
Introduction to
Web Development

Table of content

Web Development

Basic Web

The HyperText Markup Language or HTML is the standard markup language for documents designed to be **displayed** in a web browser

Cascading Style Sheets (CSS) is a stylesheet language used for **describing the presentation** (eg. color, size) of a document written in a markup language such as HTML

JavaScript is the dominant client-side scripting language of the Web, Scripts are embedded in or included from HTML documents and **interact** with the DOM (Document Object Model)

Basic Web

Display

Interaction

Styling

Web 1.0, 2.0, 3.0

Web 1.0

Read only, static website.
Eg, company website

Web 2.0

Participation, Community focus.
Eg, social media

Web 3.0

Decentralized,
Intelligent web-base
Eg, uniswap, Killswitch

Web 1.0, 2.0, 3.0

Web 1.0

HTML,CSS,
(a little bit of)
Javascript

Web 2.0

HTML, CSS,
(a lot of)
Javascript
(included javascript framework)

Web 3.0

HTML,CSS,Javascript
+ Blockchain, AI and other…

Web 1.0, 2.0, 3.0

Web 1.0

What you've learn.

Web 2.0

What you are going to learn

Web 3.0

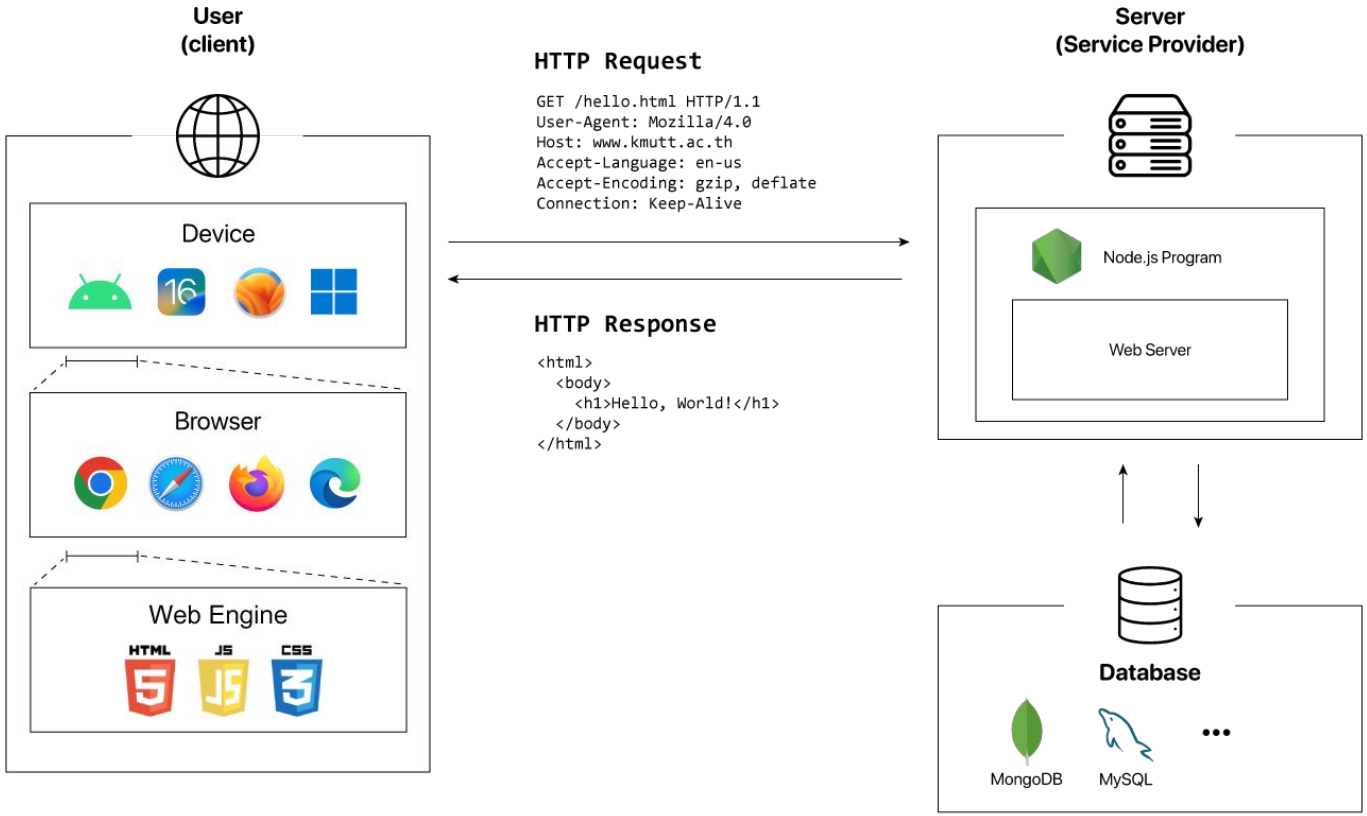
They doesn't even figure it all out yet...

