

Introduction to Hotwire Native

Build iOS and Android apps with Ruby on Rails

Mike Dalton

Introduction

- Based in Philadelphia
- Ruby on Rails developer
- Using Hotwire Native on my side project



my social media pic

I work at  Triumph



[MOTORCYCLES](#)

[ACCESSORIES](#)

[CLOTHING SHOP](#)

[OWNERS](#)

[BRAND](#)

[RACING](#)

TIGER 900 RANGE

[OVERVIEW](#)

[MODELS](#)

[SPECIFICATION](#)

[REASONS TO RIDE](#)

[ACCESSORIES](#)

[REVIEWS](#)

[TEST RIDE >](#)

[CONFIGURE](#)





Play games, win real money



triumph.com



triumph.io 👍

Wondering why everything looks different? [Learn more](#)

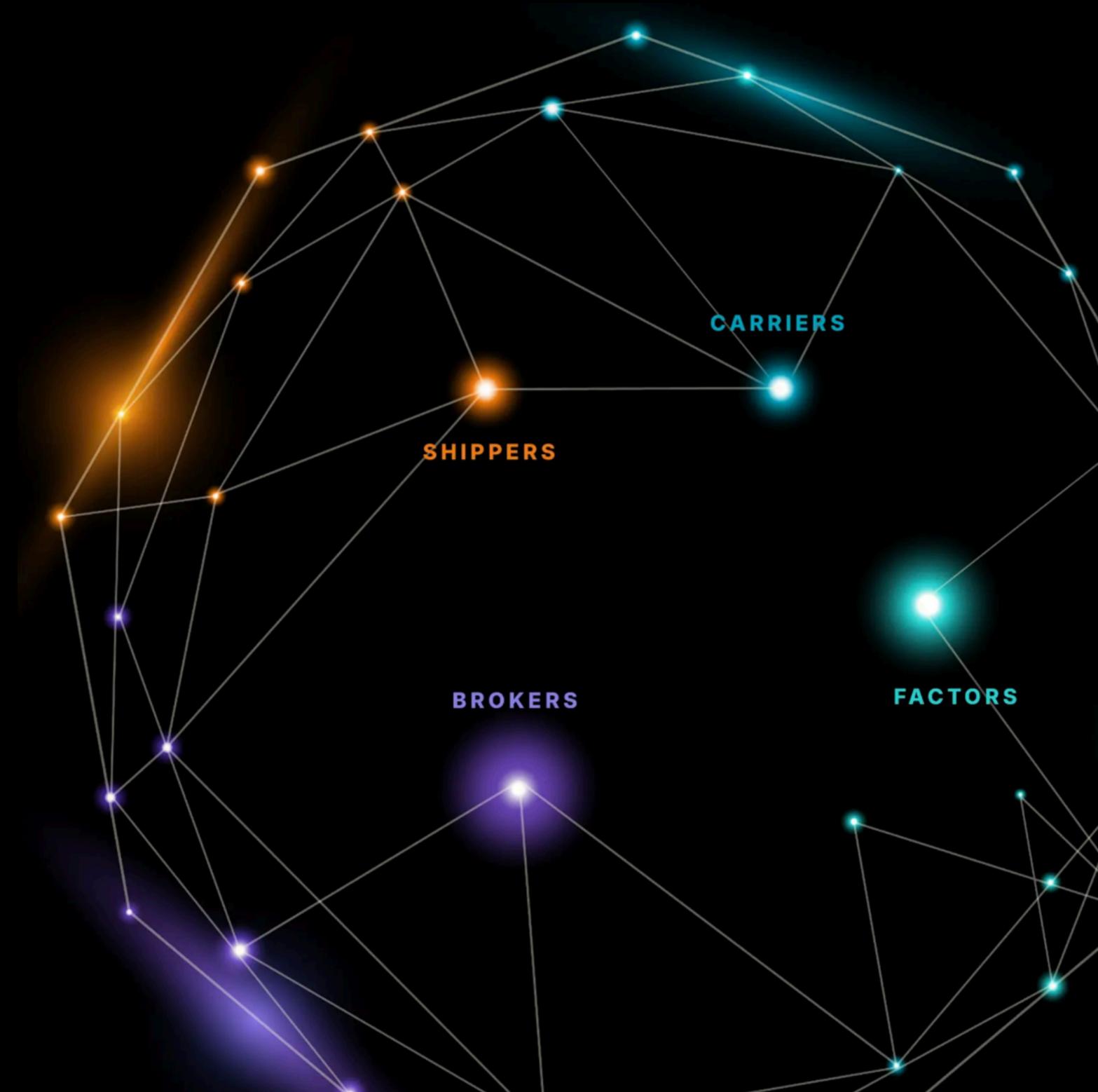


[Broker](#) ▾ [Carrier](#) ▾ [Shipper](#) ▾ [Factor](#) ▾ [Solutions](#) ▾ [About Us](#) ▾

TRANSACT CONFIDENTLY

Get ready to experience the Triumph Network — the network modernizing and simplifying freight transactions.

[Get Started](#) >





We're hiring!

Visited Chicago many times

The screenshot shows the homepage for GOTO Chicago 2019. The top navigation bar includes the 'goto;' logo, links for 'Schedule', 'Speakers', 'Masterclasses', and 'More', and additional links for 'Free GOTO Nights', 'Become a Partner', and 'Our Partners'. The main content area features a large red background with a photo of a speaker. The text reads: 'GOTO Chicago 2019', 'Conference: April 29 & 30', 'Masterclasses: April 28, May 1 & 2', 'Theme: What Works', and a list of highlights including 'Cloud Native', 'Machine Learning', 'Microservices', 'Human side of Tech', 'Programming Languages', and 'DevOps'.

2019



2024



3x times

GRUBHUB™

Remotely from 2017-2021

All projects >

Spring Boot

Spring Framework

> **Spring Data**> **Spring Cloud**

Spring Cloud Data Flow

Spring gRPC

> **Spring Security**Spring Authorization
Server

Spring for GraphQL

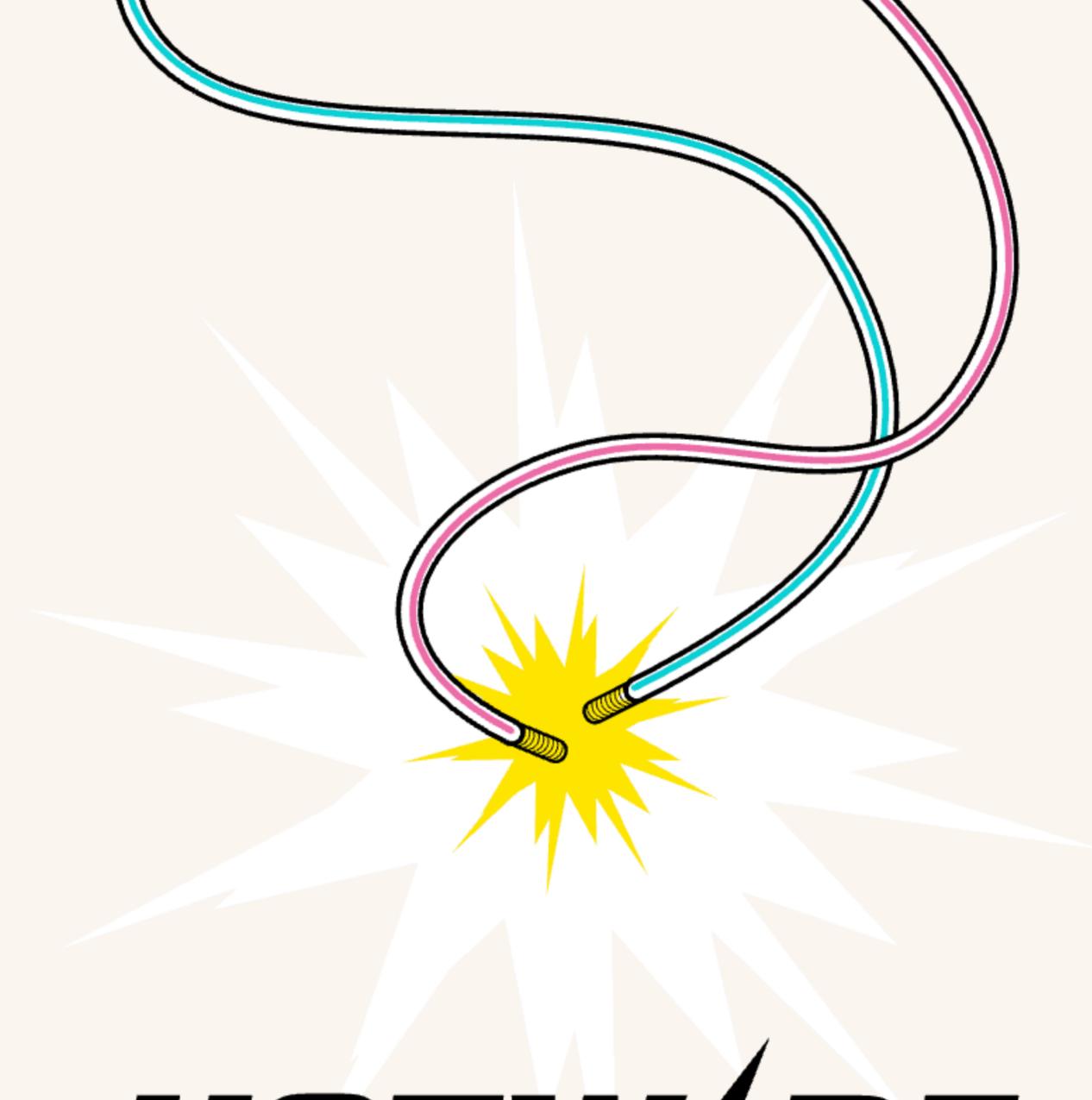
> **Spring Session** **Spring Boot** **4.0.3****OVERVIEW****LEARN****SUPPORT****SAMPLES**

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run".

We take an opinionated view of the Spring platform and third-party libraries so you can get started with minimum fuss. Most Spring Boot applications need minimal Spring configuration.

If you're looking for information about a specific version, or instructions about how to upgrade from an earlier release, check out [the project release notes section](#) on our wiki.

Features



HOTWIRE

HTML OVER THE WIRE

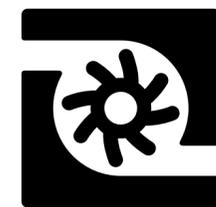
Hotwire

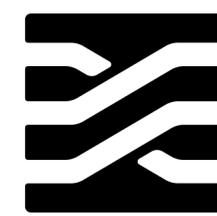
- Created by 37signals
- Default front end “framework” in Rails 7+
- HTML (not JSON) “over the wire”

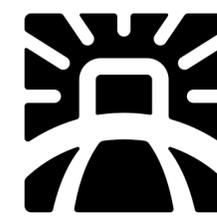


“Hotwire is an alternative approach to building modern web applications without using much JavaScript by sending HTML instead of JSON over the wire.”

<https://hotwired.dev/>

 **TURBO**

 **STIMULUS**

 **NATIVE**

Hotwire

Turbo

- Requires JavaScript but...
- No need to write your own JavaScript!
- Client-side behavior determined by server-side responses

Hotwire

Turbo Drive

- Turbolinks successor
- Works “out of the box”
- Performs requests as XHR to prevent page reload

Hotwire

Turbo Frames

- Only update parts of a page
- Links or form submissions

Reminders

Do the laundry



Make dinner



Take out the trash

Add

Hotwire

Turbo Frames

```
class RemindersController < ApplicationController
  def index
    @reminders = Reminder.all
  end

  def edit
    @reminder = Reminder.find(params.expect(:id))
  end

  def update
    reminder = Reminder.find(params.expect(:id))
    reminder.update(reminder_params)
    redirect_to reminders_path, status: :see_other
  end

  private

  def reminder_params
    params.expect(reminder: [ :title ])
  end
end
```

app/controllers/reminders_controller.rb

```
<% @reminders.each do |reminder| %>
  <%= turbo_frame_tag dom_id(reminder) do %>
    <%= link_to edit_reminder_path(reminder) do %>
      <%= reminder.title %>
    <% end %>
  <% end %>
<% end %>
```

app/views/reminders/index.html.erb

```
<%= turbo_frame_tag dom_id(@reminder) do %>
  <%= render "form", reminder: @reminder %>
  <%= form_with(model: @reminder) do |form| %>
    <%= form.text_field :title %>
    <%= form.submit %>
  <% end %>
<% end %>
```

app/views/reminders/edit.html.erb

Hotwire

Turbo Streams

- Modify any part of the page
- More control, more code, more complexity

Reminders

Do the laundry



Make dinner



Add

Hotwire

Turbo Streams

```
class RemindersController < ApplicationController
  def destroy
    reminder = Reminder.find(params.expect(:id))
    reminder.destroy!
    respond_to do |format|
      format.turbo_stream do
        render turbo_stream: turbo_stream.remove(reminder)
      end
    end
  end
end
```

app/controllers/reminders_controller.rb

```
<% @reminders.each do |reminder| %>
  <%= turbo_frame_tag dom_id(reminder) do %>
    <%= button_to reminder_path(reminder), method: :delete do %>
      <svg ...>
    </svg>
    <% end %>
  <% end %>
<% end %>
```

app/views/reminders/index.html.erb

Hotwire

Stimulus

- Most control
- Custom JavaScript
- jQuery with structure

Reminders

Do the laundry



Make dinner



Add

Hotwire

Stimulus

```
export default class extends Controller {
  static targets = ["input", "submitButton"]

  onInput() {
    if (this.inputTarget.value.length > 0) {
      this.submitButtonTarget.disabled = false
    } else {
      this.submitButtonTarget.disabled = true
    }
  }
}
```

app/javascript/controllers/savable_controller.js

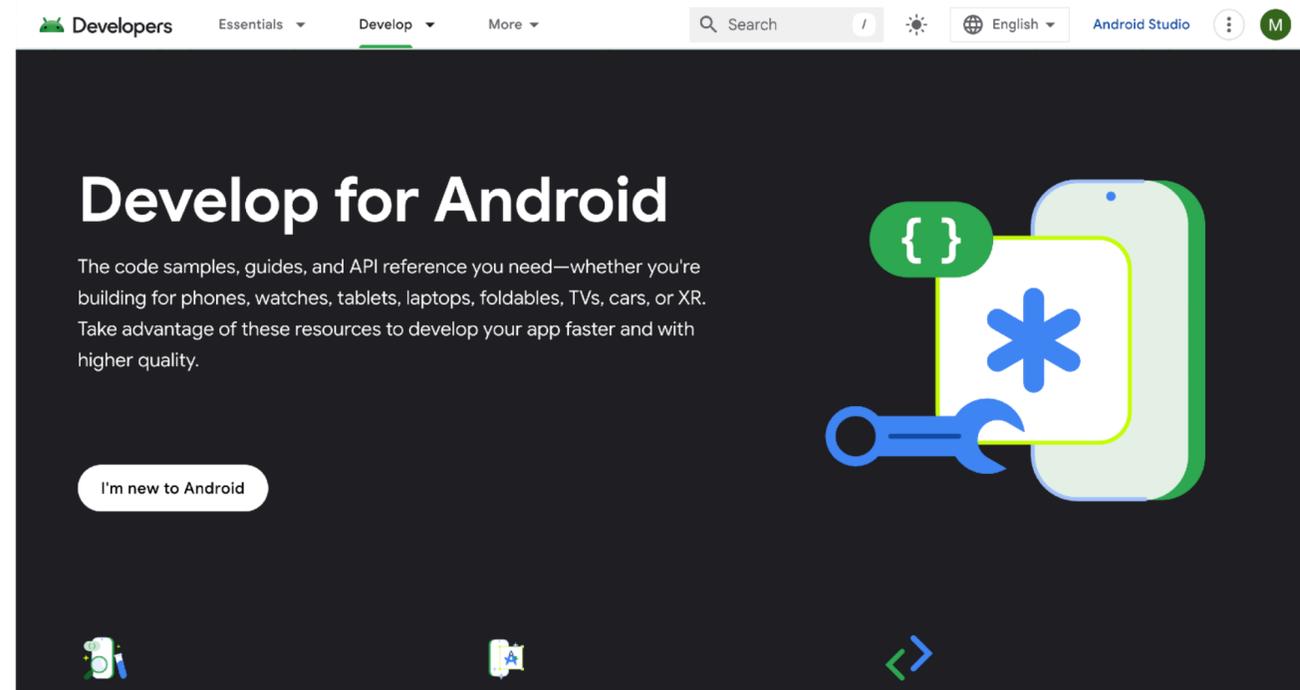
```
<%= form_with(model: reminder,
              data: { controller: "savable" }) do |form| %>
  <%= form.text_field :title,
        data: { savable_target: "input", action: "savable#onInput" } %>
  <%= form.button type: "submit",
        disabled: reminder.new_record?,
        data: { savable_target: "submitButton" } do %>
    <%= reminder.new_record? ? "Add" : "Update" %>
  <% end %>
<% end %>
```

app/views/reminders/_form.html.erb

Mobile in 2026 = Many Choices

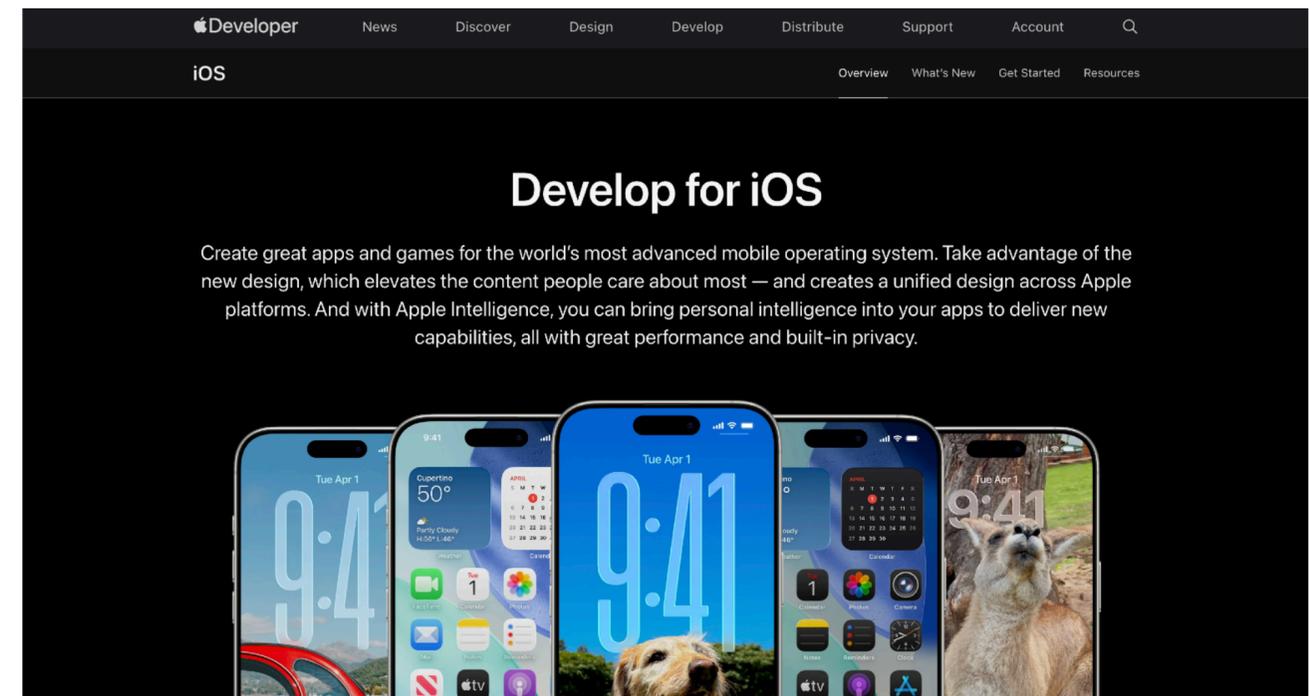
Native Alternatives

Android (Kotlin)



The screenshot shows the top navigation bar of the Android Developers website. It includes the 'Developers' logo, 'Essentials', 'Develop', and 'More' dropdown menus, a search bar, a language selector set to 'English', and 'Android Studio' and 'M' icons. The main content area has a dark background with the heading 'Develop for Android'. Below the heading is a paragraph: 'The code samples, guides, and API reference you need—whether you're building for phones, watches, tablets, laptops, foldables, TVs, cars, or XR. Take advantage of these resources to develop your app faster and with higher quality.' To the right is an illustration of a smartphone with a code editor overlay showing curly braces and a blue asterisk. A blue wrench icon is positioned below the code editor. At the bottom left, there is a white button that says 'I'm new to Android'. The bottom of the page features several small icons representing different Android devices and development tools.

