HOW TO MAKE GOOD A API FOR YOUR MOBILE APP

REST 4 MOBILE

ALEKSANDER JODŁOWSKI

- Node.js Developer
- Inspeerity Ltd.
- Go fanboy and Ruby enthusiast
- Interested in web apps security

AGENDA

- REST in general
- REST for mobile
- Rails implementation

REST IN GENERAL

WHAT IS REST?

- JSON API
- HTTP-based
- GET, POST, PUT, PATCH, DELETE
- CRUD operations
- API docs


Well yes, but actually no

WHAT IS REST, ACTUALLY?

- REpresentational State Transfer
- Architecture web design pattern
- Introduced in Roy Fielding's PhD dissertation (2000)

REST CONSTRAINTS

- Client - Server
- Stateless
- Cacheable
- Layered System
- Uniform Interface
- Code on demand

UNIFORM INTERFACE

- Identification of Resources
- Manipulation of Resources through representations
- Self-descriptive messages
- Hypermedia as the engine of state (HATEOAS)

WHAT IS HYPERMEDIA?

Hypermedia is a media, for example a text, that includes non-linear branching from one location in the media to another, via, for example, hyperlinks embedded in the media. - „Hypermedia systems“ by Carson Gross, Deniz Aksimsek and Adam Stepinski

LIKE HTML?

YES

```html
<article class="post">
  <h1>How to Ship Better Software</h1>
  <p>By Alice · Oct 5, 2025</p>
  <p>Shipping software is about feedback loops and small changes.</p>

  <section class="comments">
    <h2>Comments (2)</h2>
    <p><strong>Jordan:</strong> Great points!</p>
    <p><strong>Rina:</strong> Could you share an example?</p>
  </section>

  <form action="/posts/123/comments" method="post">
    <input name="author" placeholder="Name" required>
    <textarea name="content" placeholder="Comment..." required></textarea>
    <button type="submit">Send</button>
  </form>
</article>
```

```json
{
  "id": 123,
  "title": "How to Ship Better Software",
  "author": "Alice",
  "date": "2025-10-05",
  "content": "Shipping software is about feedback loops and small changes.
  "comments": [
    {"author": "Jordan", "text": "Great points!"},
    {"author": "Rina", "text": "Could you share an example?"}
  ]
}
```

THIS CHAPTER INTRODUCES AND ELABORATES THE REPRESENTATIONAL STATE TRANSFER (REST) ARCHITECTURAL STYLE FOR DISTRIBUTED HYPERMEDIA SYSTEMS

Roy Fielding

NATURALLY, THAT IS WHERE I HAVE TO EXPLAIN WHY "HYPERMEDIA AS THE ENGINE OF APPLICATION STATE" IS A REST CONSTRAINT. NOT AN OPTION. NOT AN IDEAL. HYPERMEDIA IS A CONSTRAINT. AS IN, YOU EITHER DO IT OR YOU AREN'T DOING REST. YOU CAN'T HAVE EVOLVABILITY IF CLIENTS HAVE THEIR CONTROLS BAKED INTO THEIR DESIGN AT DEPLOYMENT. CONTROLS HAVE TO BE LEARNED ON THE FLY. THAT'S WHAT HYPERMEDIA ENABLES

Roy Fielding

BENEFITS OF HATEOAS

- Thin, evolvable client
- Less client-server coupling
- No need for documentation - faster development
- Breaking API change does not break client
- The response just works

"I'M GOING TO CREATE A RESTFUL API..."

"USING JSON..."

"HERE ARE MY API DOCS"

REST WITH GOOD UX

REST IN MOBILE APPS

THERE ARE TWO OPTIONS

- Progressive Web Apps
- Mobile apps with embedded web view

PROGRESSIVE WEB APPS

- It’s actually a web app
- Relatively easy to make - add service-worker.js and manifest.json
- No need to place in App Store or Google Play
- You can setup offline mode via service-worker.js (even easier with Turbo Offline)
- Poor user experience - limited OS API support
- HTML is not mobile’s native UI


PROGRESSIVE
WEB APPS FOR
RAILS DEVELOPERS
EMMANUEL HAYFORD
CURACUBBY
RAILSWORLD 2024
19:40

MOBILE APPS WITH EMBEDDED WEB VIEW

- Actual mobile app
- Most of code still done on web side
- Bridge components, connecting html with native app stuff
- Access Native OS APIs
- Instant change deployments
- HTML - AGAIN!