Web 2.0

Today of web development

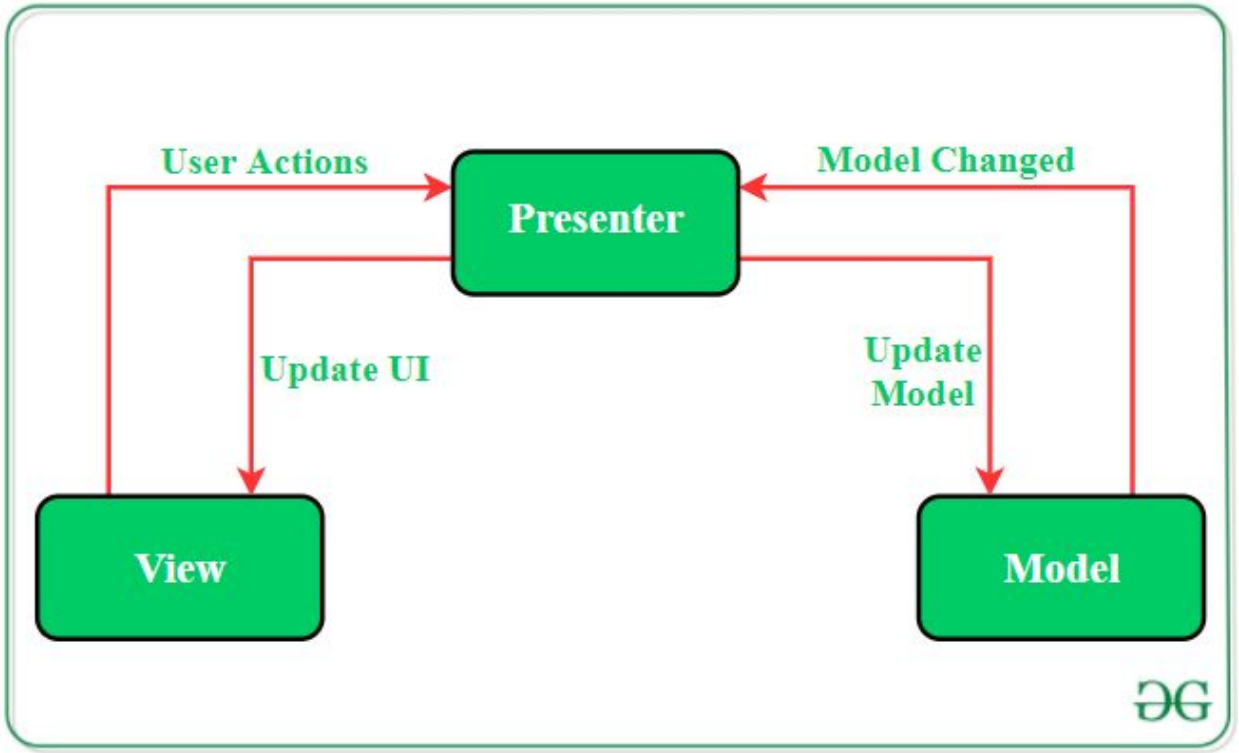
Web & Server Architecture

REST API concept, used in Facebook, Instagram, and common web applications