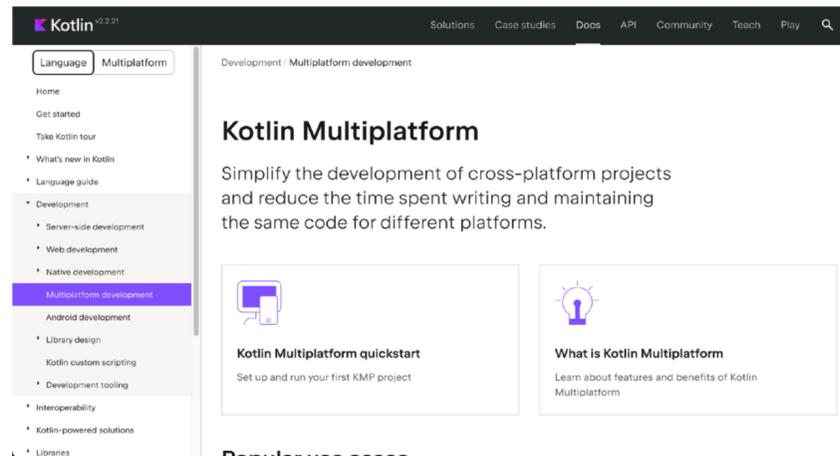
iOS (Swift)



The screenshot shows the top navigation bar of the Apple Developer website. It includes the 'Developer' logo, 'News', 'Discover', 'Design', 'Develop', 'Distribute', 'Support', and 'Account' links, and a search icon. The main content area has a dark background with the heading 'Develop for iOS'. Below the heading is a paragraph: 'Create great apps and games for the world's most advanced mobile operating system. Take advantage of the new design, which elevates the content people care about most — and creates a unified design across Apple platforms. And with Apple Intelligence, you can bring personal intelligence into your apps to deliver new capabilities, all with great performance and built-in privacy.' At the bottom, there is a row of five iPhone screens displaying various iOS home screens with different wallpapers and widgets.

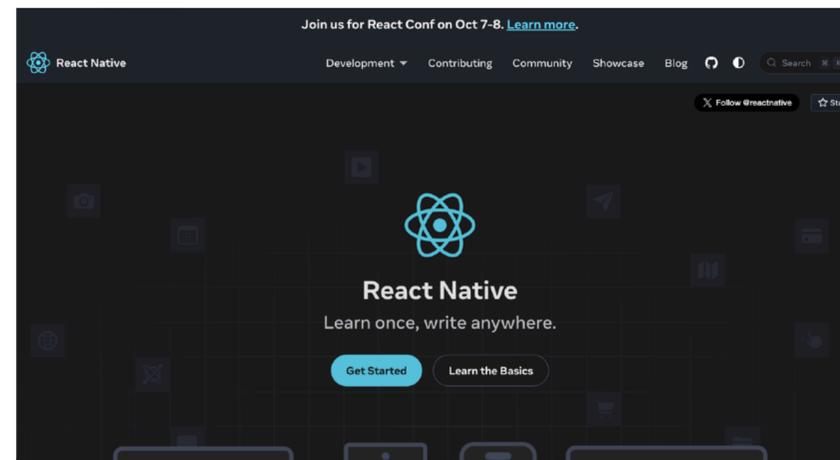
Crossplatform Alternatives

Kotlin Multiplatform



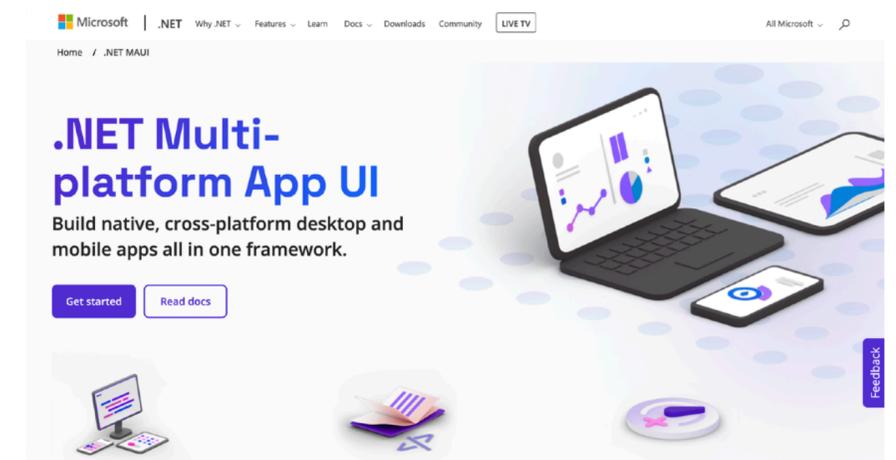
Kotlin

React Native



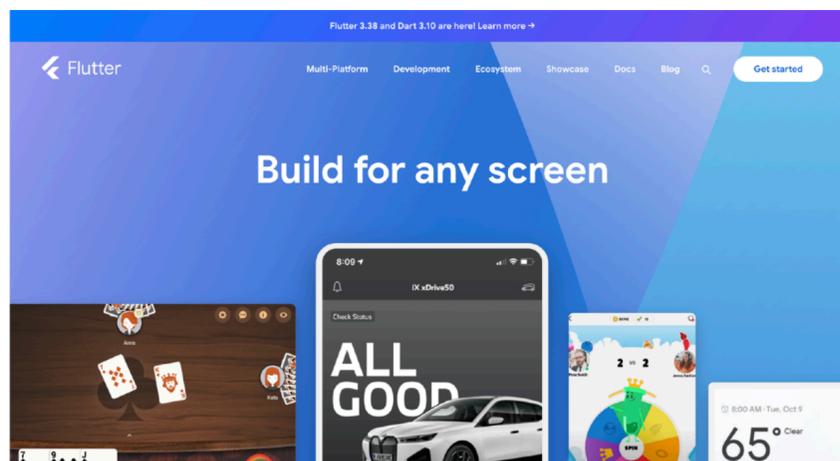
React

.NET MAUI



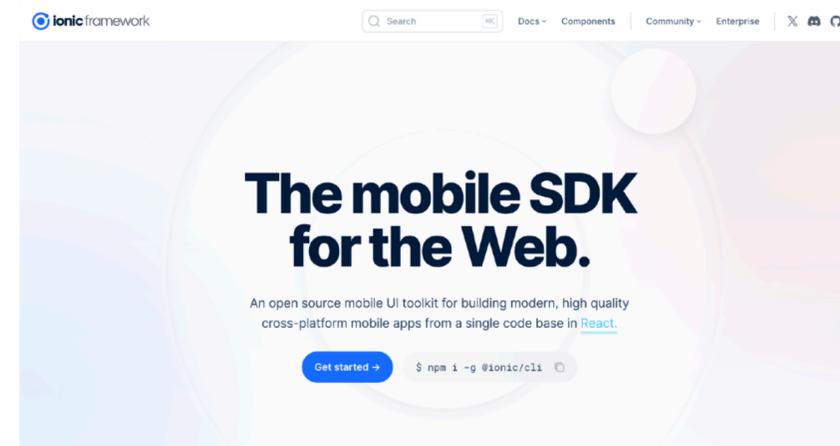
C#

Flutter



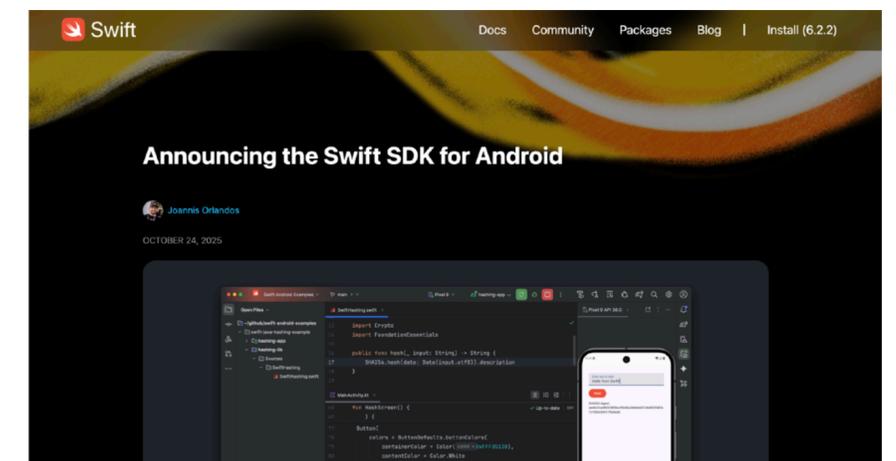
Dart

Ionic



Angular, React or Vue

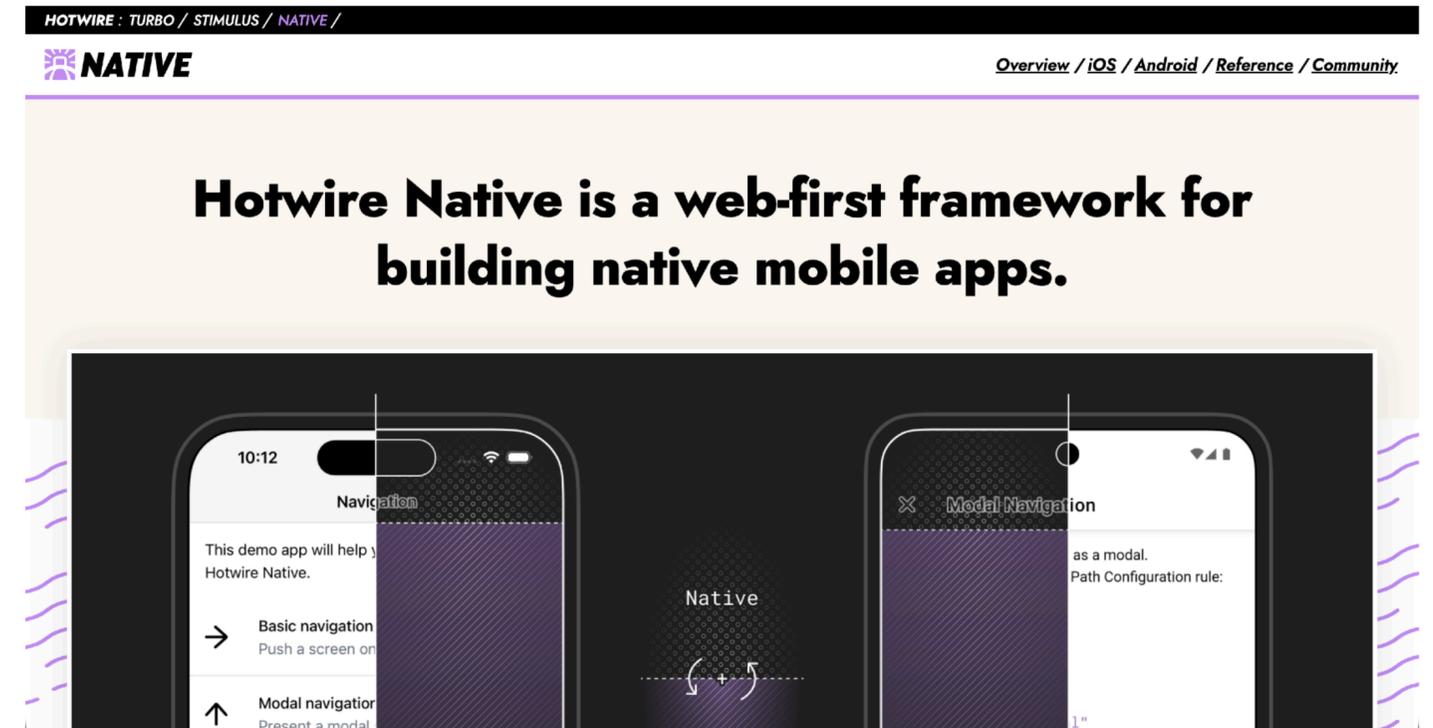
Swift for Android



Swift

Hotwire Native History

- Created by 37signals
- Turbo Native released in 2020
- Strada released in 2023
- Rebranded Hotwire Native in 2024



Hotwire Native

Who should use it

- You have the need for a web app, iOS and Android app
- You want to build with Hotwire and Turbo
- You're comfortable being an early adopter

Hotwire Native

How does it work

- Native navigation and animation
- Embedded web browser
- Doesn't look like a browser
- Views look the same as the web

Project Setup

Project Setup

- No generator like rails new
- Hotwire Native apps are Native apps

Project Setup

ios

1. Use the Xcode New Project Wizard
2. Add `hotwire-native-ios` package dependency
3. Replace `SceneDelegate.swift` with...

```
import HotwireNative
import UIKit

let rootURL = URL(string: "https://hotwire-native-demo.dev")!

class SceneDelegate: UIResponder, UIWindowSceneDelegate {
    var window: UIWindow?

    private let navigator = Navigator(configuration: .init(
        name: "main",
        startLocation: rootURL
    ))

    func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions) {
        window?.rootViewController = navigator.rootViewController
        navigator.start()
    }
}
```

SceneDelegate.swift

Project Setup

Android

1. Use the Android Studio New Project Wizard
2. Add `hotwire-native-android` dependencies to Gradle build files
3. Enable internet access in `AndroidManifest.xml`
4. Replace `MainActivity.kt` with...
5. Replace `activity_main.xml` with...

```
package com.example.myapplication

import android.os.Bundle
import android.view.View
import androidx.activity.enableEdgeToEdge
import dev.hotwire.navigation.activities.HotwireActivity
import dev.hotwire.navigation.navigator.NavigatorConfiguration
import dev.hotwire.navigation.util.applyDefaultImeWindowInsets

class MainActivity : HotwireActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        enableEdgeToEdge()
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        findViewById<View>(R.id.main_nav_host).applyDefaultImeWindowInsets()
    }

    override fun navigatorConfigurations() = listOf(
        NavigatorConfiguration(
            name = "main",
            startLocation = "https://hotwire-native-demo.dev",
            navigatorHostId = R.id.main_nav_host
        )
    )
}
```

MainActivity.kt

```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.fragment.app.FragmentContainerView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/main_nav_host"
    android:name="dev.hotwire.navigation.navigator.NavigatorHost"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:defaultNavHost="false" />
```

activity_main.xml

Project Setup

Alternative

hotwired / hotwire-native-android

<> Code Issues 12 Pull requests 4 Discussions Actions Security

Files

main + 🔍

Go to file t

- > .github
- > core
- ▼ demo
- ▼ src

hotwire-native-android / demo / 📄

 jayohms Update Gradle, plugins, and Kotlin ✓

Name
..
src

hotwired / hotwire-native-ios

<> Code Issues 9 Pull requests 15 Discussions Actions Security

Files

main + 🔍

Go to file t

- > .github
- > .swiftpm
- ▼ Demo
- ▼ Assets.xcassets

hotwire-native-ios / Demo / 📄

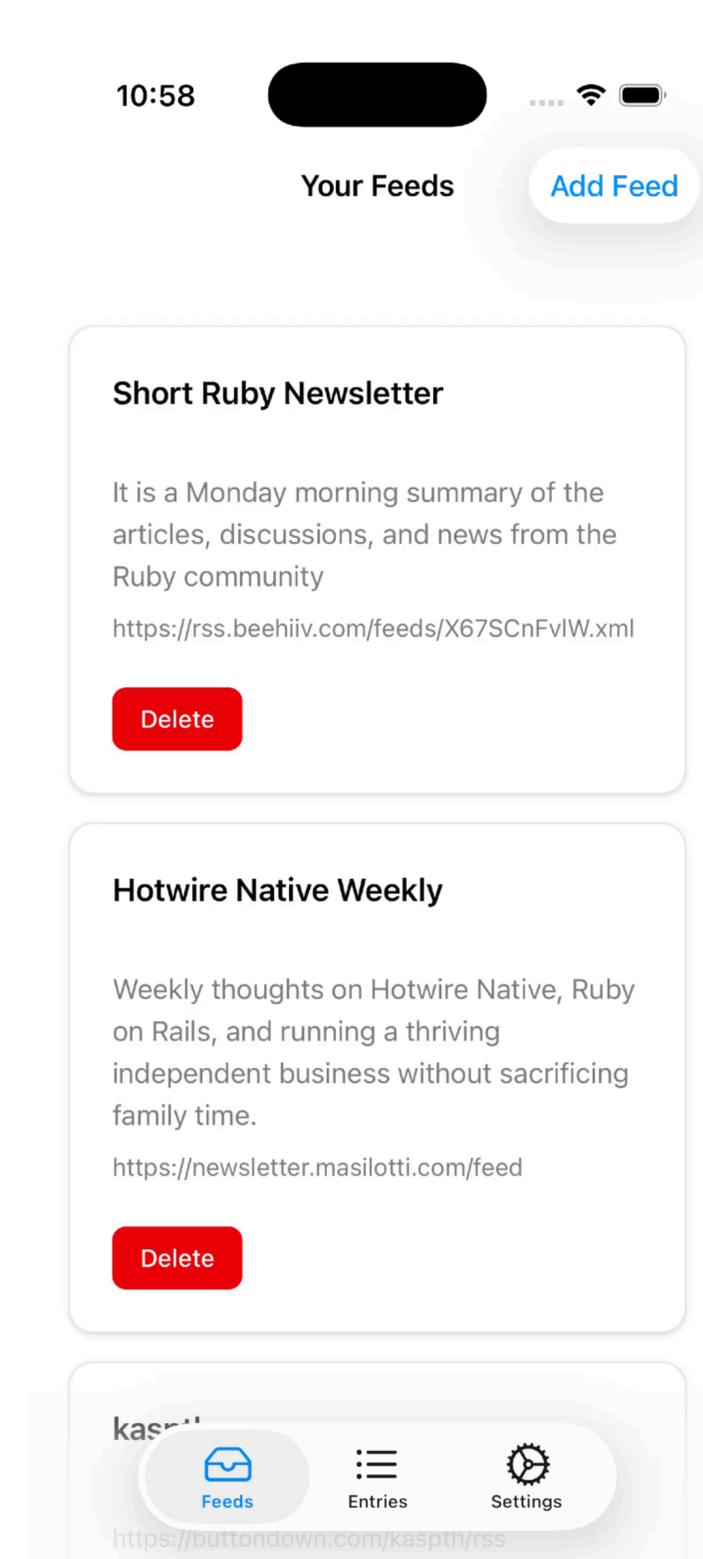
 joemasilotti Merge pull request #166 from hotwired

