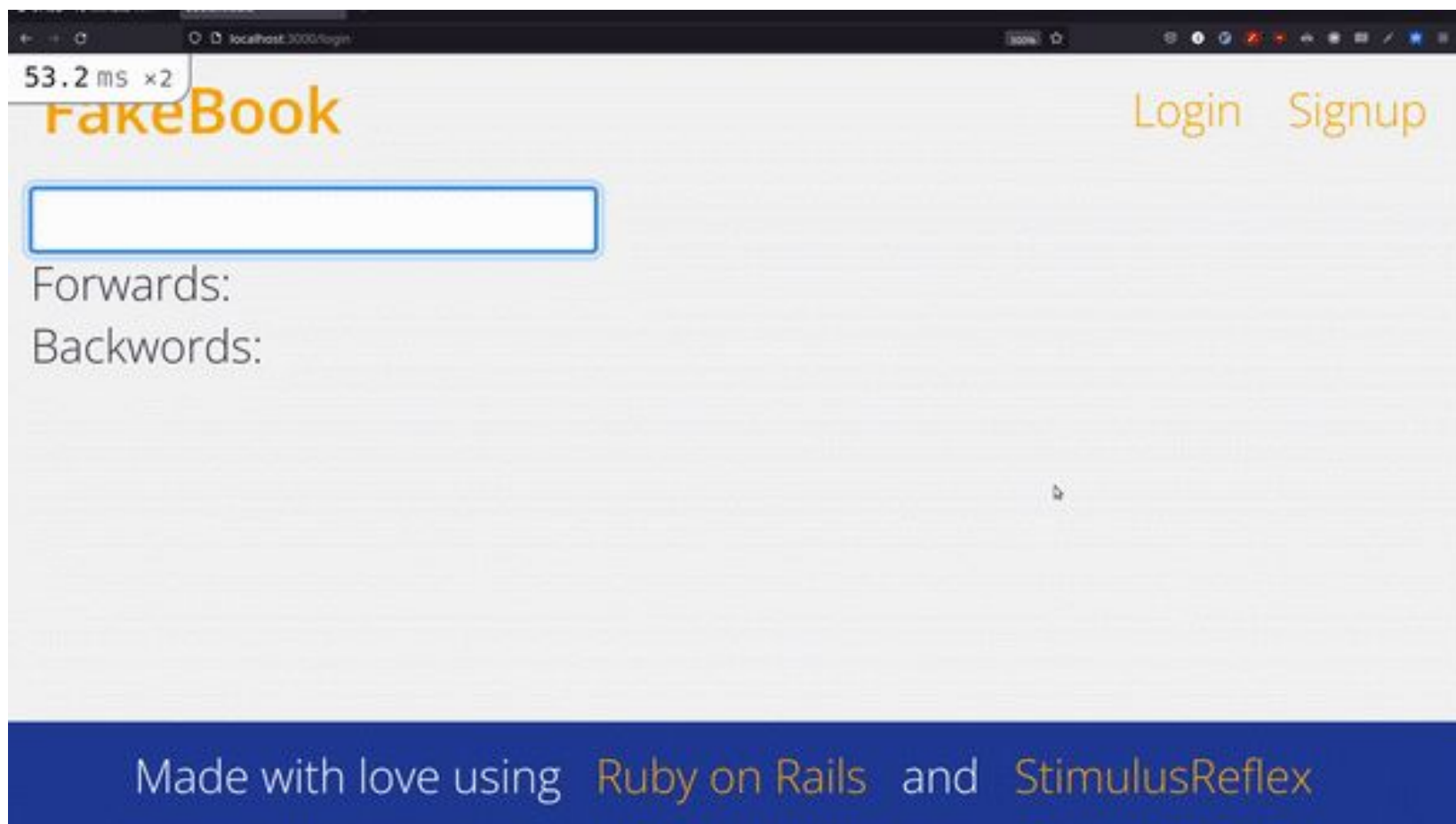
```
2
3 export default class extends Controller {
4   static Targets = ['forwards', 'backwards']
5
6   words(event) {
7     // With each keyup event
8     // Get the event.currentTarget's value
9     // Display the value as is for one div
10    // Display the reversed value for another div
11
12    let value = event.currentTarget.value
13    this.forwardTarget.textContent = value
14    this.backwardTarget.textContent = value.reverse()
15  }
16 }
```



```
1 <div data-reflex-root=".forward,.backward">
2   <input type="text" data-reflex="keyup->Example#words">
3   <div class="forward">Forwards: <%= @words %></div>
4   <div class="backward">Backwards: <%= @words&.reverse %></div>
5 </div>
```

```
2
3 class ExampleReflex < ApplicationReflex
4   def words
5     @words = element[:value]
6   end
7
```





Say hello to Morph

- Morph is short for morphdom
- A Javascript library that makes changes to the existing DOM in a very efficient manner



What is Cable Ready?