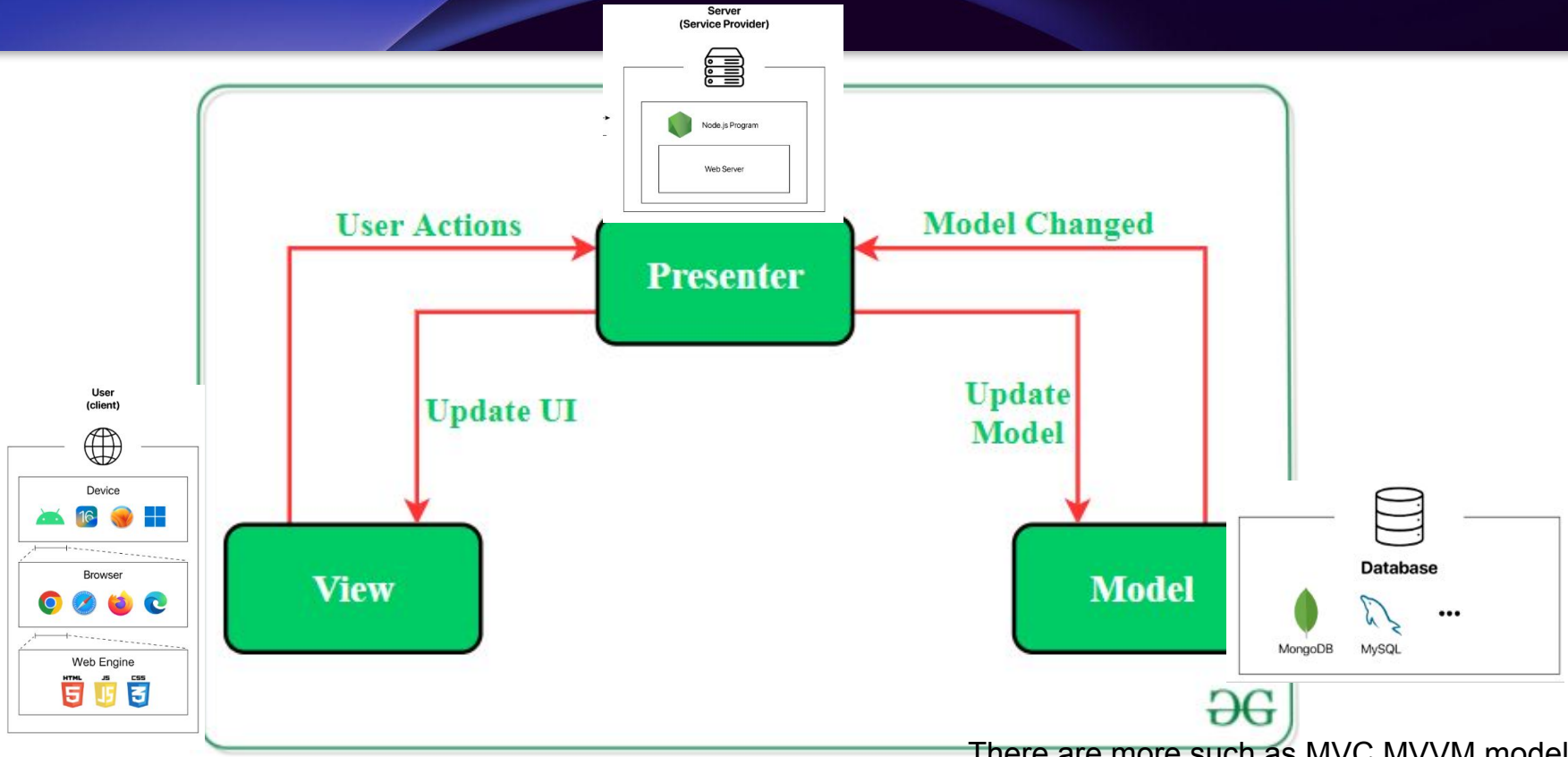
MVP

Model View Presenter Software design pattern



MVP

Model View Presenter Software design pattern



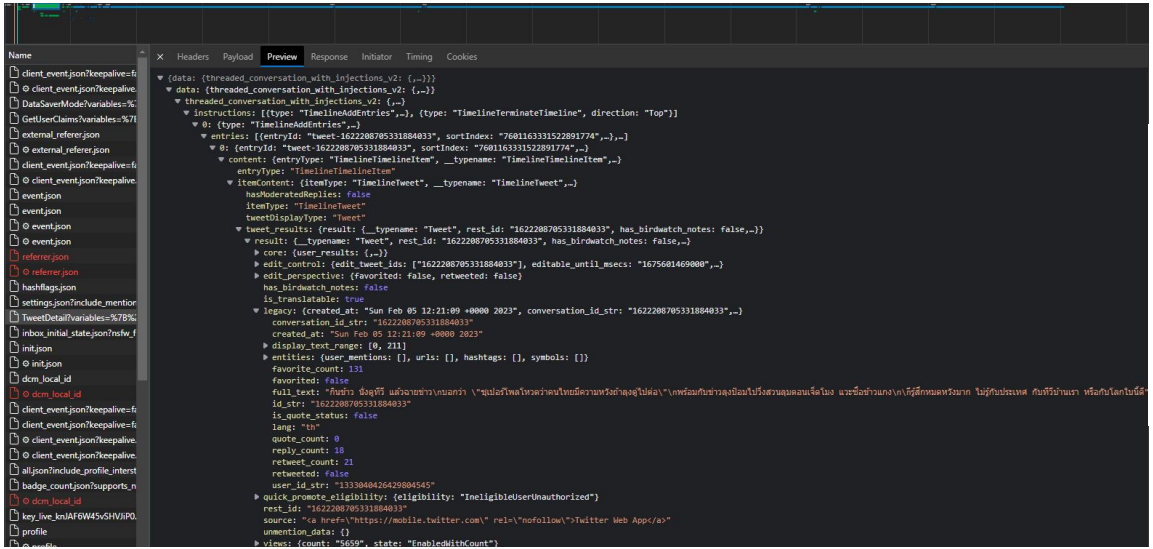
There are more such as MVC,MVVM model

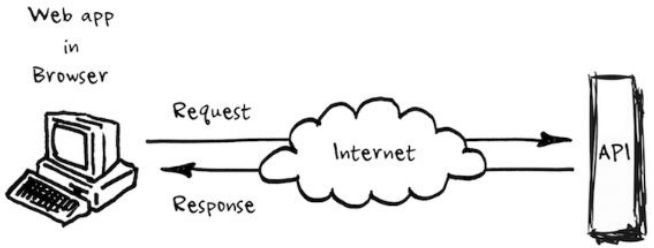
API … ?

An **application programming interface** (**API**)
is a way for two or more computer programs to communicate with each other