Name
..
Assets.xcassets

Screen Navigation

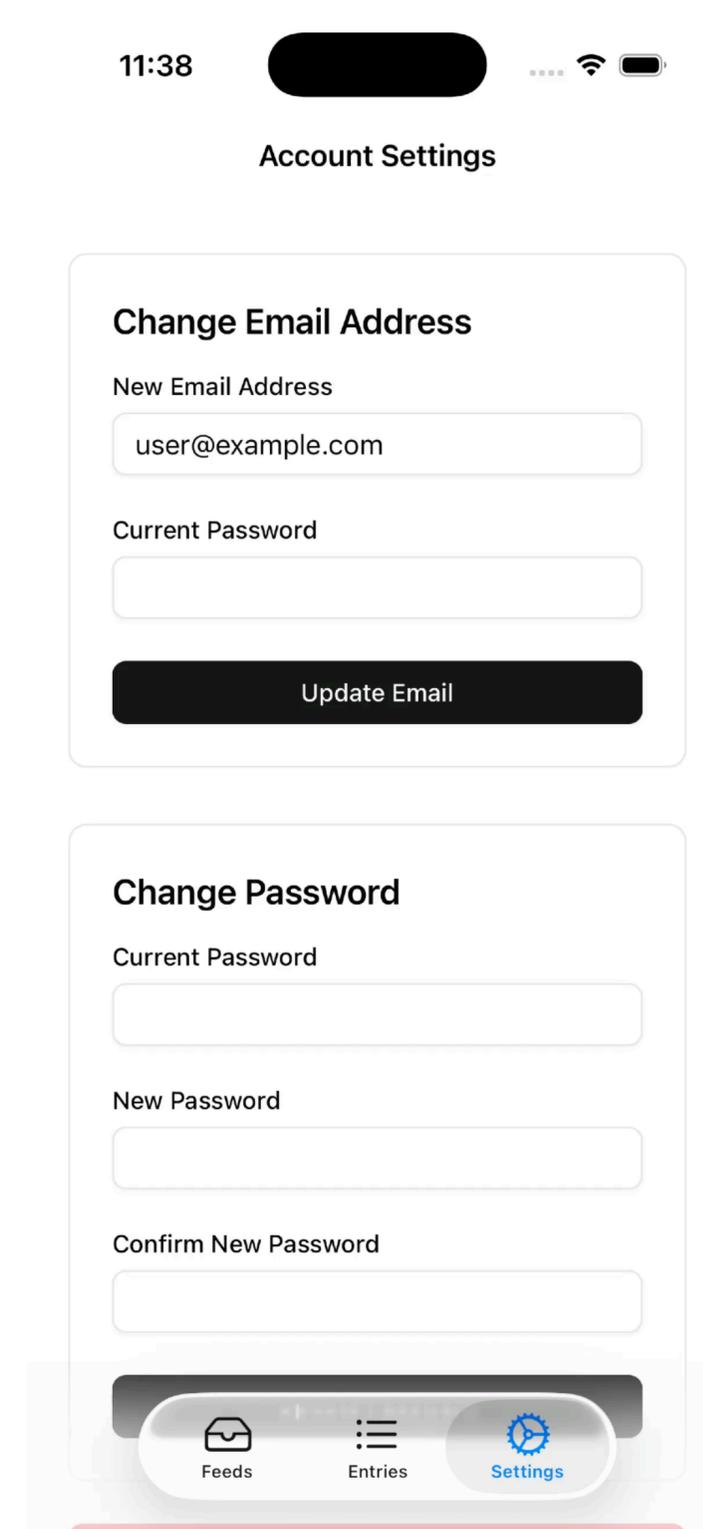
Screen Navigation

Push and Pop



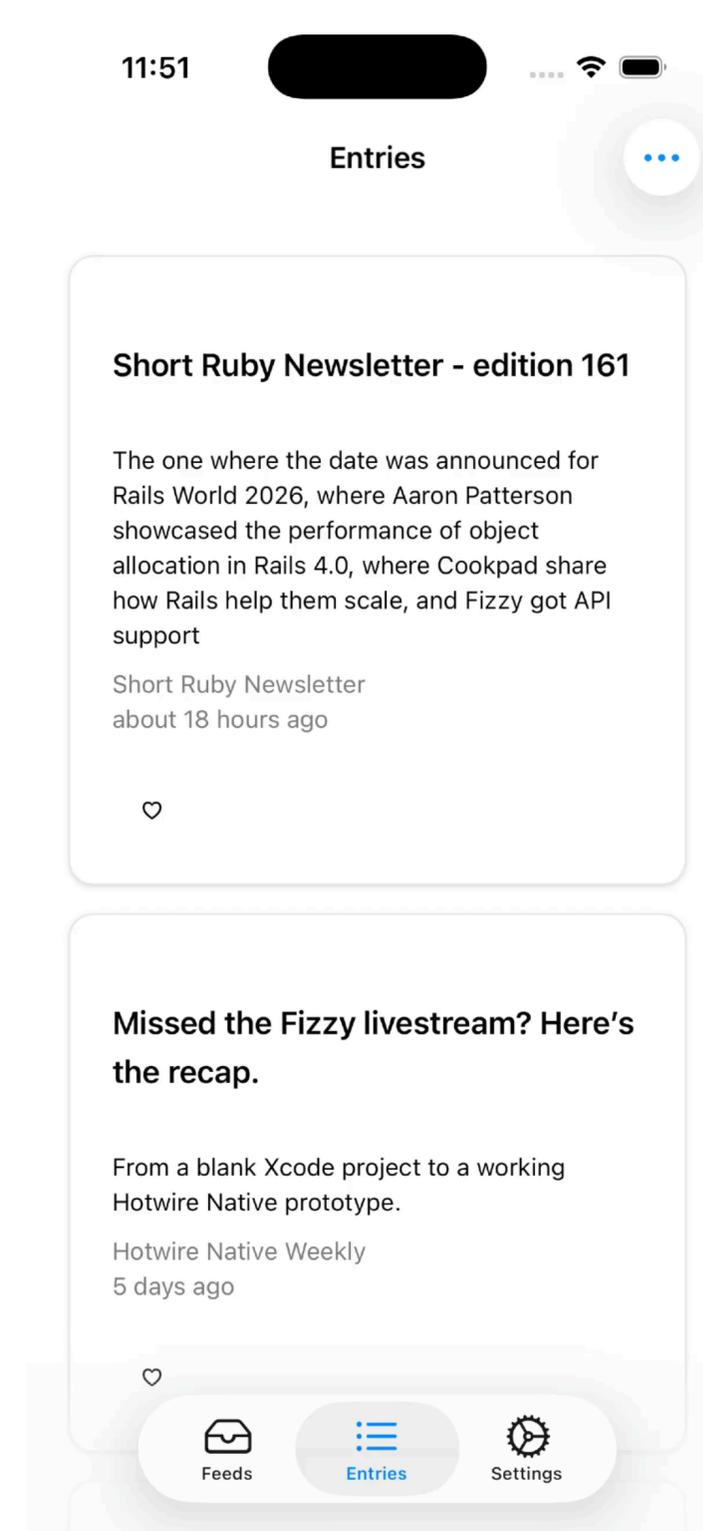
Screen Navigation

Replace



Screen Navigation

External Links



Path Configuration

Path Configuration

- Should the path use the default view or a modal?
- Should the path use a Rails view or a native view?

Path Configuration

Format

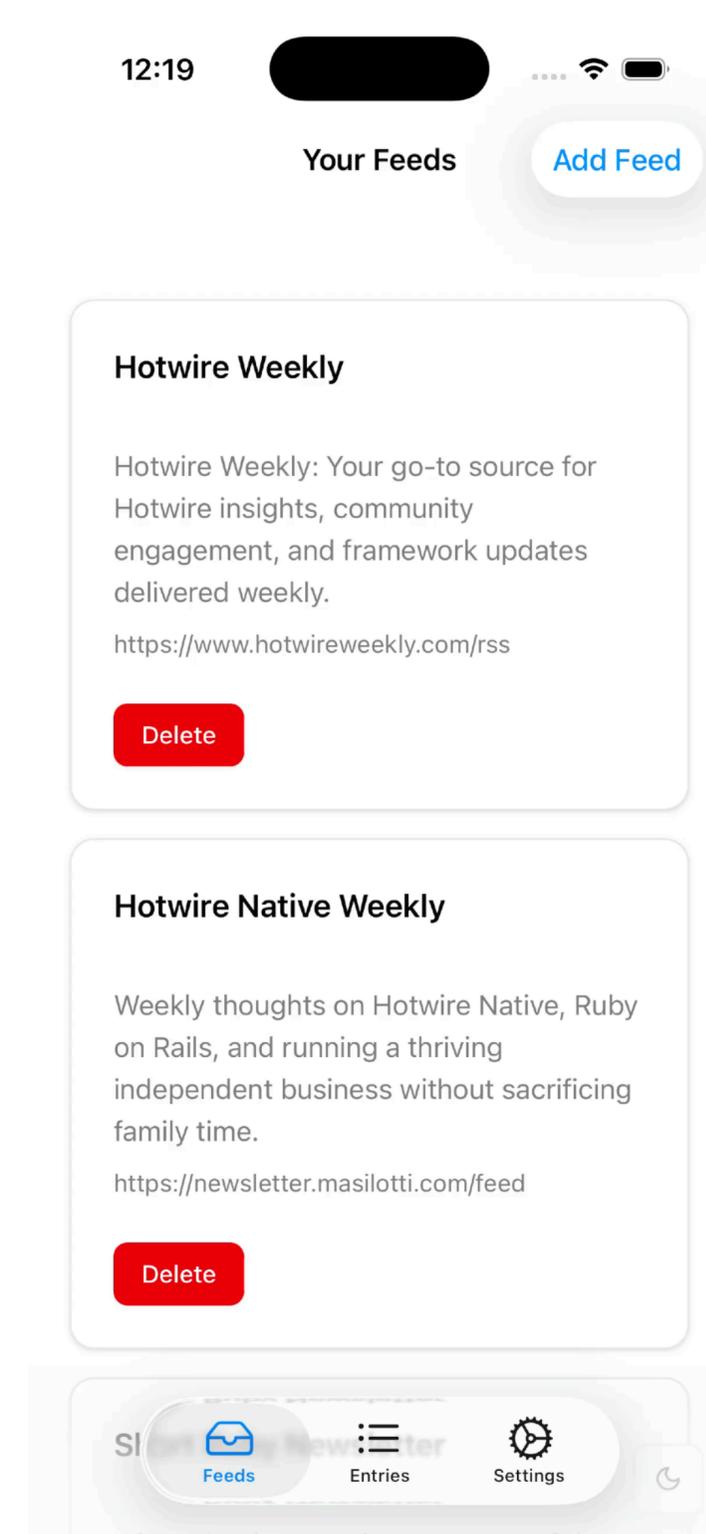
- JSON
- Stored in native app, or
- Fetched from Rails app

```
{
  "rules": [
    {
      "patterns": [
        ".*"
      ],
      "properties": {
        "context": "default",
        "pull_to_refresh_enabled": true
      }
    },
    {
      "patterns": [
        "/new$",
        "/edit$"
      ],
      "properties": {
        "context": "modal",
        "pull_to_refresh_enabled": false
      }
    }
  ]
}
```

Path Configuration

Basic Navigation

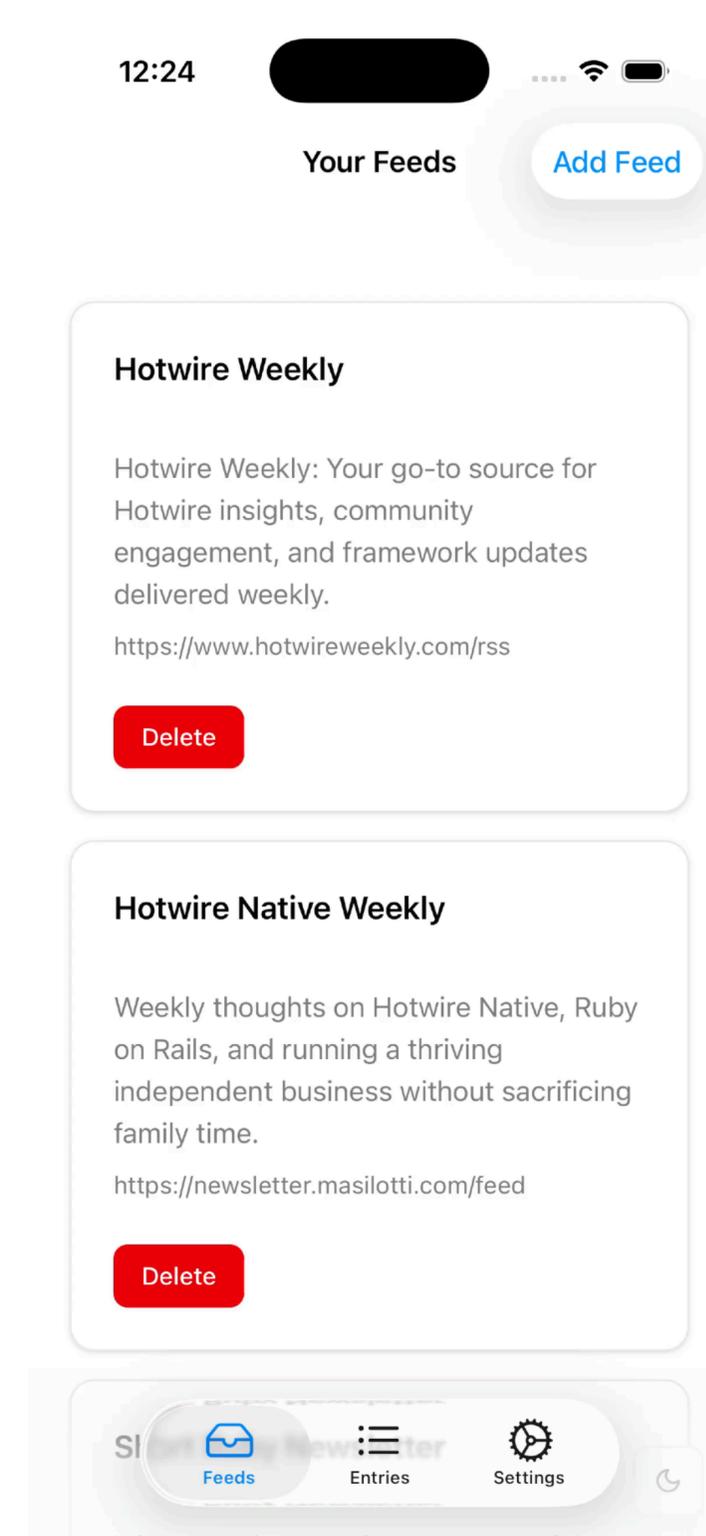
```
{
  "rules": [
    {
      "patterns": [
        ".*"
      ],
      "properties": {
        "context": "default",
        "pull_to_refresh_enabled": true
      }
    }
  ]
}
```