MOBILE APPS WITHOUT THE MADNESS
RAILS WORLD 2025
SEPT 4 & 5, AMSTERDAM
JOE MASILOTTI
FOUNDER, AUTHOR
45:52

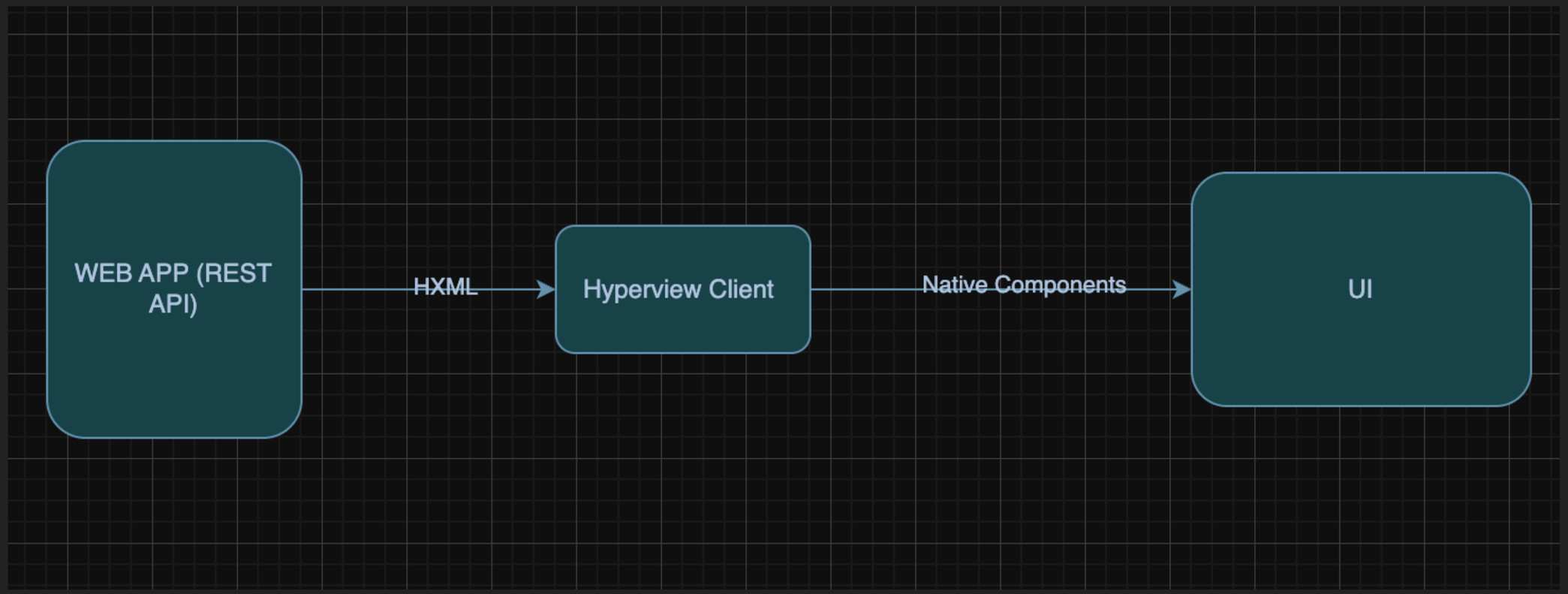
BUT ACTUALLY THERE'S A THIRD OPTION

LET'S CREATE A MOBILE HYPERMEDIA SYSTEM

OH WAIT...
IT'S ALREADY HERE

HYPERVIEW IS A SYSTEM CONSISTING OF

- Special XML data format called HXML
- React Native based client, that consumes HXML and renders native UI components.