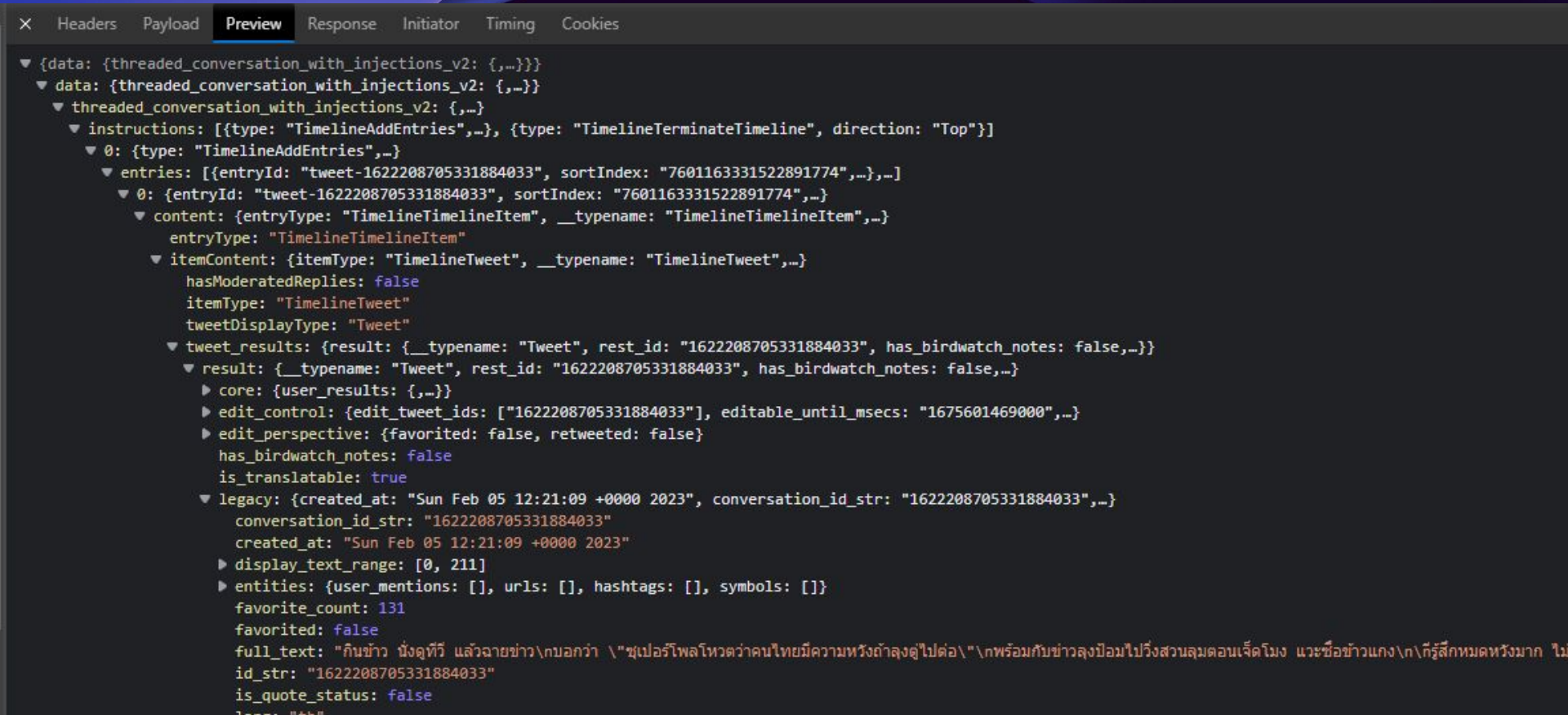
https://en.wikipedia.org/wiki/API

API Example: Twitter's Tweet Detail





API Example: Twitter's Tweet Detail




Web
Framework.

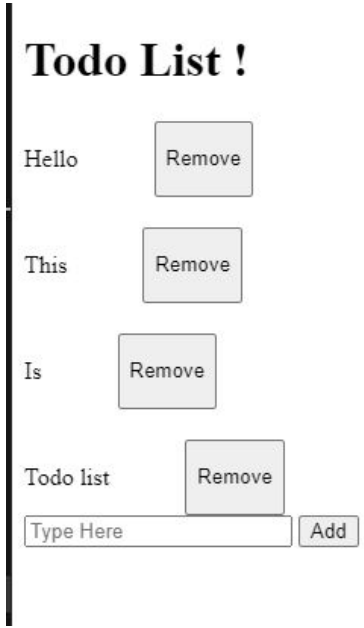
Why we don’t want to write only HTML, CSS, JS ?

Because it hard as hell.

Creating web application without library would take us too much time.