- CableReady is a Ruby gem that was first released in May 2017.

It lets you create great real-time user experiences by triggering client-side DOM changes, events and notifications over ActionCable web sockets.



Cavalry

v4.5.0 >

Stream Identifiers

Broadcasting to Resources

Customization

Troubleshooting >

Advocating for "Reactive Rails"

Release History

REFERENCE

Methods

Operations ▾

DOM Mutations

Element Property Mutations

DOM Events

Browser Manipulations

Notifications

Operations

⋮

DOM Mutations

- [append](#)
- [graft](#)
- [inner_html](#)
- [insert_adjacent_html](#)
- [insert_adjacent_text](#)
- [morph](#)
- [outer_html](#)
- [prepend](#)
- [remove](#)
- [replace](#)
- [text_content](#)


Element Property Mutations

- [add_css_class](#)



Powered By **GitBook**

Infantry

 v4.5.0 >

Stream Identifiers

Broadcasting to Resources

Customization

Troubleshooting >

Advocating for "Reactive Rails"

Release History

REFERENCE

Methods

Operations ▾


DOM Mutations

Element Property Mutations

DOM Events

Browser Manipulations

Notifications

 Powered By GitBook

Element Property Mutations

- `add_css_class`
- `remove_attribute`
- `remove_css_class`
- `set_attribute`
- `set_dataset_property`
- `set_property`
- `set_style`
- `set_styles`
- `set_value`