Path Configuration

Modal Navigation

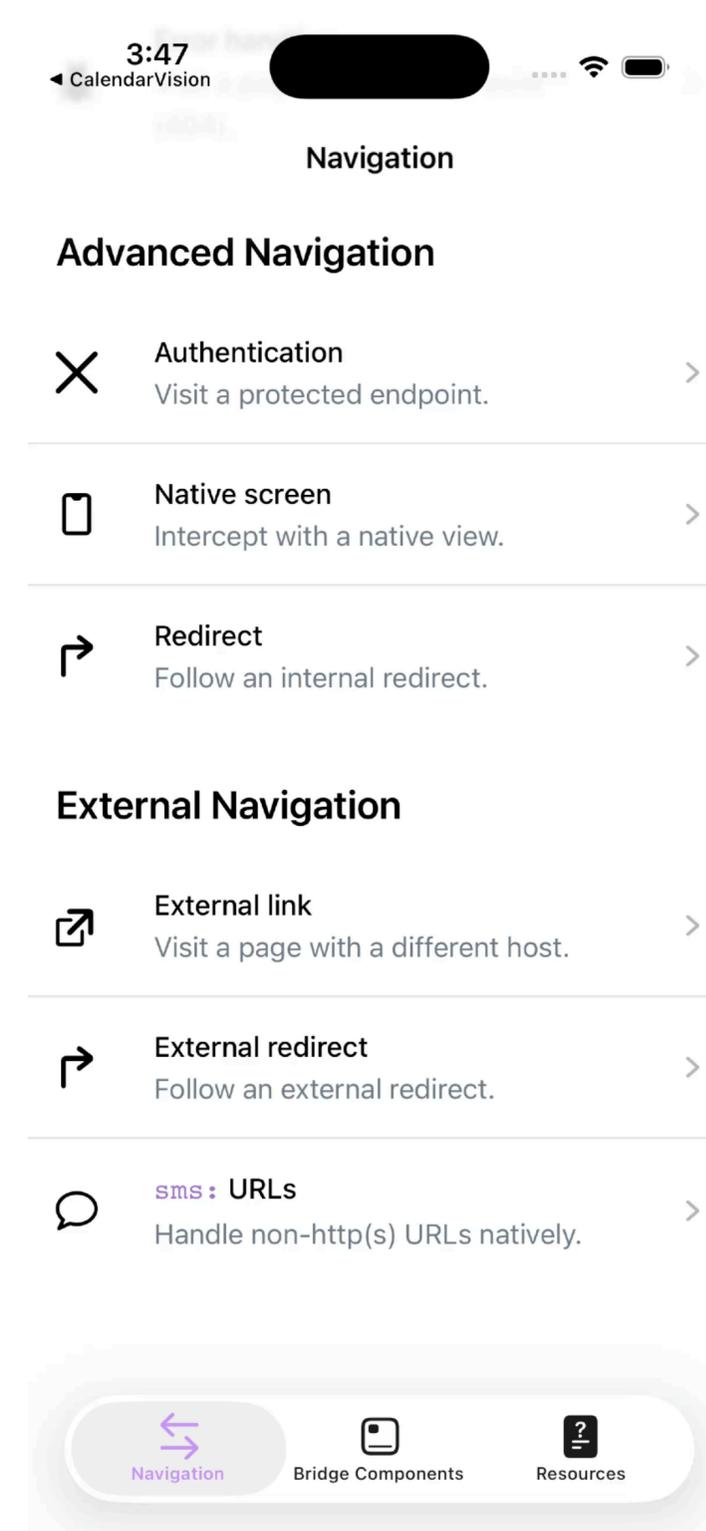
```
{  
  "rules": [  
    {  
      "patterns": [  
        "/new$",  
        "/edit$"  
      ],  
      "properties": {  
        "context": "modal",  
        "pull_to_refresh_enabled": false  
      }  
    }  
  ]  
}
```



Path Configuration

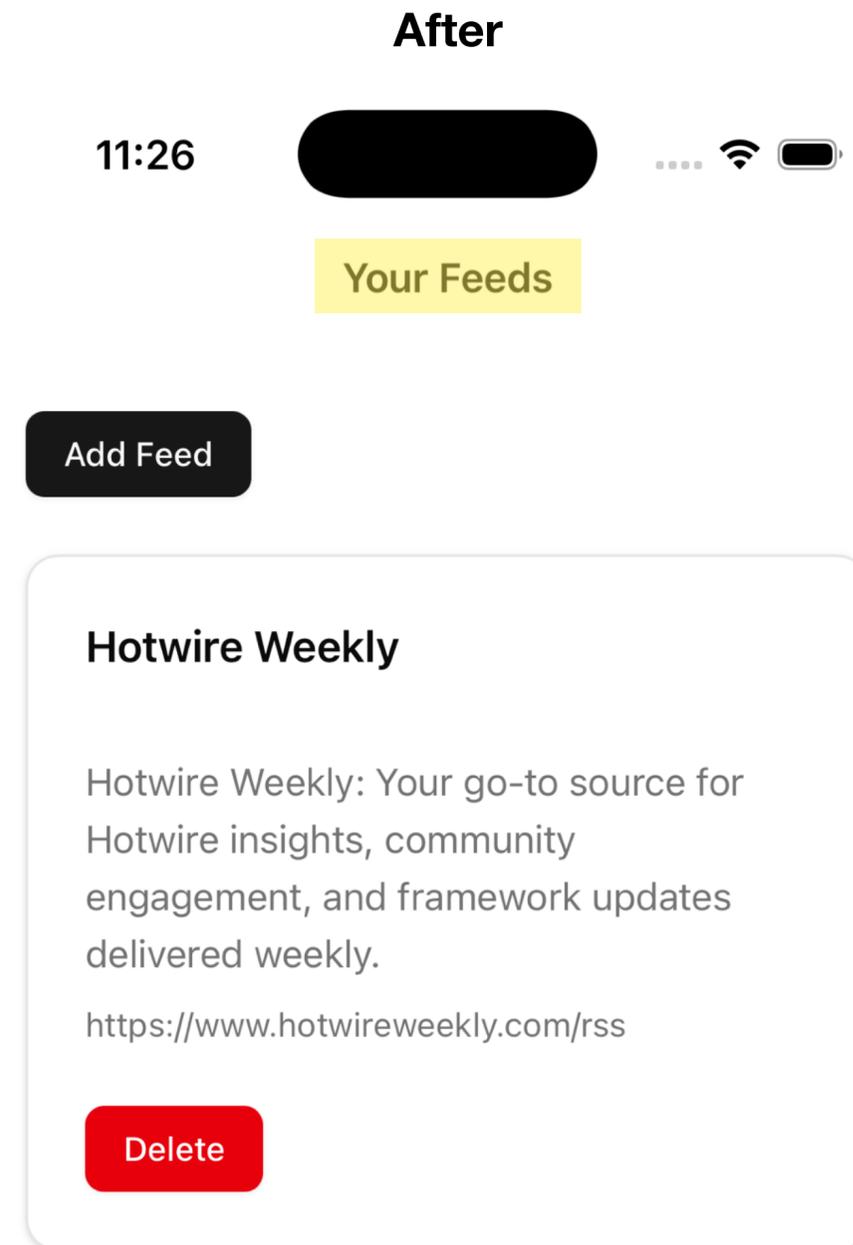
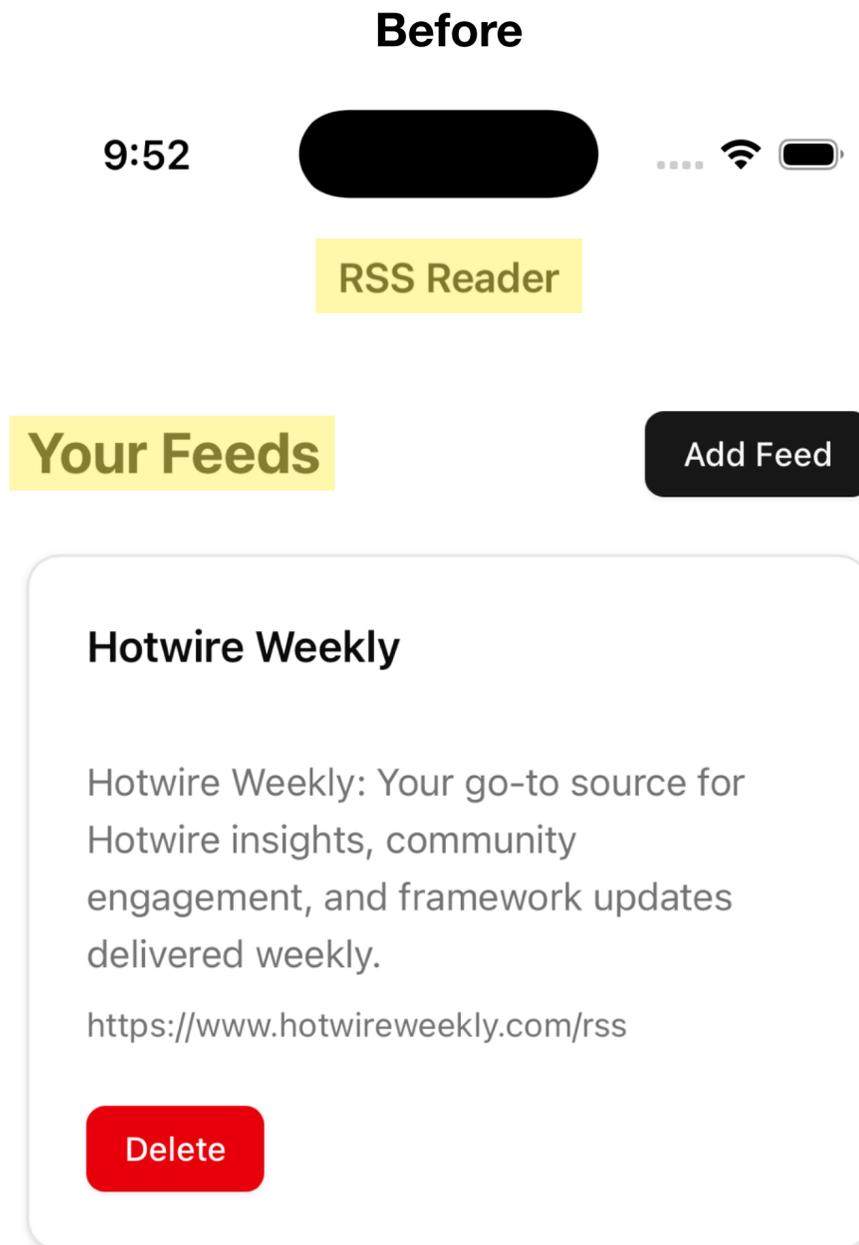
Native Screens

```
{  
  "rules": [  
    {  
      "patterns": [  
        "/numbers$"  
      ],  
      "properties": {  
        "view_controller": "numbers"  
      }  
    }  
  ]  
}
```



Navigation Bar Title

Navigation Bar Title



Navigation Bar Title

Implementation steps

- ❑ Add page-specific `title` tag in Rails code base
- ❑ Hide `h1` tag in Rails code base

Navigation Bar Title

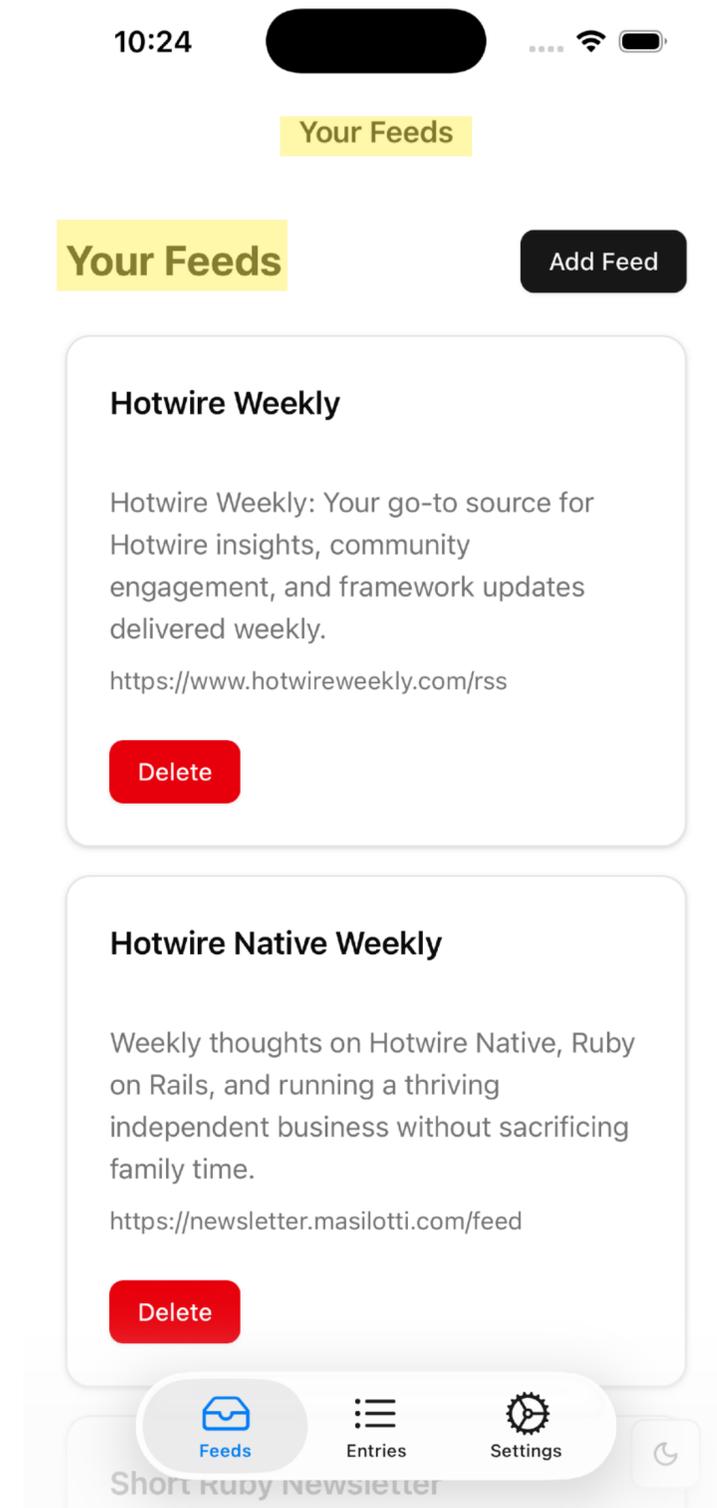
Add title tag

```
<%= content_for :title, "Your Feeds" %>
```

```
<div class="max-w-4xl mx-auto p-6 space-y-6">  
  <div class="flex justify-between items-center">  
    <h1 class="text-2xl font-bold">Your Feeds</h1>  
    <%= link_to "Add Feed", new_feed_path, class: "btn-primary" %>  
  </div>
```

```
  ...  
</div>
```

index.html.erb



Navigation Bar Title

Implementation steps

- ~~Add page-specific title tag in Rails code base~~
- Hide h1 tag in Rails code base

Navigation Bar Title

Add hotwire-native CSS variant

```
<!DOCTYPE html>
<%= tag.html(
  data: {
    hotwire_native: hotwire_native_app?,
  },
) do %>
  <head>
    <title><%= content_for(:title) || "RSS Reader" %></title>
    ...
<% end %>
```

application.html.erb

```
@variant hotwire-native {
  html[data-hotwire-native="true"] & {
    @slot
  }
}
```

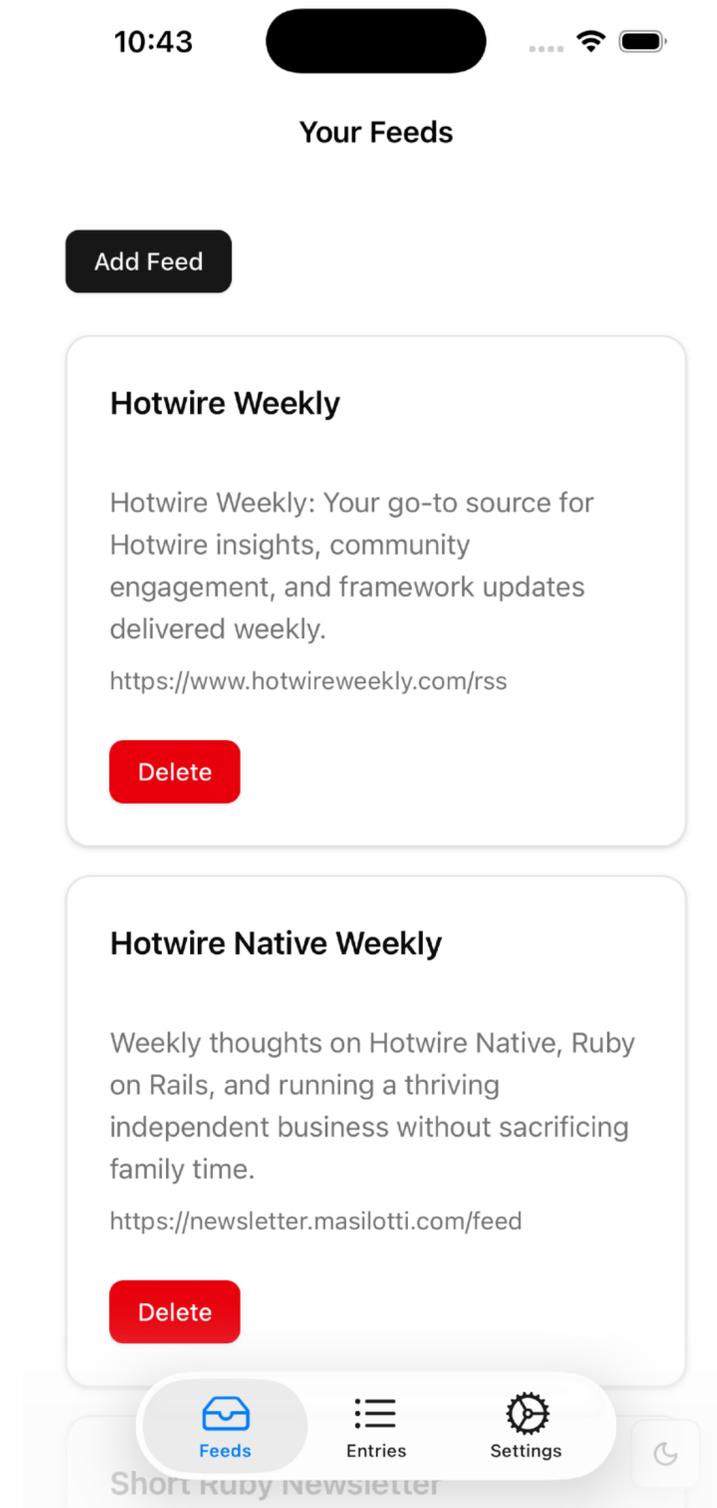
application.css

Navigation Bar Title

Hide h1 on Hotwire Native apps

```
<%= content_for :title, "Your Feeds" %>  
  
<div class="max-w-4xl mx-auto p-6 space-y-6">  
  <div class="flex justify-between items-center">  
    <h1 class="hotwire-native:hidden text-2xl font-bold">Your Feeds</h1>  
    <%= link_to "Add Feed", new_feed_path, class: "btn-primary" %>  
  </div>  
  
  ...  
</div>
```