WEB APP (REST API)
HXML
Hyperview Client
Native Components
UI

HXML

```
<!-- home.xml -->
<doc xmlns="https://hyperview.org/hyperview">
  <screen>
    <styles>
      <style id="Body" backgroundColor="white" flex="1" padding="48" />
      <style id="Label" fontSize="18" lineHeight="24" />
      <style id="Label--Link" color="blue" fontSize="18" />
    </styles>
    <body style="Body">
      <text style="Label">This is screen 1.</text>
      <text
        style="Label Label--Link"
        href="/case_studies/navigation/screen2.xml"
      >
        Click me
      </text>
    </body>
  </screen>
</doc>
```

BENEFITS AND DRAWBACKS OF HYPERVIEW

- Native App
- No web based UI components
- React Native - client can be extended with JS/TS
- Instant change deployments
- If both mobile and web - need to support both HTML and HXML

RAILS IMPLEMENTATION

DEFINE A NAVIGATOR AND ENTRYPOINT

app > views > hxml > index.xml.erb

```
1  <doc xmlns="https://hyperview.org/hyperview">
2      <%= render "hxml/styles" %>
3      <navigator id="root" type="stack">
4          <nav-route id="posts_index" href="/posts.xml" />
5      </navigator>
6  </doc>
```

EXAMPLE HXML VIEW

app > views > posts > index.xml.erb

```
<screen>
  <%= render "hxml/styles" %>
  <body style="Body">
    <header style="Header">
        <text action="push" style="text-button--primary" href="/posts/new.xml">Create Post</text>
    </header>
    <behavior trigger="refresh" action="reload" href="/posts.xml"/>
    <view style="Container">
        <text style="Text--Large" marginBottom="16">Blog Posts</text>
        <list>
            <% @posts.each do |post| %>
                <item action="push" href="<%= post_path(post, :xml) %>" style="post--item">
                    <text style="post--item--title"><%= post.title %></text>
                    <% if post.description.present? %>
                        <text style="post--item--description"><%= post.description %></text>
                    <% end %>
                    <text style="Text--Small">Created: <%= post.created_at.strftime("%B %d, %Y") %></text>
                </item>
            <% end %>
        </list>
    </view>
  </body>
</screen>
```

EXAMPLE HXML FORM

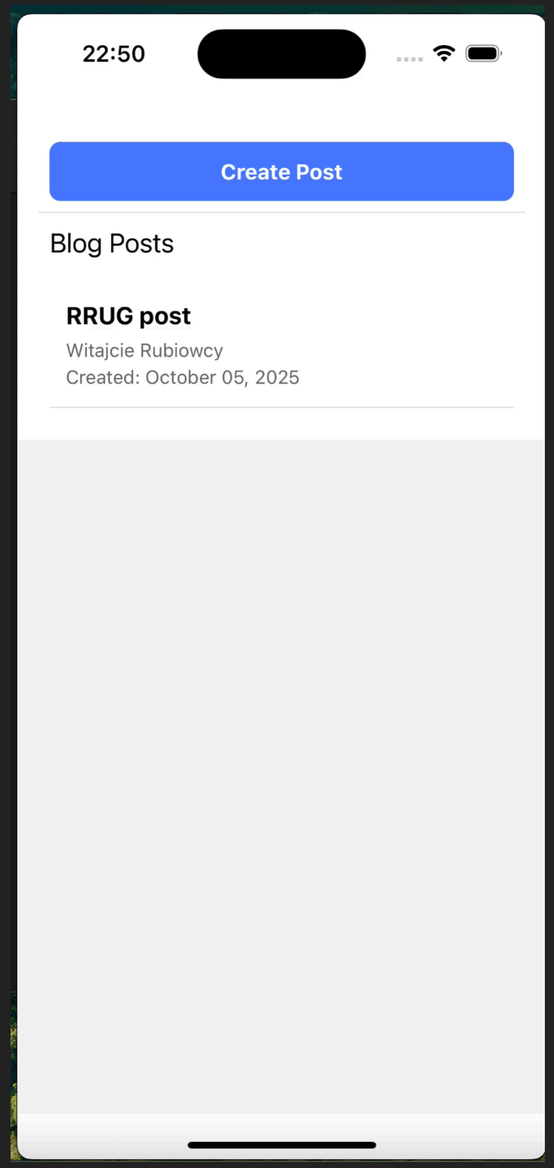
```
<form id="comments_form" scroll="true">
  <view style="form-group">
    <text style="label">Add a Comment</text>
    <text-field
      name="post_comment[content]"
      placeholder="Write your comment here..."
      placeholderTextColor="#8D9494"
      style="input" />
    <text-field
      name="authenticity_token"
      style="input"
      hide="true"
      value="<%= form_authenticity_token %>"
    />

    <text
      action="append"
      target="comments_list"
      style="text-button--primary"
      href="<%= post_comments_path(@post, :xml) %>" verb="post">
      Add Comment
    </text>
  </view>
</form>
</view>
```