Let create the Todo app using pure HTML,CSS and JS

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Parcel Sandbox</title>
    <meta charset="UTF-8" />
  </head>

  <body>
    <div id="app">
      <h1>Todo List !</h1>
      <div
        id="list"
        style="display: flex; gap: 20px; flex-direction: column;"
      ></div>
      <form onsubmit="return handleSubmit()">
        <input placeholder="Type Here" id="input" />
        <button type="submit">Add</button>
      </form>
    </div>

    <script>
      let todos = [];

      function deleteTodo(key) {
        todos.pop(key);
        renderTodos();
      }

      function handleSubmit() {
        let input = document.getElementById("input");
        const inputValue = input.value;
        input.value = "";
        todos.push(inputValue);
        renderTodos();
        return false;
      }

      function renderTodos() {
        const listBox = document.getElementById("list");
        listBox.innerHTML = "";
        for (let todoIndex in todos) {
          const todo = todos[todoIndex];
          listBox.append(renderTodoElement(todoIndex, todo));
        }
      }

      function renderTodoElement(key, value) {
        const todoBox = document.createElement("div");
        todoBox.style = "display:flex; gap:50px;";
        todoBox.innerHTML = `
                  <p>${value}</p>
                  <button onclick="deleteTodo(${key})"> Remove </button>
                  `;
        return todoBox;
      }
    </script>
  </body>
</html>
```



https://codesandbox.io/s/simple-todolist-in-vanila-qd4h4c

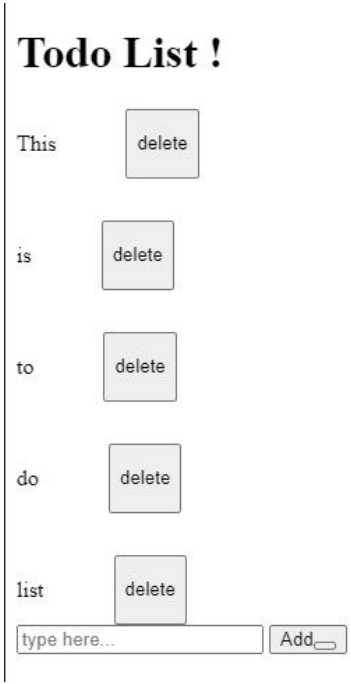
Let create the Todo app using React

```jsx
import "./styles.css";
import { useRef, useState } from "react";

export default function App() {
  const [todos, setTodos] = useState([]);
  const inputRef = useRef();
  const onSubmit = (e) => {
    e.preventDefault();
    const inputText = inputRef.current.value;
    setTodos((d) => [...d, inputText]);
    inputRef.current.value = "";
  };
  const deleteTodo = (key) => {
    setTodos((v) => v.filter((todo, index) => index !== key));
  };
  return (
    <div className="App">
      <h1> Todo List ! </h1>

      <div
        style={{
          display: "flex",gap: 5,flexDirection: "column",alignItems: "center"
        }}
      >
        {todos.map((todo, key) => (
          <div key={key} style={{ display: "flex", gap: 10 }}>
            <p>{todo}</p>
            <button onClick={() => deleteTodo(key)}>delete</button>
          </div>
        ))}
        <form onSubmit={onSubmit}>
          <input placeholder="type here..." ref={inputRef} />
          <button>Add</button>
        </form>
      </div>
    </div>
  );
}
```

https://codesandbox.io/s/todo-react-e7jwc7

Let create the Todo app using Svelte

```
<script>
  let todos = [];
  let todoText = "";
  const addTodos = () => {
    todos = [...todos, todoText];
    todoText = "";
  };

  const deleteTodo = key => {
    todos = todos.filter((data, index) => index !== key);
  };
</script>