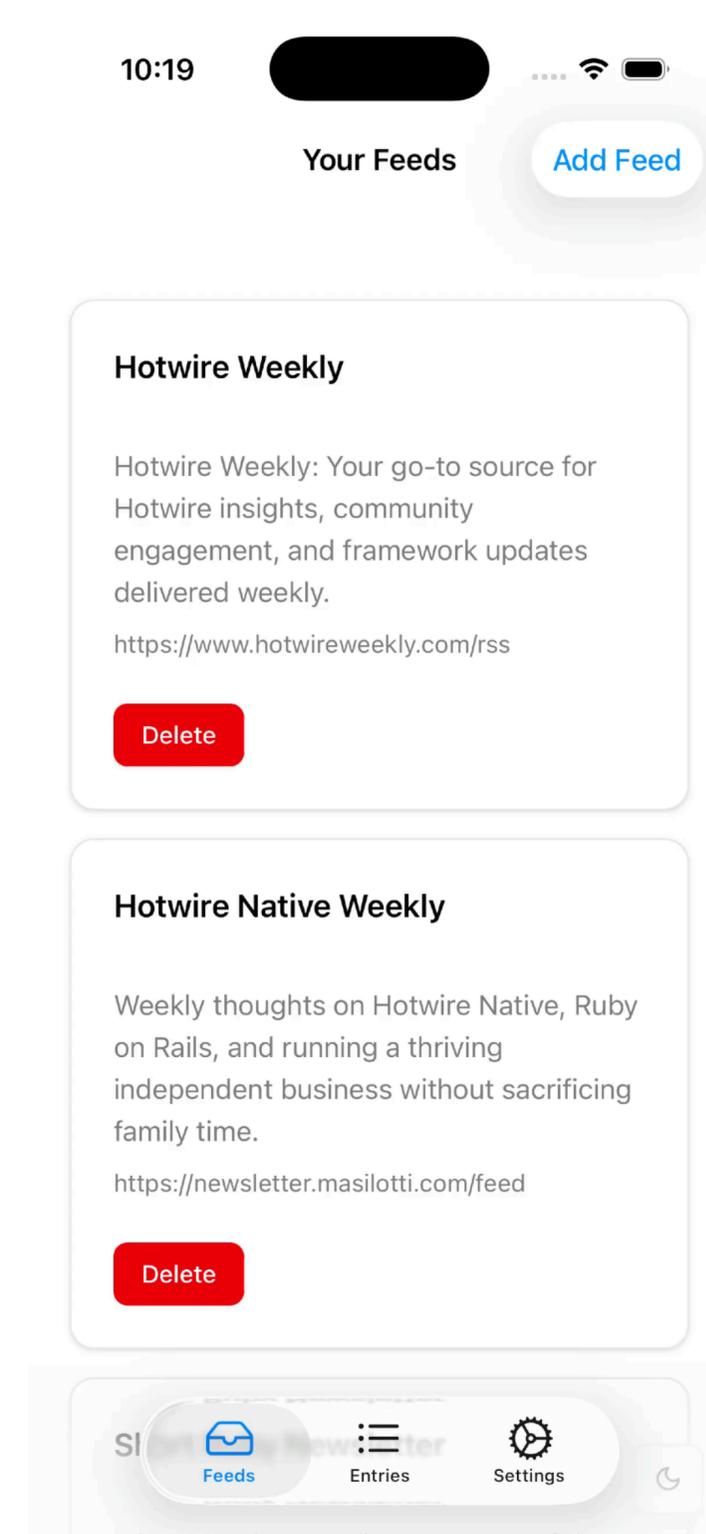
index.html.erb



Native Tab Bar

Native Tab Bar

- Built in to Hotwire Native
- Each tab is a...
 - separate web view
 - separate navigation stack



Native Tab Bar

Implementation steps

- ❑ Add native tab bar to iOS code base
- ❑ Add native tab bar to Android code base
- ❑ Hide web-based navigation in Rails code base

Native Tab Bar

ios

```
let baseUrl = URL(string: "http://localhost:3000")!

extension HotwireTab {
    static let all: [HotwireTab] = {
        var tabs: [HotwireTab] = [
            .feeds,
            ...
        ]

        return tabs
    }()

    static let feeds = HotwireTab(
        title: "Feeds",
        image: .init(systemName: "tray")!,
        url: baseUrl.appending(path: "/feeds")
    )
    ...
}
```

Tabs.swift

Native Tab Bar

ios

```
class SceneController: UIResponder {  
    ...  
    private lazy var tabBarController = HotwireTabBarController(navigatorDelegate: self)  
}  
  
extension SceneController: UIWindowSceneDelegate {  
    func scene(_ scene: UIScene, willConnectTo session: UISceneSession,  
              options connectionOptions: UIScene.ConnectionOptions) {  
        ...  
        tabBarController.load(HotwireTab.all)  
    }  
}
```

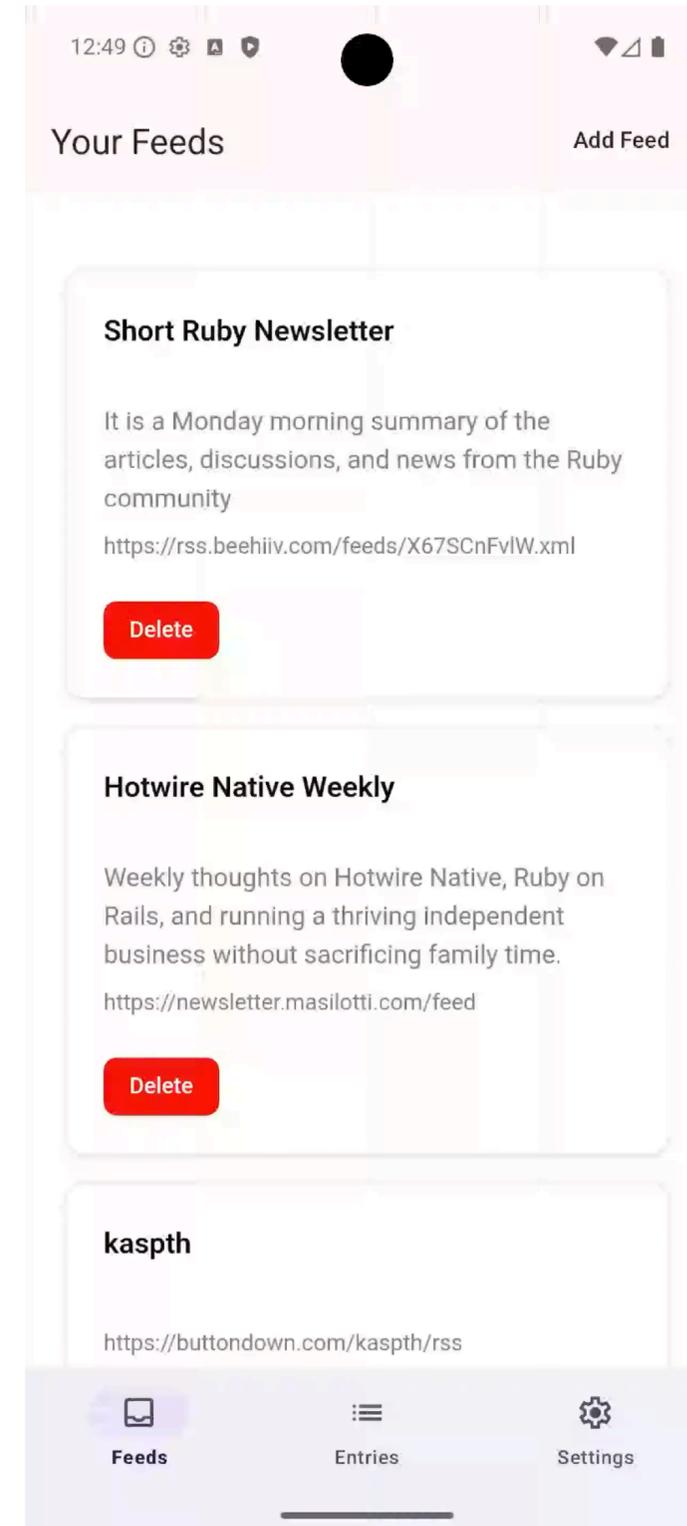
SceneController.swift

Native Tab Bar

Implementation steps

- ~~Add native tab bar to iOS code base~~
- Add native tab bar to Android code base
- Hide web-based navigation in Rails code base

Native Tab Bar Android



Native Tab Bar

Android

```
private val feeds = HotwireBottomTab(  
    title = "Feeds",  
    iconResId = R.drawable.inbox_24px,  
    configuration = NavigatorConfiguration(  
        name = "feeds",  
        navigatorHostId = R.id.feeds_nav_host,  
        startLocation = "$baseUrl/feeds"  
    )  
)  
  
...  
  
val mainTabs = listOf(  
    feeds,  
    ...  
)
```

Tabs.kt

Native Tab Bar

Android

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    ...>

    <androidx.fragment.app.FragmentContainerView
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        android:id="@+id/feeds_nav_host"
        android:name="dev.hotwire.navigation.navigator.NavigatorHost"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:defaultNavHost="false"
        app:layout_constraintBottom_toTopOf="@id/bottom_nav"
        app:layout_constraintTop_toTopOf="parent" />

    ...

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_nav"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:labelVisibilityMode="labeled"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

activity_main.xml
```

Native Tab Bar

Android

```
class MainActivity : HotwireActivity() {  
    private lateinit var bottomNavController: HotwireBottomNavController  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        ...  
  
        initializeBottomTabs()  
    }  
  
    private fun initializeBottomTabs() {  
        val bottomNavigationView = findViewById<BottomNavigationView>(R.id.bottom_nav)  
        bottomNavController = HotwireBottomNavController(this, bottomNavigationView)  
        bottomNavController.load(mainTabs, 0)  
    }  
}
```

MainActivity.kt

Native Tab Bar

Implementation steps

- ~~Add native tab bar to iOS code base~~
- ~~Add native tab bar to Android code base~~
- Hide web-based navigation in Rails code base

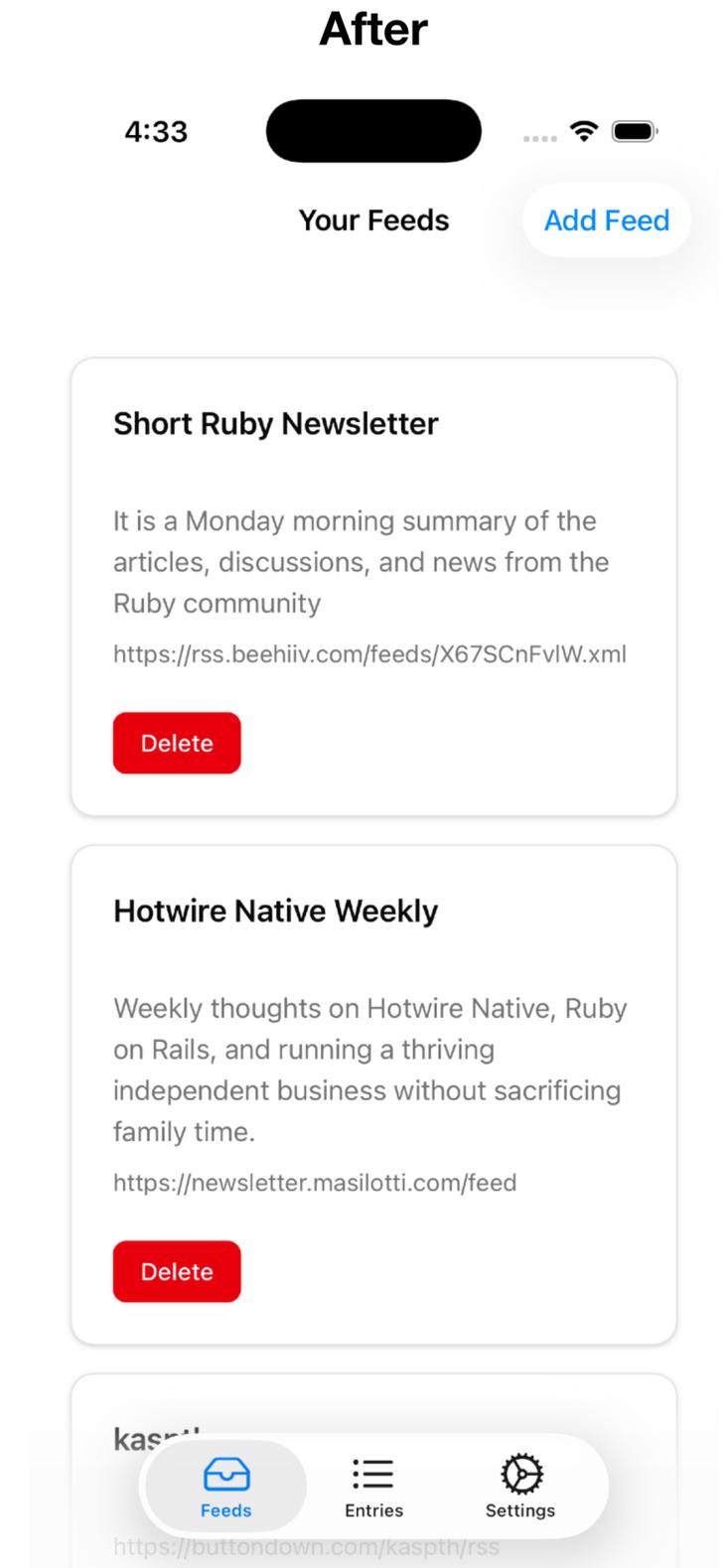
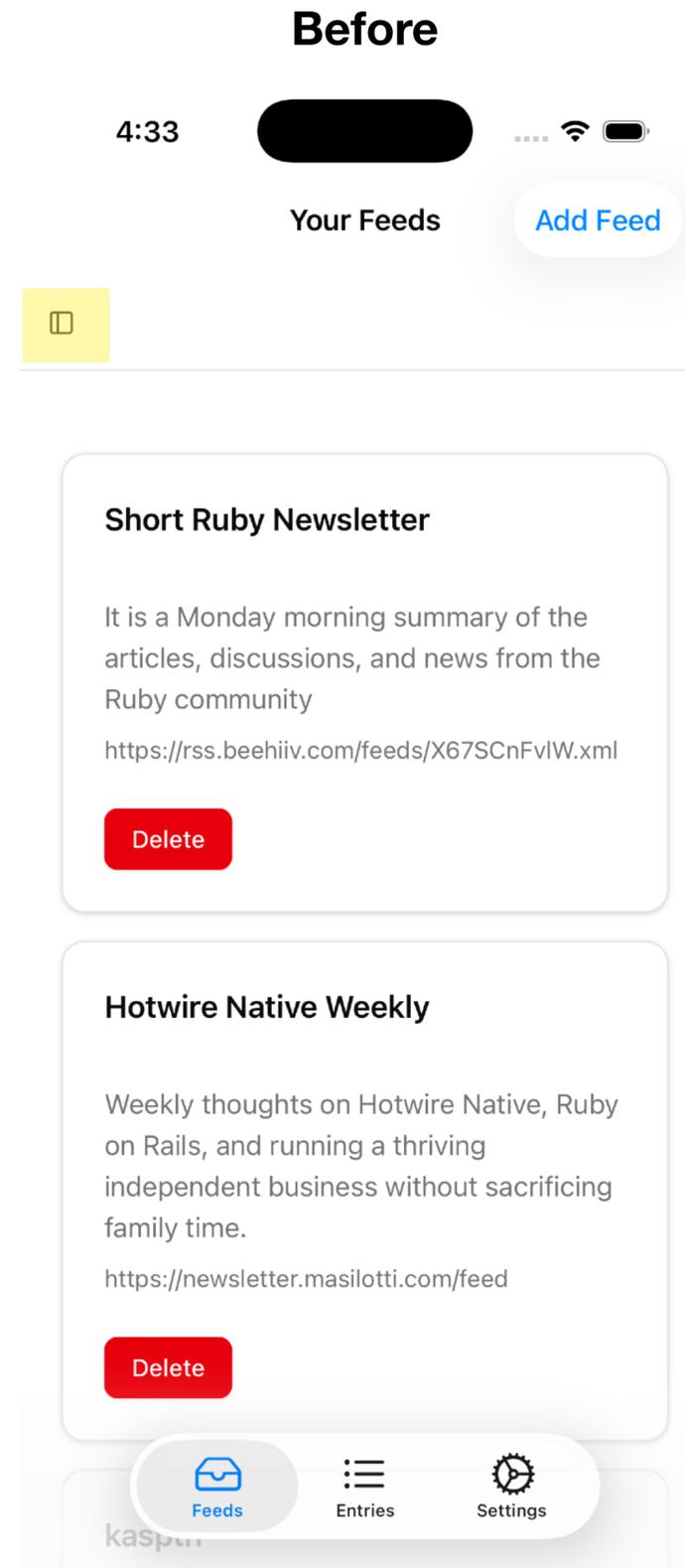
Native Tab Bar

Hide Web Navigation

```
<body>
  ...
  <main>
    <% if authenticated? %>
      <header class="hotwire-native:hidden flex ..." >
        <button...>
        </button>
      </header>
    <% end %>