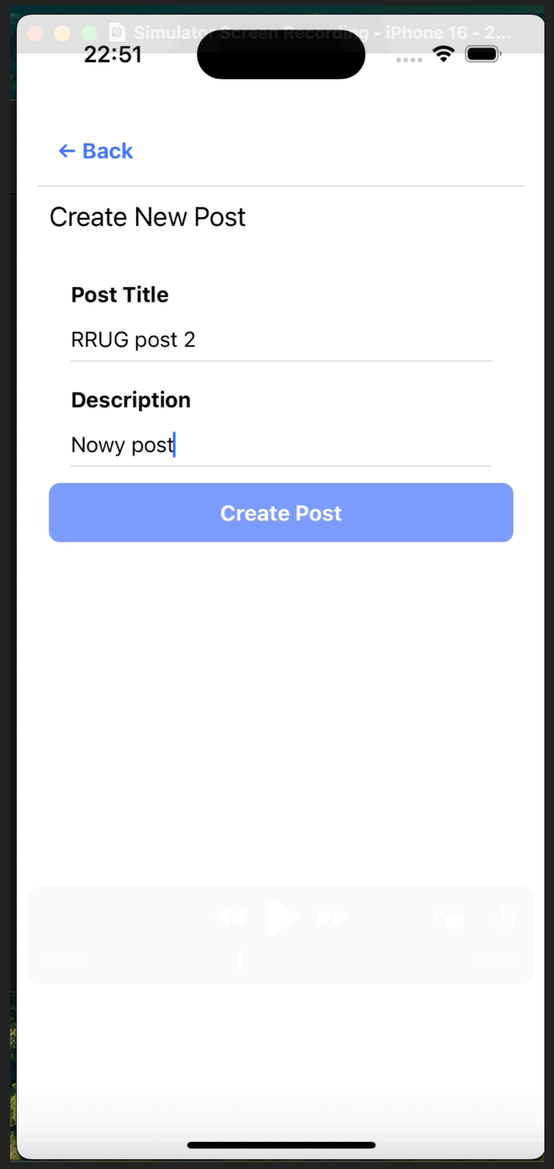
SETTING UP REACT NATIVE CLIENT

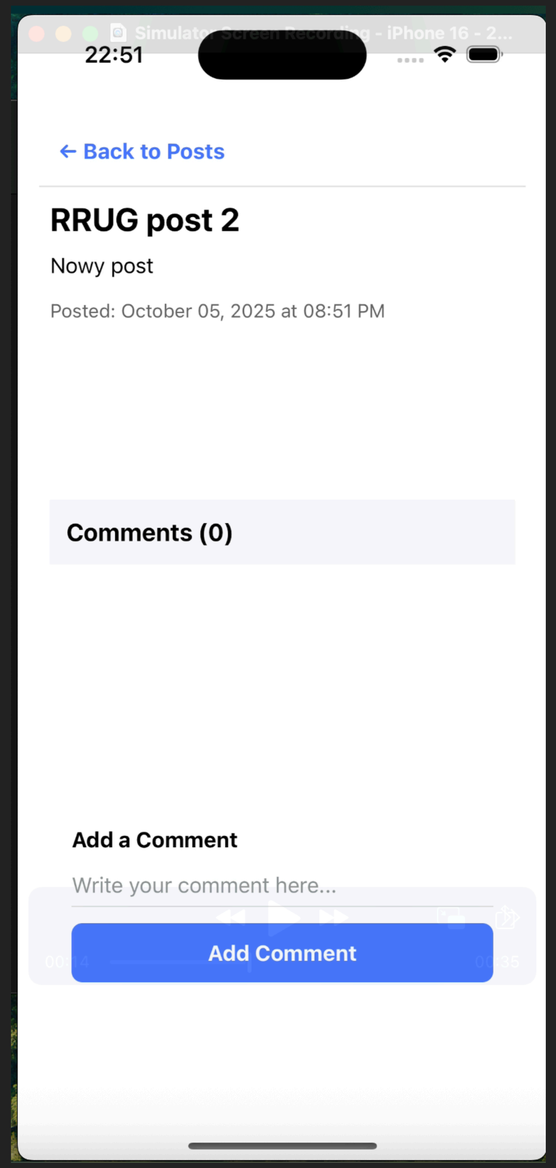
index.tsx

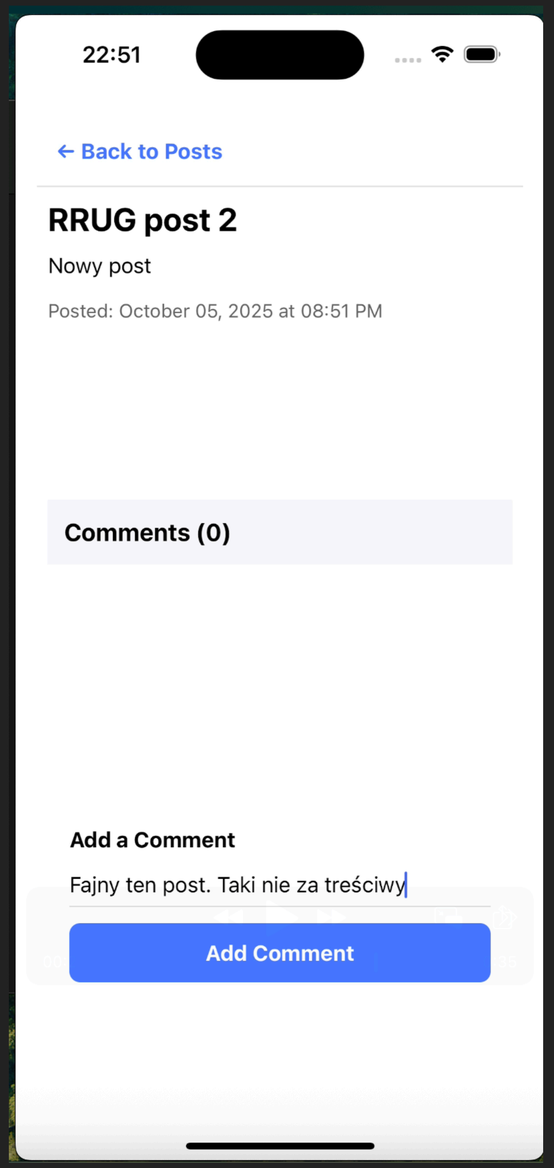
```tsx
import  Hyperview  from "hyperview";
import * as Linking from 'expo-linking';
import { SafeAreaProvider } from "react-native-safe-area-context";
import { NavigationContainer } from "@react-navigation/native";
import React from "react";

const pathPrefix = ""
const linking = {
  config: {
    screens: {
      card: {
        path: `${pathPrefix}/card`,
      },
      modal: {
        path: `${pathPrefix}/modal`,
      },
      tabs: {
        path: `${pathPrefix}/tabs`,
      },
    },
  },
  prefixes: [Linking.createURL('/')],
};

export default function Index() {
  return (
        <Hyperview
          entrypointUrl={`http://127.0.0.1:3000/hxml.xml`}
          fetch={fetch}
          formatDate={function (data) {throw new Error("Not defined")}}
        />
 );
}
```