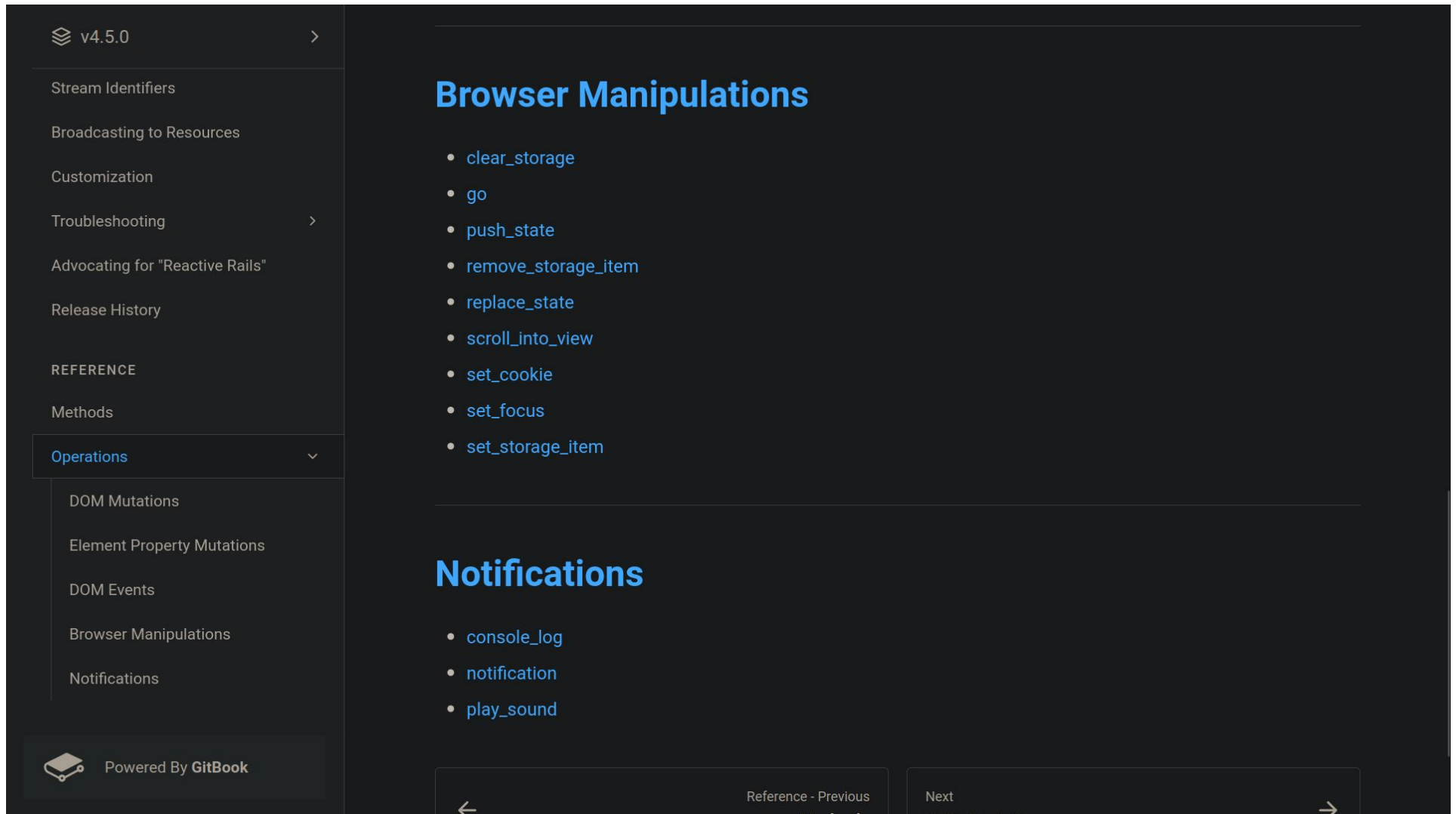
DOM Events

- `dispatch_event`

Browser Manipulations



Administration



The screenshot displays a web application interface with a dark-themed sidebar on the left and a main content area on the right. The sidebar includes a version indicator 'v4.5.0' at the top, followed by a list of navigation items: 'Stream Identifiers', 'Broadcasting to Resources', 'Customization', 'Troubleshooting', 'Advocating for "Reactive Rails"', 'Release History', and a 'REFERENCE' section. Under 'REFERENCE', there is a 'Methods' section which is expanded to show 'Operations'. The 'Operations' list includes 'DOM Mutations', 'Element Property Mutations', 'DOM Events', 'Browser Manipulations', and 'Notifications'. At the bottom of the sidebar, there is a 'Powered By GitBook' logo. The main content area features two sections: 'Browser Manipulations' and 'Notifications', each with a list of methods. The 'Browser Manipulations' list includes 'clear_storage', 'go', 'push_state', 'remove_storage_item', 'replace_state', 'scroll_into_view', 'set_cookie', 'set_focus', and 'set_storage_item'. The 'Notifications' list includes 'console_log', 'notification', and 'play_sound'. At the bottom of the main content area, there are navigation buttons: 'Reference - Previous' and 'Next'.

v4.5.0

Stream Identifiers

Broadcasting to Resources

Customization

Troubleshooting

Advocating for "Reactive Rails"

Release History

REFERENCE

Methods

Operations

DOM Mutations

Element Property Mutations

DOM Events

Browser Manipulations

Notifications

Powered By **GitBook**

Browser Manipulations

- `clear_storage`
- `go`
- `push_state`
- `remove_storage_item`
- `replace_state`
- `scroll_into_view`
- `set_cookie`
- `set_focus`
- `set_storage_item`