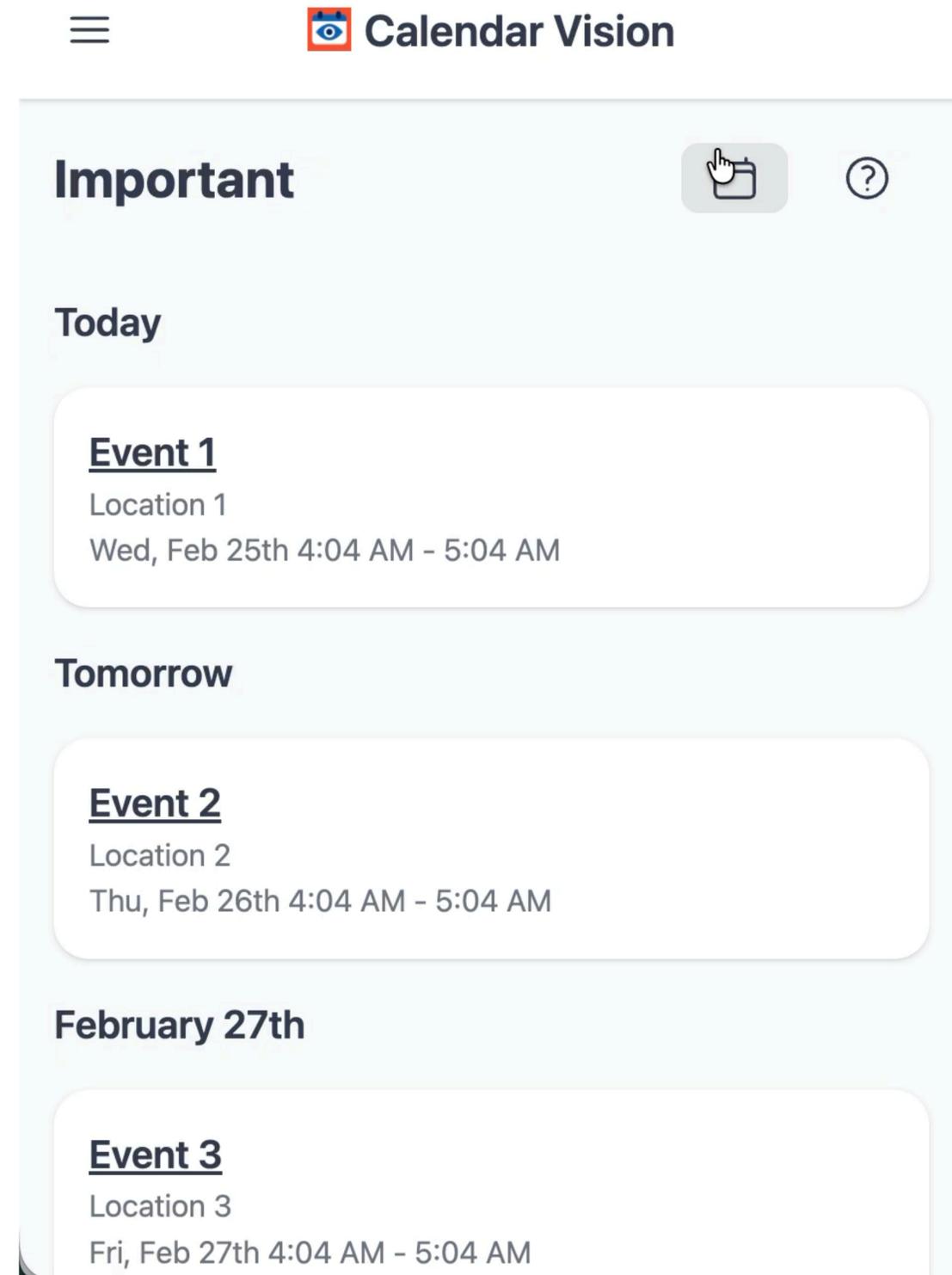
    <%= yield %>
  </main>

  ...
</body>
```



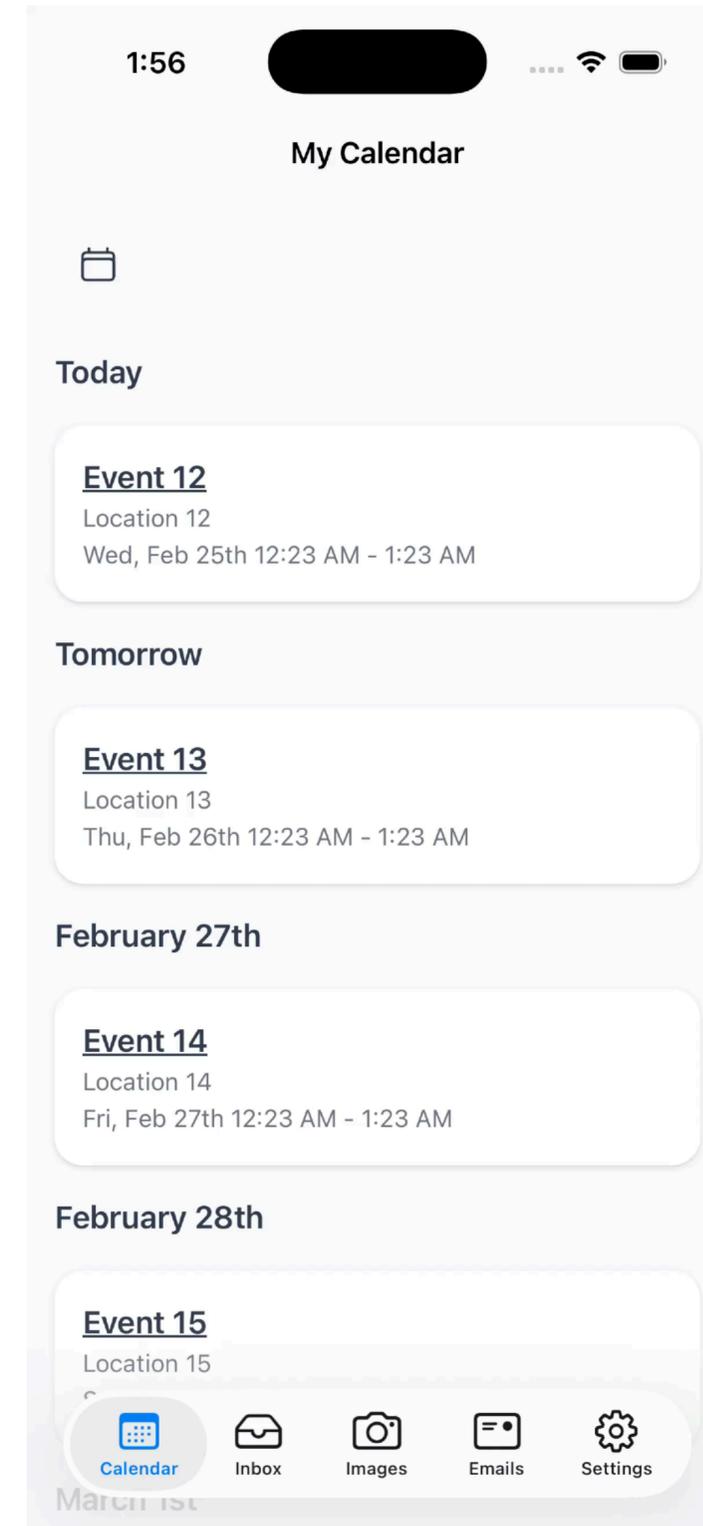
Calendar Switcher

Web



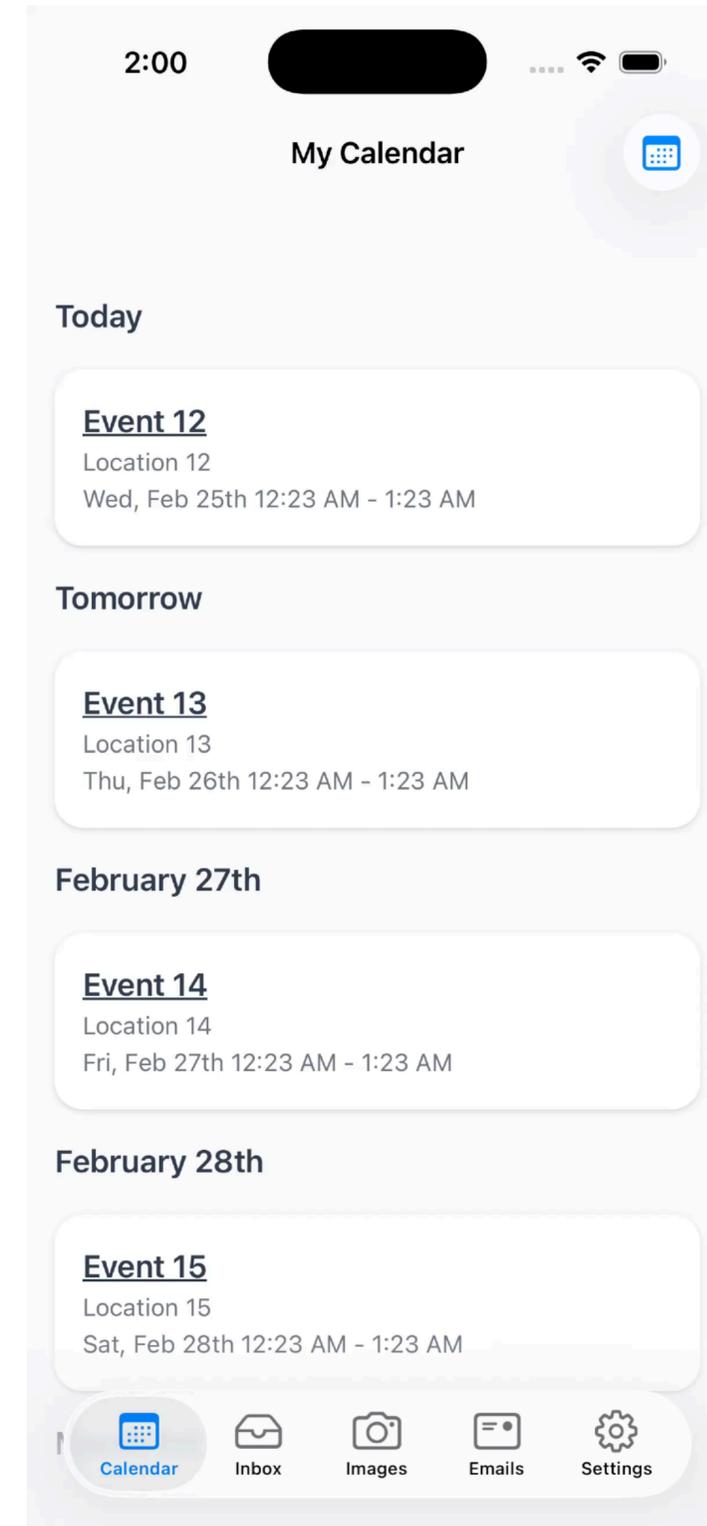
Calendar Switcher

iOS App (HTML Dialog)



Calendar Switcher

iOS App (Native Modal)



Web Implementation

```
<%= link_to calendars_path,  
  data: {  
    turbo_stream: true  
  } do %>  
  <%= render "heroicons/arrows_up_down" %>  
<% end %>
```

app/views/calendars/show.html.erb

```
class CalendarsController < ApplicationController  
  def index  
    @calendars = current_user.calendars  
    respond_to do |format|  
      format.turbo_stream  
    end  
  end  
end  
app/controllers/calendars_controller.rb
```

```
<%= turbo_stream.replace "dialog-container" do %>  
  <div id="dialog-container">  
    <dialog class="modal" open>  
      <%= render "index", calendars: @calendars %>  
    </dialog>  
  </div>  
<% end %>  
app/views/calendars/index.turbo_stream.erb
```

Native Implementation

```
<%= link_to calendars_path,  
  data: {  
    turbo_stream: !hotwire_native_app?  
  } do %>  
  <%= render "heroicons/arrows_up_down" %>  
<% end %>
```

app/views/calendars/show.html.erb

```
class ConfigurationsController < ApplicationController  
  def ios_v1  
    render json: {  
      rules: [  
        {  
          patterns: [  
            "#{calendars_path}$"  
          ],  
          properties: {  
            context: "modal",  
            presentation: "default",  
            pull_to_refresh_enabled: false  
          }  
        }  
      ]  
    }  
  end  
end
```

app/controllers/configurations_controller.rb

```
class CalendarsController < ApplicationController  
  def index  
    @calendars = current_user.calendars  
    respond_to do |format|  
      format.html  
      format.turbo_stream  
    end  
  end  
end
```

app/controllers/calendars_controller.rb

```
<%= render "index", calendars: @calendars %>
```

app/views/calendars/index.html.erb

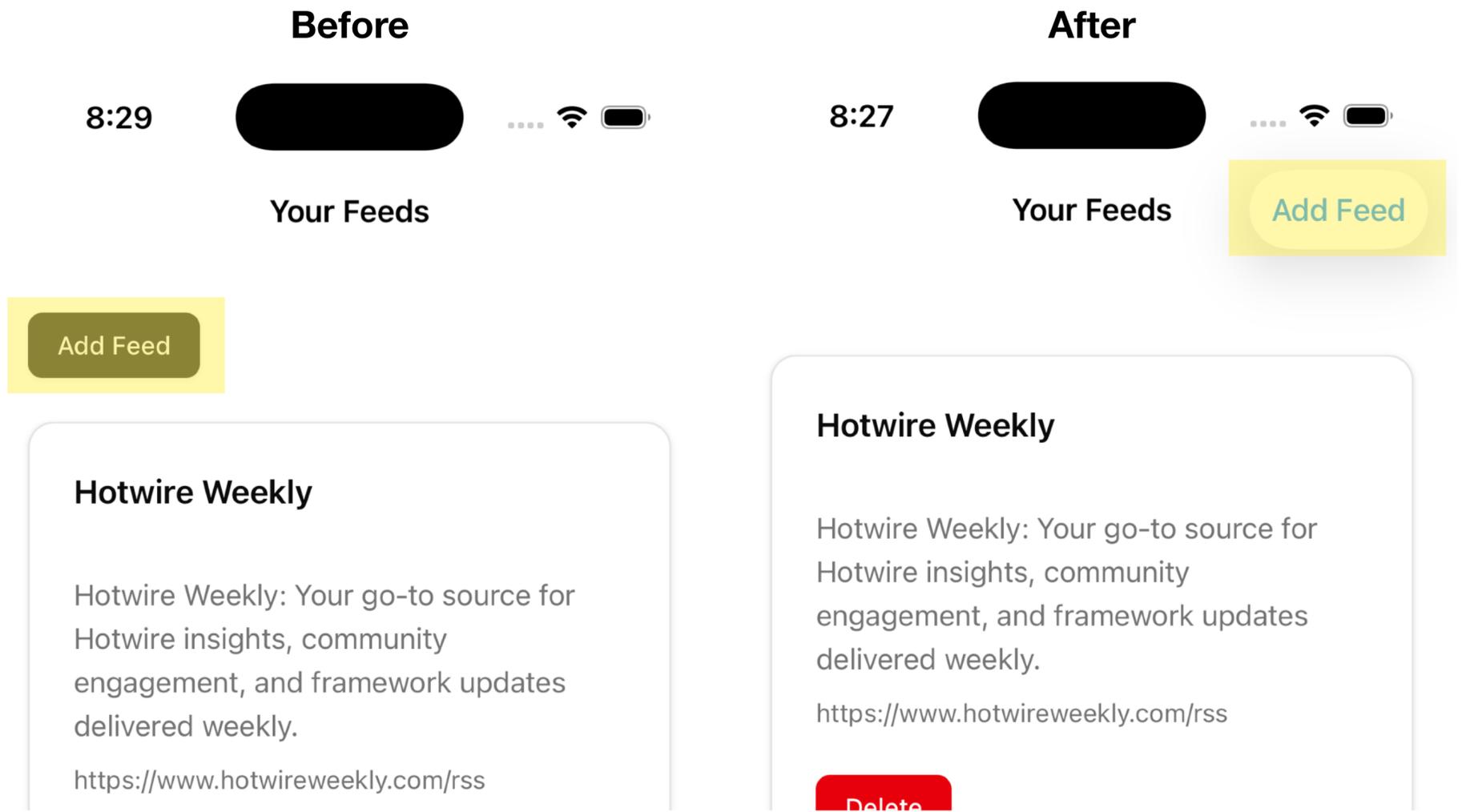
Bridge Components

Bridge Components

- Formerly called Strada
- Three parts
 - Stimulus controller
 - iOS component
 - Android fragment

Bridge Components

Navigation Bar Button



Bridge Components

Navigation Bar Button

- ❑ Add Stimulus controller
- ❑ Update the `link_to` helper to use the Stimulus controller
- ❑ Add iOS component
- ❑ Add Android component

Bridge Components

Navigation Bar Button

```
import { BridgeComponent } from "@hotwired/hotwire-native-bridge"

export default class extends BridgeComponent {
  static component = "button"

  connect() {
    super.connect()

    const element = this.bridgeElement
    const title = element.bridgeAttribute("title")
    this.send("connect", {title}, () => {
      this.element.click()
    })
  }
}
```

app/javascript/controllers/bridge/button_controller.js

Bridge Components

Navigation Bar Button

```
<%= link_to "Add Feed", new_feed_path,  
  class: "hotwire-native:hidden btn-primary",  
  data: {  
    controller: "bridge--button",  
    bridge_title: "Add Feed",  
  } %>
```

app/views/feeds/index.html.erb

Bridge Components

Navigation Bar Button

```
import HotwireNative
import UIKit

final class ButtonComponent: BridgeComponent {
    override class var name: String { "button" }

    override func onReceive(message: Message) {
        guard let viewController else { return }
        addButton(via: message, to: viewController)
    }

    private var viewController: UIViewController? {
        delegate?.destination as? UIViewController
    }

    private func addButton(via message: Message, to viewController: UIViewController) {
        guard let data: MessageData = message.data() else { return }

        let action = UIAction { [unowned self] _ in
            self.reply(to: "connect")
        }
        let item = UIBarButtonItem(title: data.title, primaryAction: action)
        viewController.navigationItem.rightBarButtonItem = item
    }
}

private extension ButtonComponent {
    struct MessageData: Decodable {
        let title: String
    }
}
```

ButtonComponent.swift

Bridge Components

Navigation Bar Button

```
import HotwireNative
import UIKit

@main
class AppDelegate: UIResponder, UIApplicationDelegate {
    func application(_ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        Hotwire.registerBridgeComponents([
            ButtonComponent.self
        ])
        return true
    }
}
```

AppDelegate.swift

Bridge Components

Navigation Bar Button

```
class ButtonComponent(
    name: String,
    private val delegate: BridgeDelegate<HotwireDestination>
) : BridgeComponent<HotwireDestination>(name, delegate) {

    override fun onReceive(message: Message) {
        // Handle incoming messages based on the message `event`.
        when (message.event) {
            "connect" -> handleConnectEvent(message)
            else -> Log.w("ButtonComponent", "Unknown event for message: $message")
        }
    }

    private fun handleConnectEvent(message: Message) {
        val data = message.data<MessageData>() ?: return

        // Write native code to display a native submit button in the
        // toolbar displayed in the delegate.destination. Use the
        // incoming data.title to set the button title.
    }

    private fun performButtonClick(): Boolean {
        return replyTo("connect")
    }