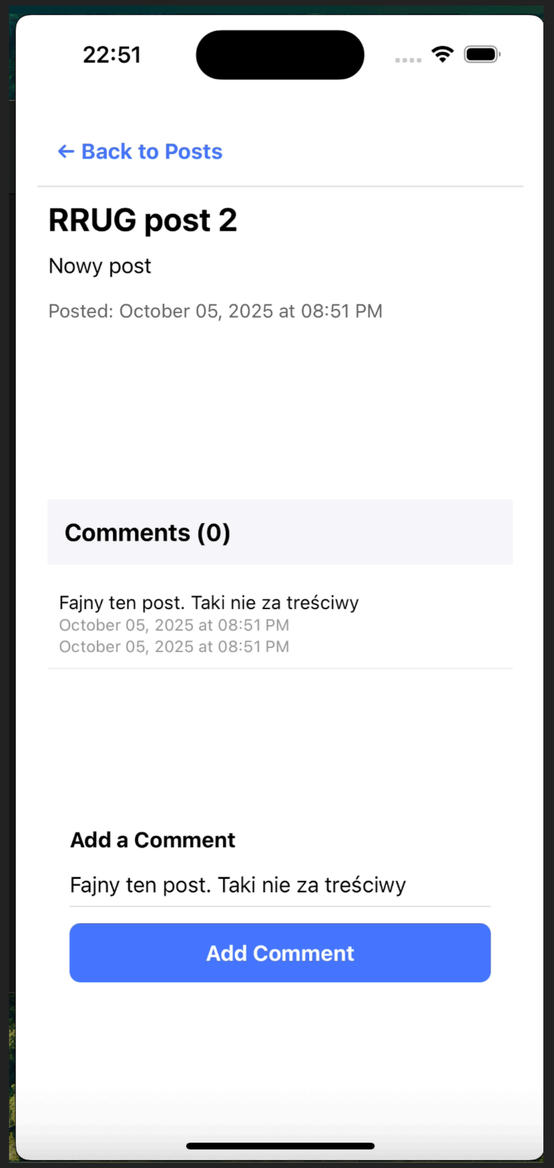
WORKING APP


22:50
Create Post
Blog Posts
RRUG post
Witajcie Rubiowcy
Created: October 05, 2025


Simulator Screen Recording - iPhone 16 - 2...
22:51
← Back
Create New Post
Post Title
RRUG post 2
Description
Nowy post
Create Post


22:51
← Back to Posts
RRUG post 2
Nowy post
Posted: October 05, 2025 at 08:51 PM
Comments (0)
Add a Comment
Write your comment here...
Add Comment


22:51
← Back to Posts
RRUG post 2
Nowy post
Posted: October 05, 2025 at 08:51 PM
Comments (0)
Add a Comment
Fajny ten post. Taki nie za treściwy
Add Comment


22:51
← Back to Posts
RRUG post 2
Nowy post
Posted: October 05, 2025 at 08:51 PM
Comments (0)
Fajny ten post. Taki nie za treściwy
October 05, 2025 at 08:51 PM
October 05, 2025 at 08:51 PM
Add a Comment
Fajny ten post. Taki nie za treściwy
Add Comment

WHAT SHOULD I PICK?

- PWA -> you want to have it quickly
- Hotwire Native -> Give an access to OS API for Rails App (or don’t know what to pick), you have already Hotwire in your rails app
- Hyperview -> true, web based mobile app
- Classical Mobile App + JSON API -> if your API is more like an addon, or mobile app is local first

SUM UP

- REST is not JSON API - you need to use hypermedia
- True REST for mobile is possible

REFERENCES

- „Hypermedia Systems” by Carson Gross, Deniz Aksimsek and Adam Stepinski (hypermedia.systems)
- „Architectural Styles and the Design of Network-based Software Architectures” by Roy Fielding
- „HATEOAS klucz do posiadania REST API” by Gabriel Ślawski
- https://hyperview.org/