Notifications

- `console_log`
- `notification`
- `play_sound`

Reference - Previous

Next

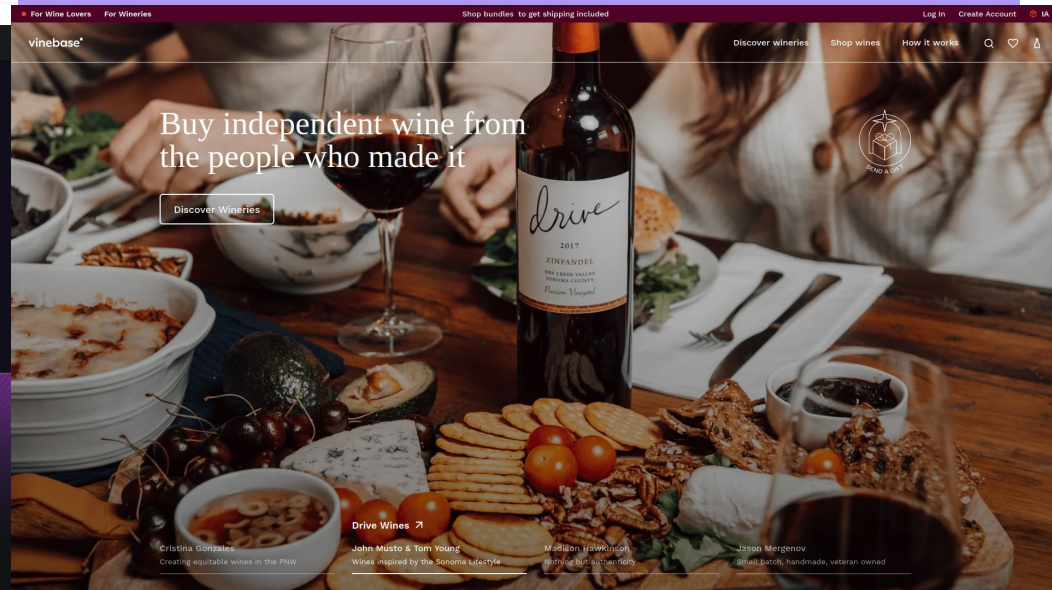
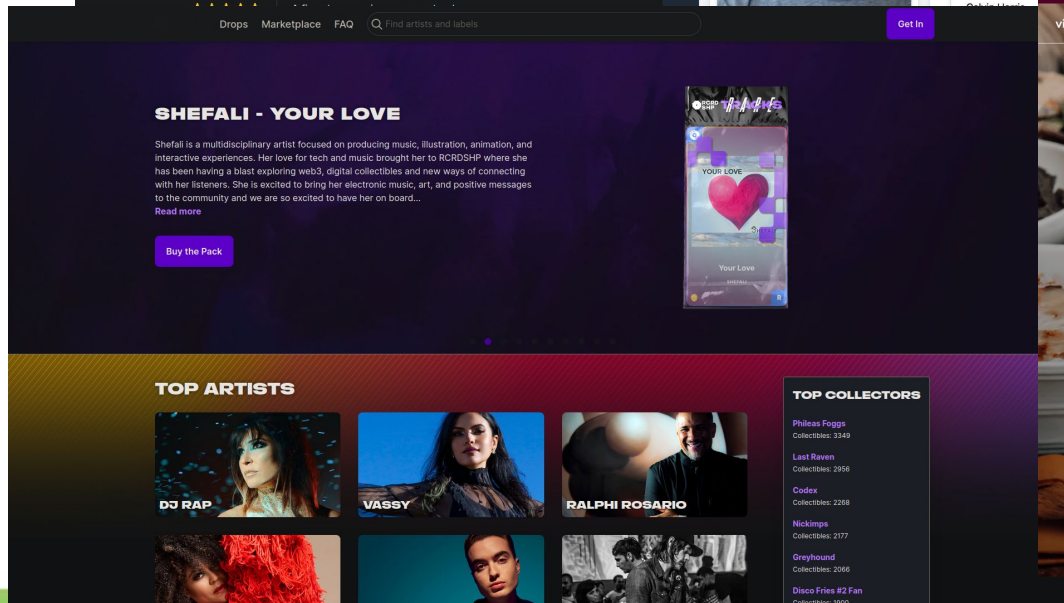
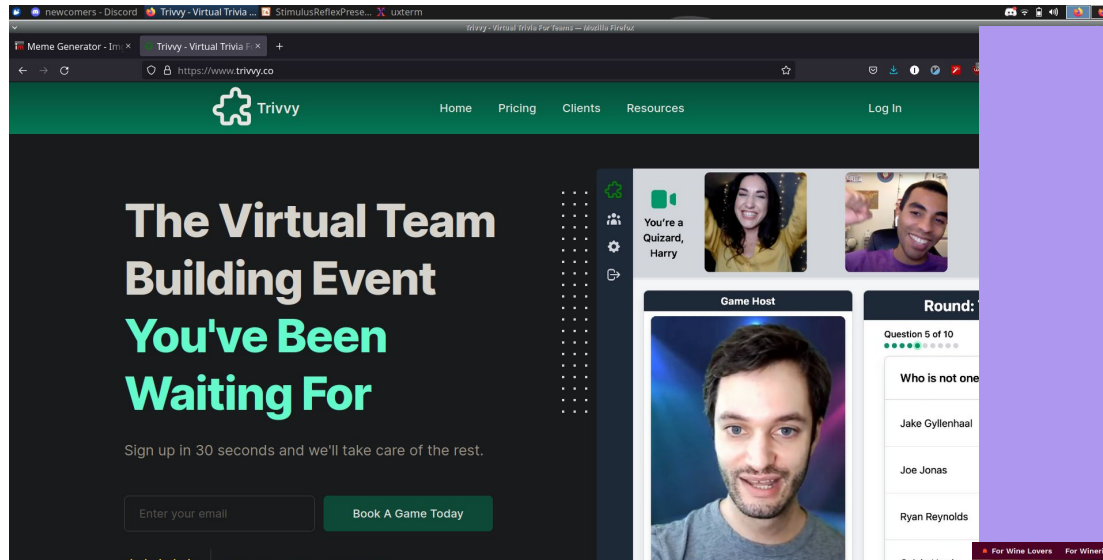
```
1 # frozen_string_literal: true
2
3 class ExampleReflex < ApplicationReflex
4   def words
5     cable_ready.console_log(message: "Hello Ruby!")
6     @words = element[:value]
7   end
8
```



```
456
457   // Notifications
458
459   consoleLog: operation => {
460     before(document, operation)
461     operate(operation, () => {
462       const { message, level } = operation
463       level && ['warn', 'info', 'error'].includes(level)
464         ? console[level](message)
465         : console.log(message)
466     })
467     after(document, operation)
468   },
469
```



Production Reflexes



- Email: inscription4@outlook.com
- Github/Discord: binarygit