    // Use kotlinx.serialization annotations to define a serializable
    // data class that represents the incoming message.data json.
    @Serializable
    data class MessageData(
        @SerializedName("title") val title: String
    )
}
```

ButtonComponent.kt

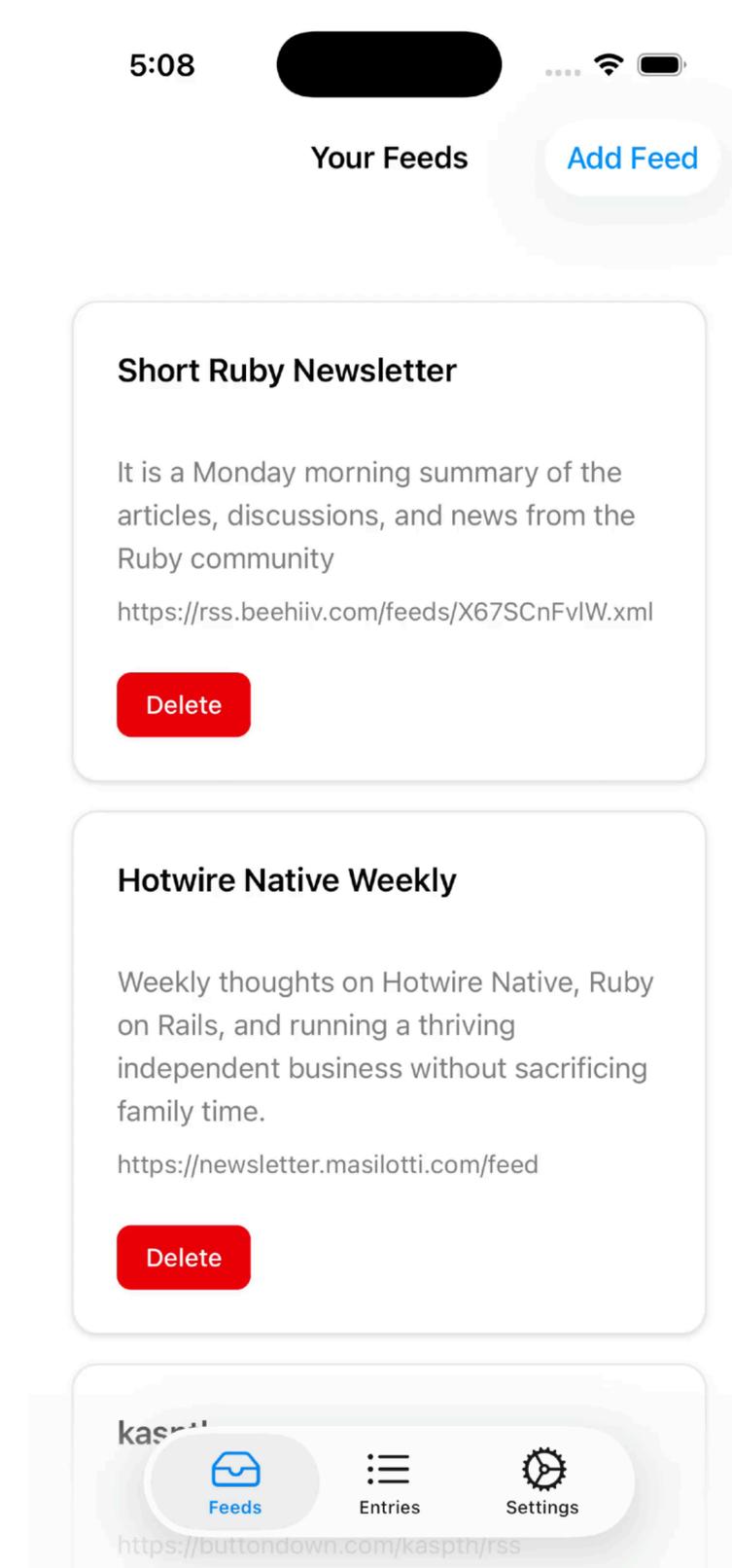
Bridge Components

Navigation Bar Button

```
Hotwire.registerBridgeComponents(  
  BridgeComponentFactory("button", ::ButtonComponent)  
)
```

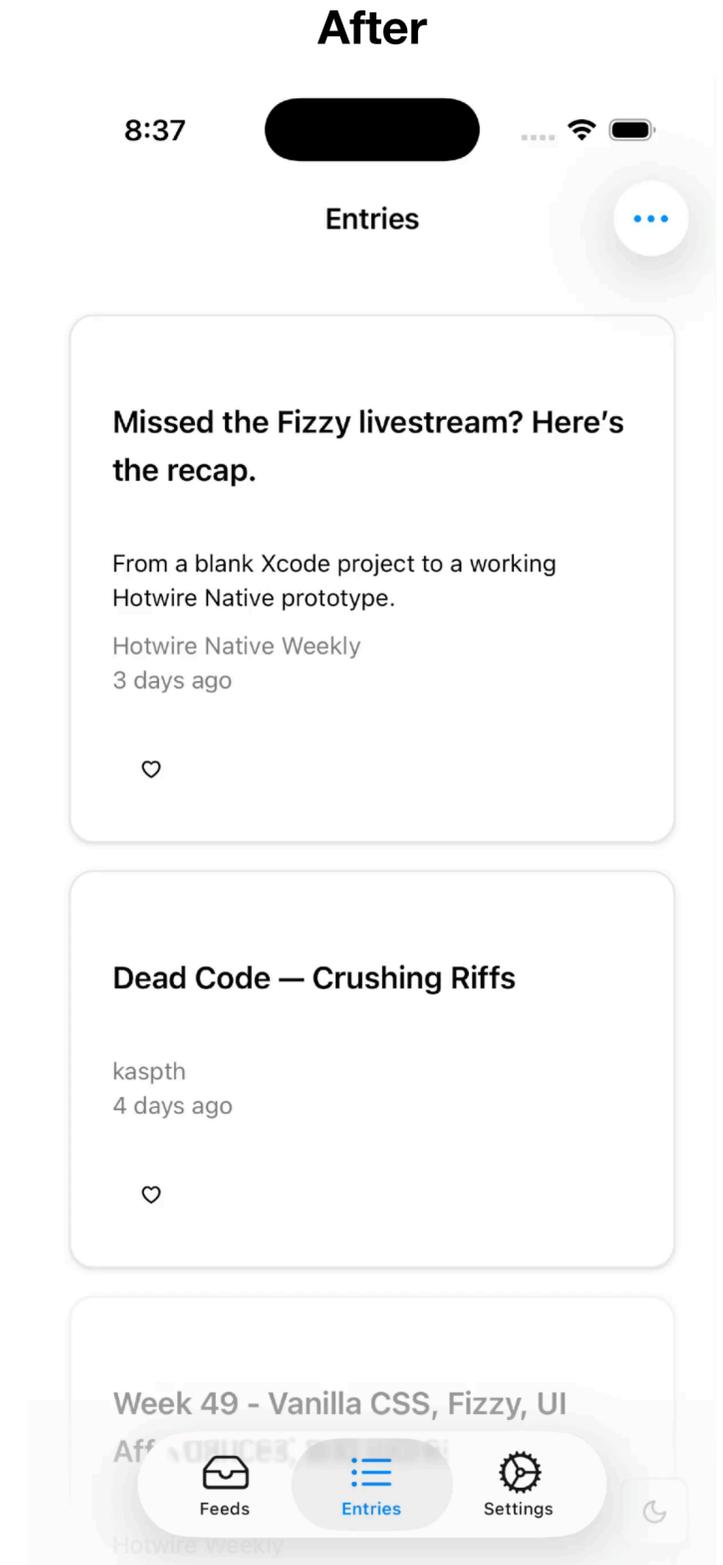
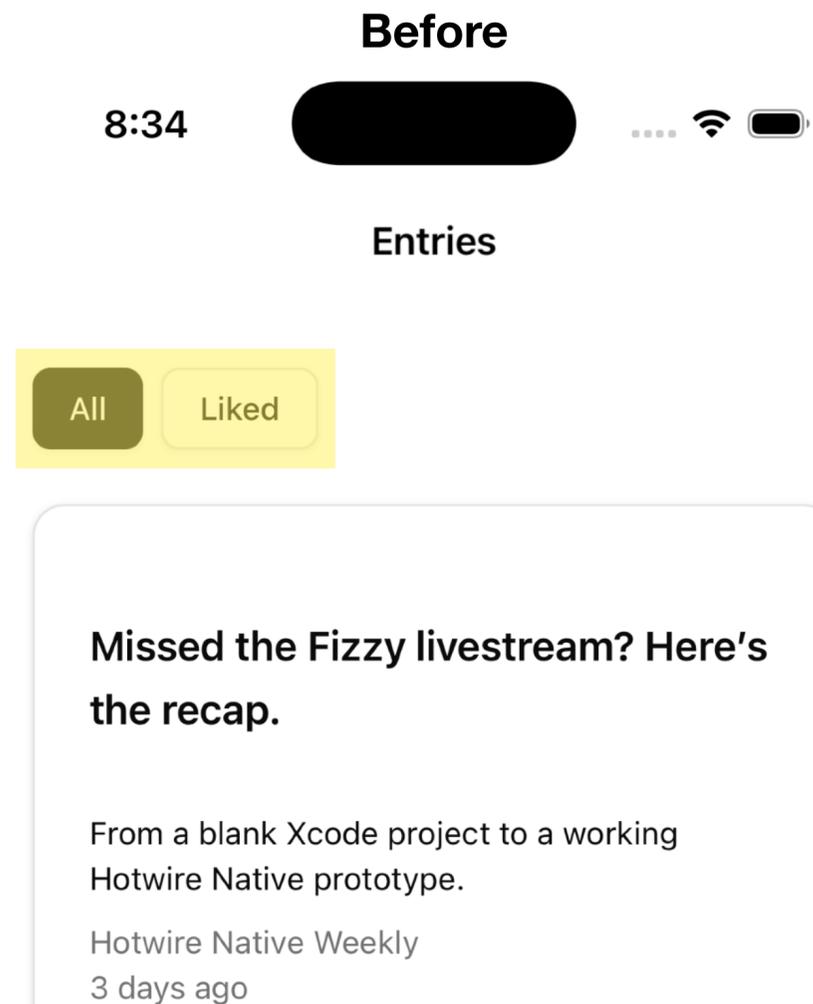
Bridge Components

Navigation Bar Button



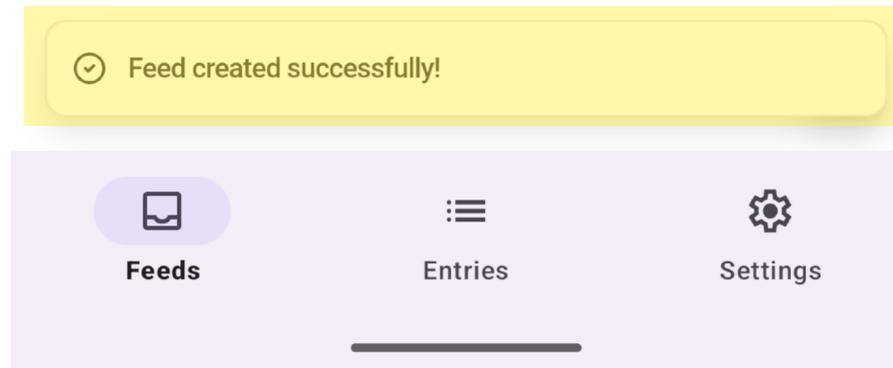
Bridge Components

Navigation Bar Menu

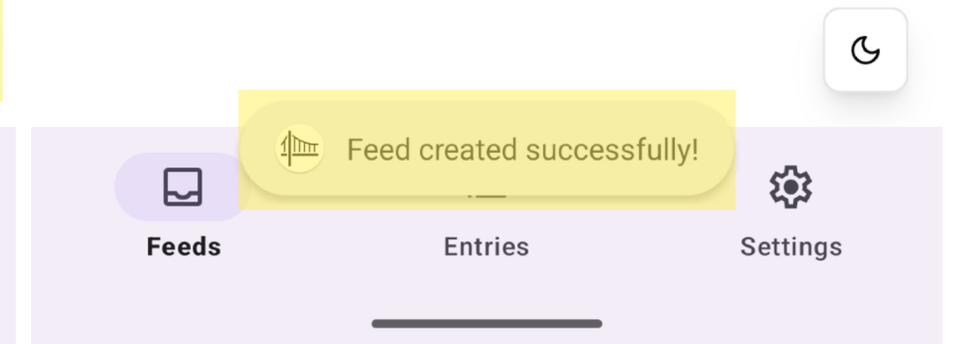


Bridge Components Toast Messages

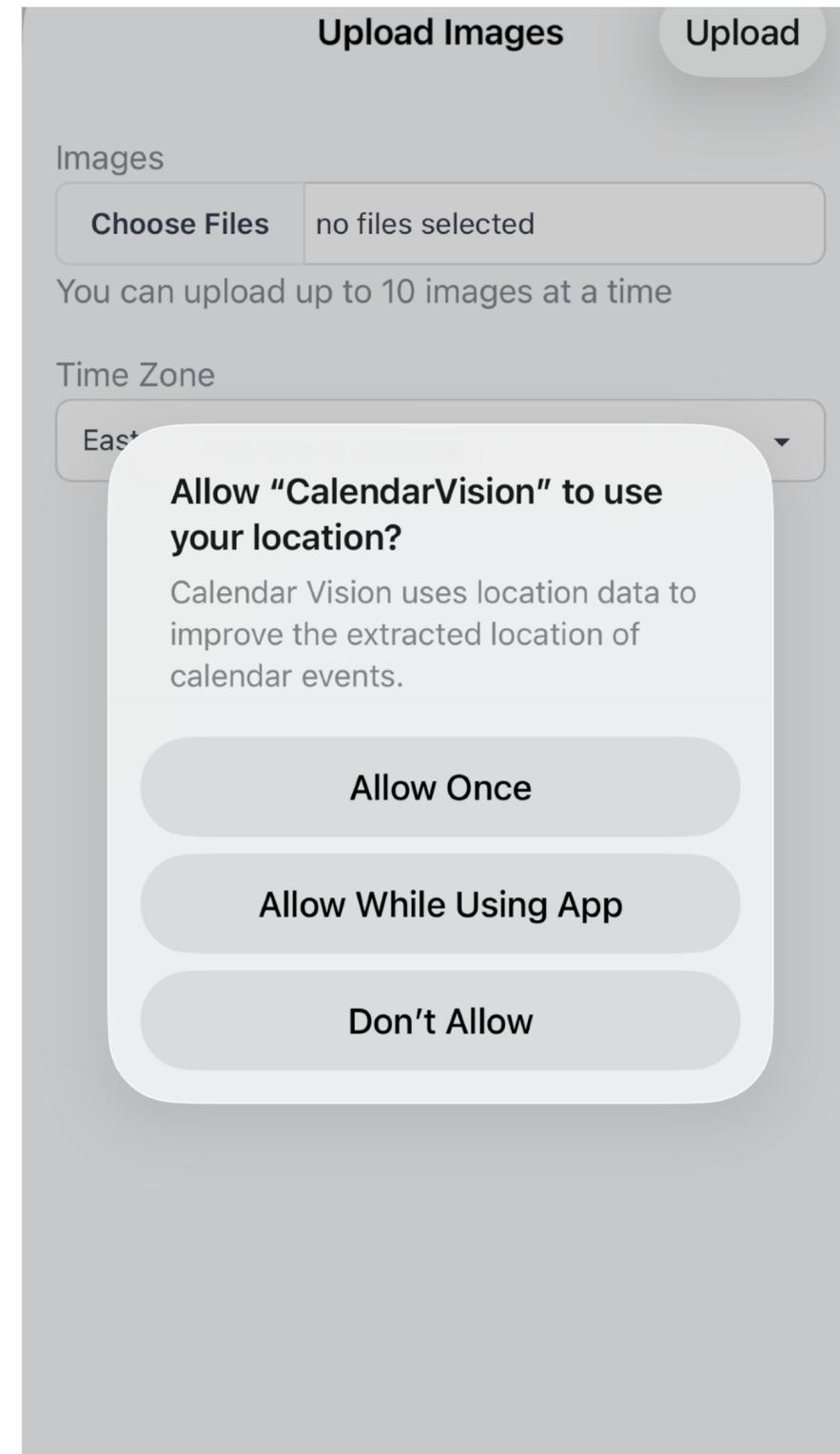
Before



After



Bridge Components Request Permissions



Bridge Components

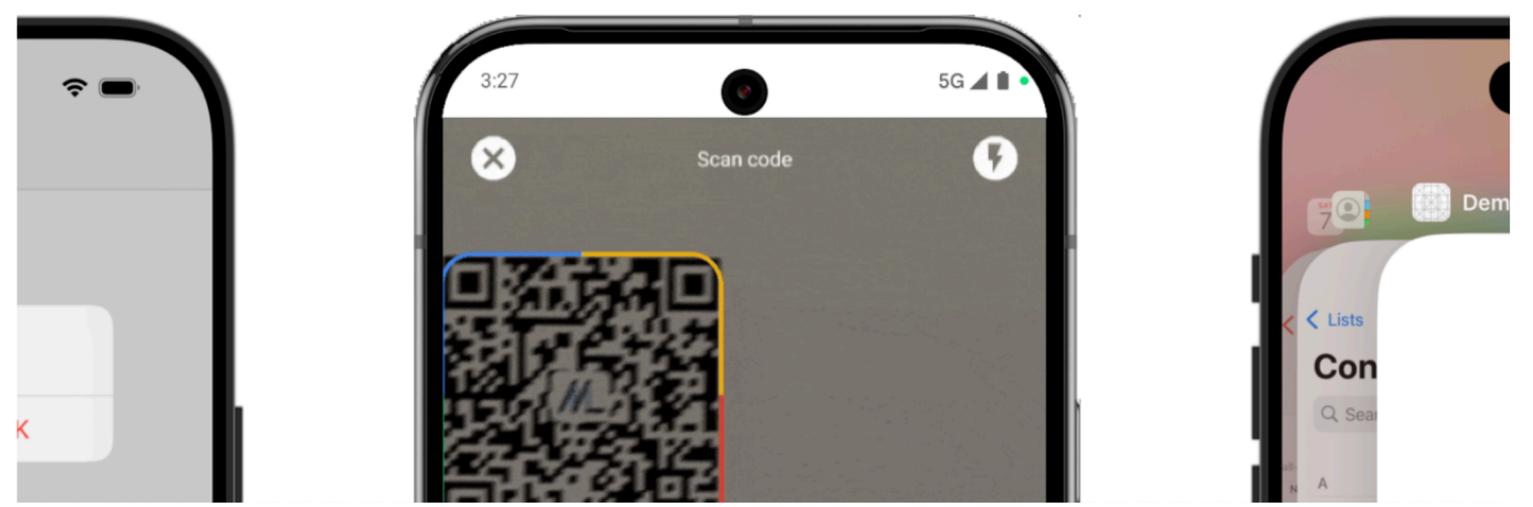
Joe Masilotti's library

Components Newsletter Installation PRO

Bridge Components

Native integrations for Hotwire Native apps

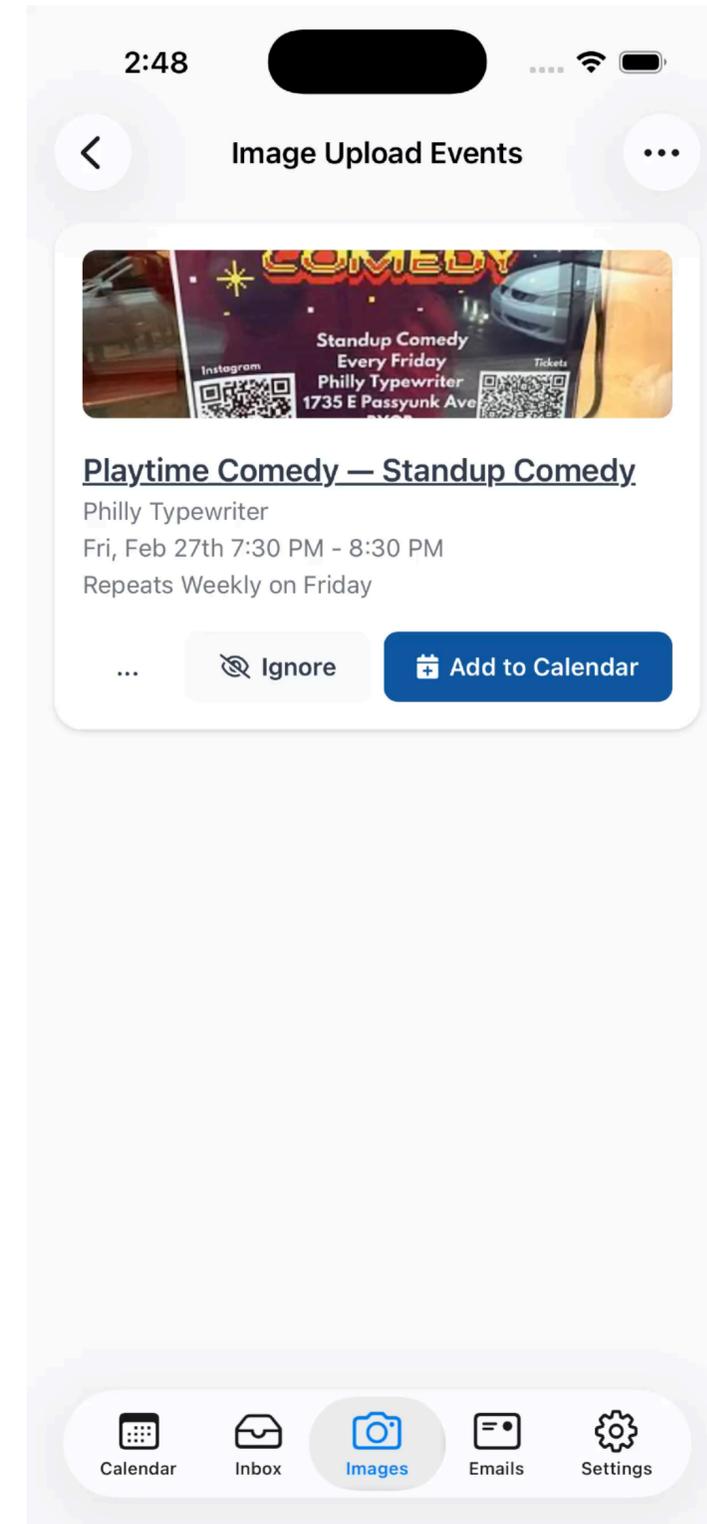
Build iOS and Android features with HTML, *no Swift or Kotlin required.*



<https://masilotti.com/bridge-components/>

Native Components

Add to Calendar iOS



Handle URL with a native component

```
class SceneController: UIResponder {
    private var addToCalendarController: AddToCalendarController?
}

extension SceneController: UINavigationControllerDelegate {
    func handle(proposal: VisitProposal, from navigator: UINavigationController) -> ProposalResult {
        switch proposal.viewController {
        case "add_to_calendar":
            let idRegex = /(?<id>\d+)\//add_to_calendar/
            let url = proposal.url.absoluteString
            if let match = url.firstMatch(of: idRegex) {
                let calendarEventId = Int(match.id)!
                Task { await addToCalendarController?.addToCalendar(calendarEventId: calendarEventId) }
            }
            return .reject
        default:
            return .accept
        }
    }
}
```

SceneController.swift

Show Event View

- Get event data from Rails app
- Launch native event view

```
class AddToCalendarController: NSObject {
    private weak var window: UIWindow?

    init(window: UIWindow?) {
        self.window = window
        super.init()
    }

    func addToCalendar(calendarEventId: Int) async {
        let url = baseUrl.appendingPathComponent("calendar_events/\(calendarEventId)")
        let viewModel = CalendarEventViewModel(url: url)

        await viewModel.fetchCalendarEvent()

        guard let calendarEvent = viewModel.calendarEvent else {
            print("Error: Could not fetch calendar event")
            return
        }

        let event = EKEvent(eventStore: CalendarEventStore.shared)
        event.calendar = CalendarEventStore.shared.defaultCalendarForNewEvents
        event.title = calendarEvent.name
        event.location = calendarEvent.location
        event.startDate = calendarEvent.startsAt
        event.endDate = calendarEvent.endsAt

        await MainActor.run {
            let eventViewController = AddToCalendarEKEEventEditViewController()
            eventViewController.calendarEventId = calendarEventId
            eventViewController.event = event
            eventViewController.eventStore = CalendarEventStore.shared
            eventViewController.editViewDelegate = self

            window?.rootViewController?.present(eventViewController, animated: true)
        }
    }
}
```

AddToCalendarController.swift

Dismiss Event View

- Event gets added to their calendar
- Dismiss event view
- Show success toast message

```
extension AddToCalendarController: EKEEventEditViewDelegate {
    func eventEditViewController(
        _ controller: EKEEventEditViewController,
        didCompleteWith action: EKEEventEditViewAction
    ) {
        controller.dismiss(animated: true)

        switch action {
        case .saved:
            if let rootViewController = window?.rootViewController {
                ToastController(rootViewController).showToast("Event added!")
            }
            print("User saved the event to their calendar")
        default:
            print("User did not save the event to their calendar")
        }
    }
}
```

AddToCalendarController.swift

Configure URL to use native component

```
class ConfigurationsController < ApplicationController
  def ios_v1
    render json: {
      rules: [
        {
          patterns: [
            "/calendar_events/[0-9]+/add_to_calendar"
          ],
          properties: {
            view_controller: "add_to_calendar"
          }
        }
      ]
    }
  end
end
```

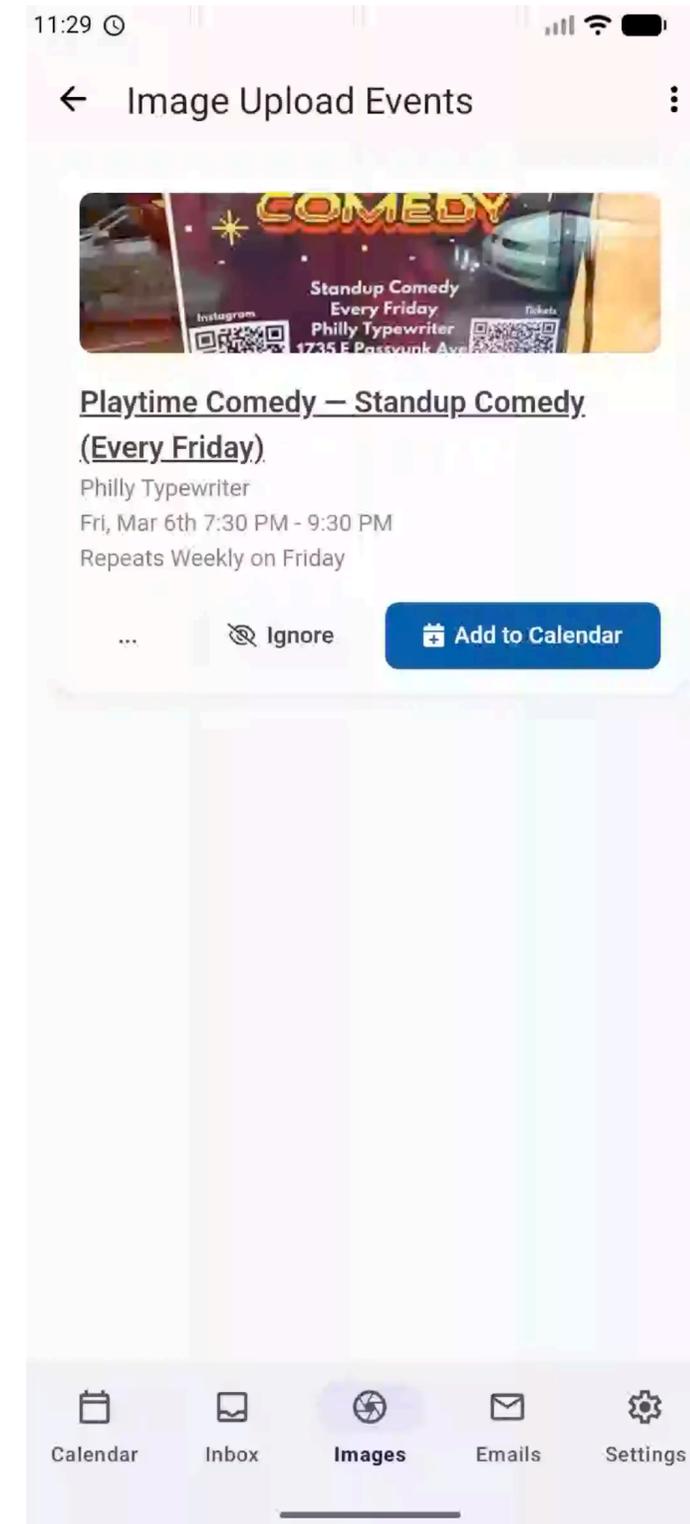
app/controllers/configurations_controller.rb

Add link to open native component

```
<%= link_to "/calendar_events/#{calendar_event.id}/add_to_calendar" do %>  
  Add to Calendar  
<% end %>
```

_add_to_calendar.html.erb

Add to Calendar Android



Route Decision Handler

Overview

- Change what happens when a URL is received
- By default
 - Internal URLs -> App
 - HTTP External URL -> System Browser
 - Non-HTTP External URLs -> Device Navigation

Route Decision Handler Interface

```
class AddToCalendarRouteDecisionHandler : Router.RouteDecisionHandler {  
    override val name = "add-to-calendar-app-navigation"  
  
    override fun matches(  
        location: String,  
        configuration: NavigatorConfiguration  
    ): Boolean {  
        ...  
    }  
  
    override fun handle(  
        location: String,  
        configuration: NavigatorConfiguration,  
        activity: HotwireActivity  
    ): Router.Decision {  
        ...  
    }  
}
```

Route Decision Handler

matches function

```
override fun matches(  
    location: String,  
    configuration: NavigatorConfiguration  
): Boolean {  
    val doesMatch = configuration.startLocation.toUri().host == location.toUri().host  
        && Regex(".*calendar_events/\\d+/add_to_calendar$").matches(location)  
    return doesMatch  
}
```

Route Decision Handler

handle function

```
override fun handle(
    location: String,
    configuration: NavigatorConfiguration,
    activity: HotwireActivity
): Router.Decision {
    val viewModel = buildCalendarEventViewModel(location)
    viewModel.fetchCalendarEvent()
    viewModel.calendarEventState.observe(activity) { calendarEvent ->
        openCalendarEventInIntent(calendarEvent, activity)
    }
    return Router.Decision.CANCEL
}

private fun buildCalendarEventViewModel(location: String): CalendarEventViewModel {
    val path = location.replace(Regex("/add_to_calendar$"), "")
    return CalendarEventViewModel(url = "${path}.json")
}

private fun openCalendarEventInIntent(
    calendarEvent: CalendarEvent,
    activity: HotwireActivity
) {
    var startsAtInMillisecondsSinceEpoch = calendarEvent.startsAtInMillisecondsSinceEpoch
    var endsAtInMillisecondsSinceEpoch = calendarEvent.endsAtInMillisecondsSinceEpoch
    var intent = Intent(Intent.ACTION_INSERT)
        .setData(CalendarContract.Events.CONTENT_URI)
        .putExtra(CalendarContract.Events.TITLE, calendarEvent.name)
        .putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, startsAtInMillisecondsSinceEpoch)
        .putExtra(CalendarContract.EXTRA_EVENT_END_TIME, endsAtInMillisecondsSinceEpoch)
        .putExtra(CalendarContract.Events.ALL_DAY, calendarEvent.allDay)
    activity.startActivity(intent)
}
```

Route Decision Handler

Registration

```
class CalendarVisionApplication : Application() {  
    override fun onCreate() {  
        super.onCreate()  
        Hotwire.registerRouteDecisionHandlers(  
            AddToCalendarRouteDecisionHandler(),  
            AppNavigationRouteDecisionHandler(),  
            BrowserTabRouteDecisionHandler(),  
            SystemNavigationRouteDecisionHandler(),  
        )  
    }  
}
```

Add to Calendar native component

- Different implementations on iOS and Android
- Same Rails code for triggering the native component

Review

Native navigation and animation

**Path configuration to change
navigation behavior**

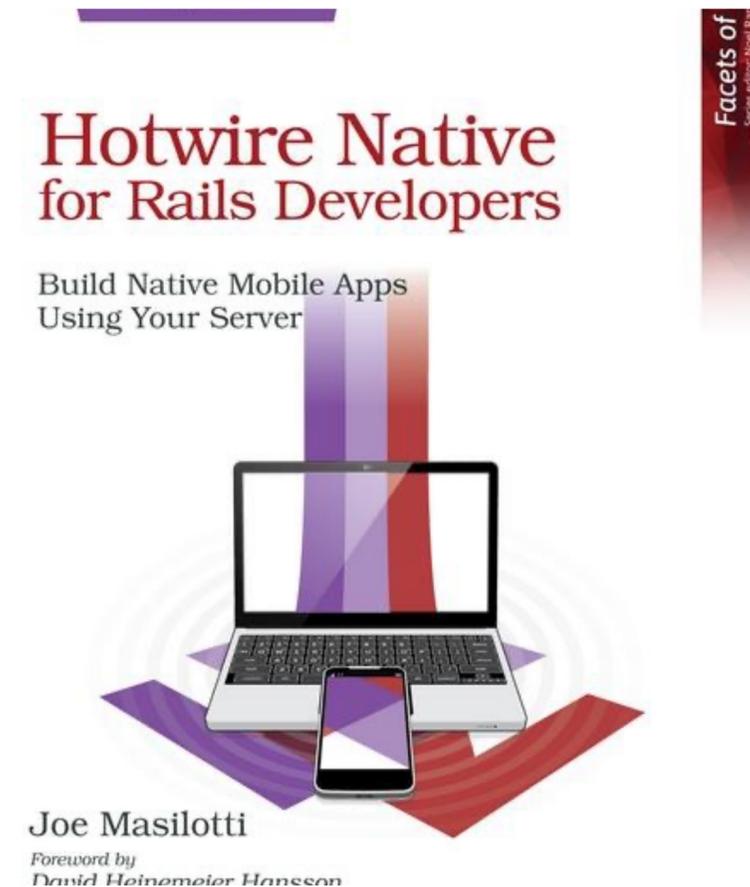
**The Native Tab Bar is an easy
native component to add**

**Bridge components when you need
a little bit of native functionality**

**Native components when bridge
components aren't enough**

Resources

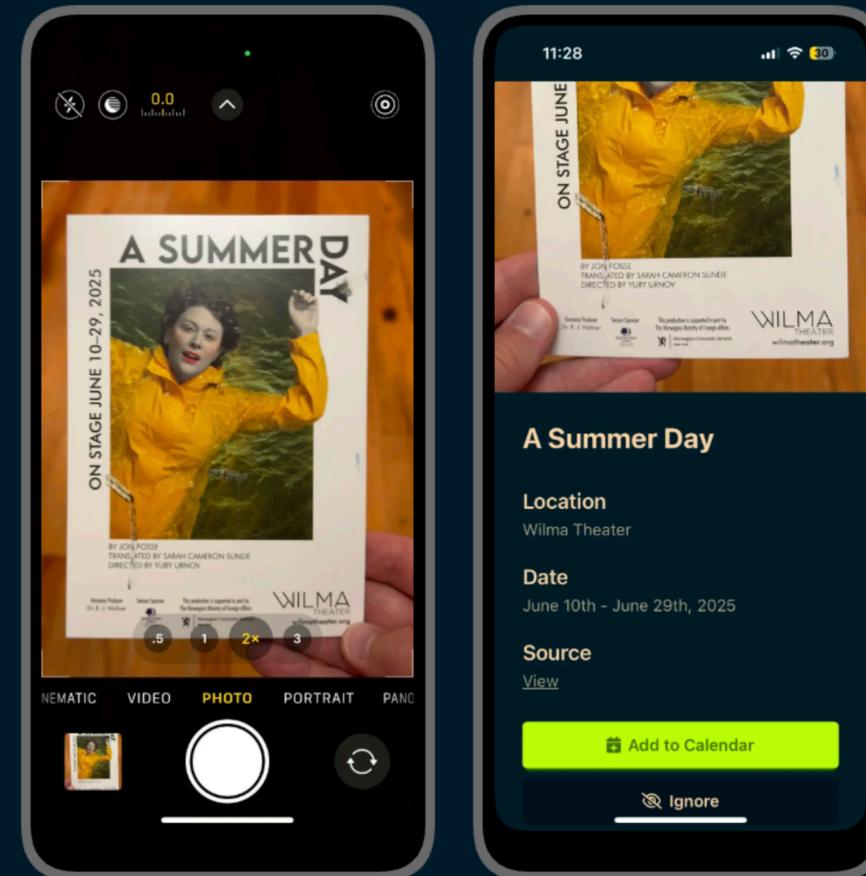
- [Hotwire Native Handbook](#) by 37signals
- [Hotwire Native for Rails Developers](#) book by Joe Masilotti
- [Learn Hotwire](#) course by Chris Oliver and William Kennedy



⚡ Learn Hotwire

Turn images into events in your calendar

Take a photo of a flyer, screenshot, or email and we'll extract the event details for you and add it to your calendar.



Extract Events From Anything



Native iOS/Android applications backed by web views

Add my app

1500cals >
Health & Fitness

Amba - Health and Wellbeing >
Health & Fitness

Baheth >
Education

Basecamp >
Project Management

BeerMenus >
Food & Drink

BlackCandy
Music

BraveCo >

Buzzsprout >

Calendar Vision >

Thanks!



<https://calendarvision.app/>



<https://mikedalton.co/socials/>