<main>
  <h1>Todo List ! </h1>
  <div style="display:flex;gap:30px;flex-direction:column" >
  {#each todos as todo,key}
  <div style="display:flex;gap:50px">
  <p>{todo}</p>
  <button on:click={()=>deleteTodo(key)}>delete</button>
  </div>
  {/each}
  </div>
  <form on:submit|preventDefault={addTodos}>
  <input bind:value={todoText} placeholder="type here..." />
  <button type="submit" >Add<button>
  </form>
</main>
```



https://codesandbox.io/s/todo-svelte-jcljln

Advantage of using web framework.

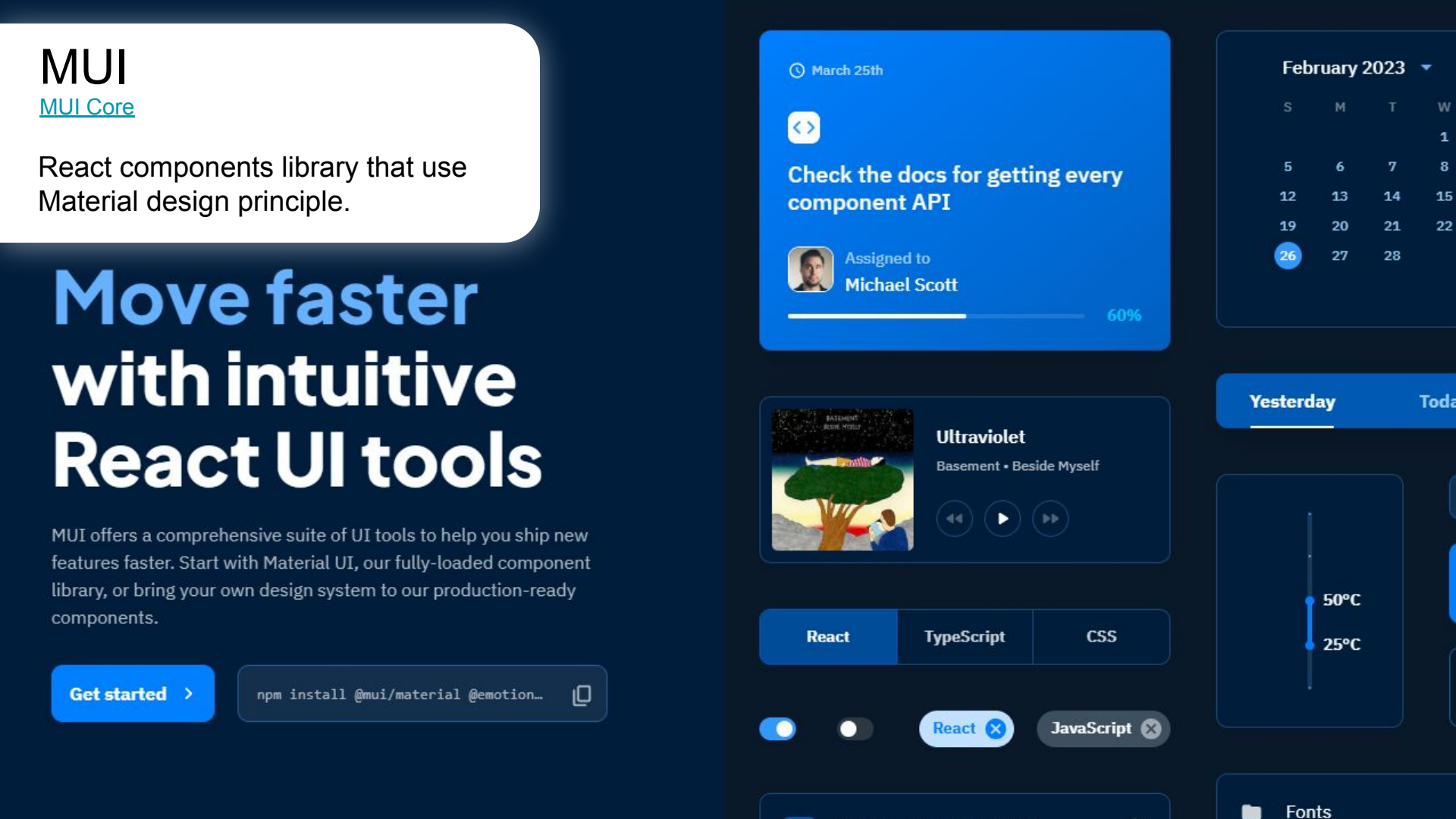
- Easy to learn.
- Easy to set up.
- There are a lot of libraries to use.

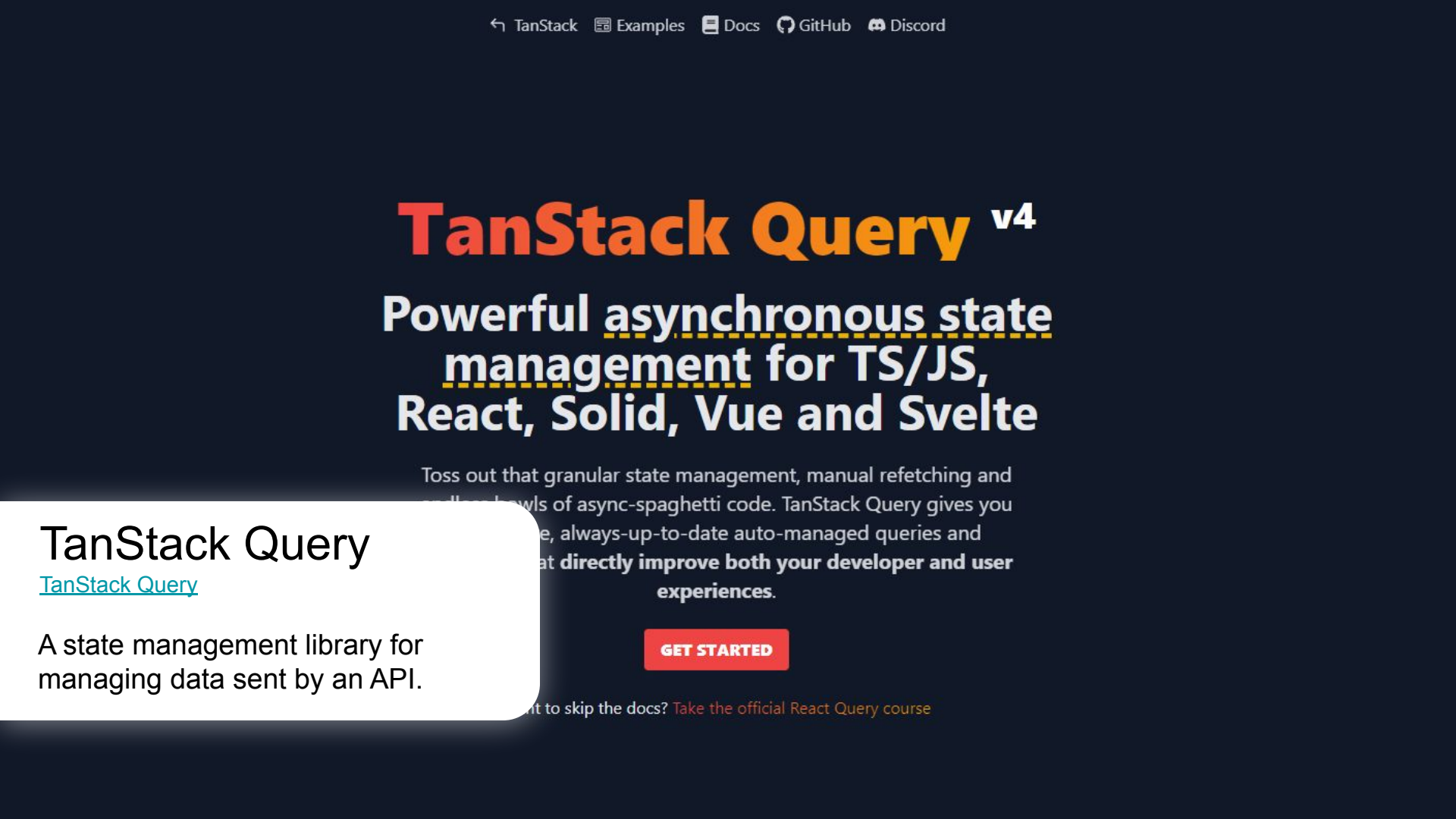
Useful Frontend Library

MUI

MUI Core

React components library that use Material design principle.

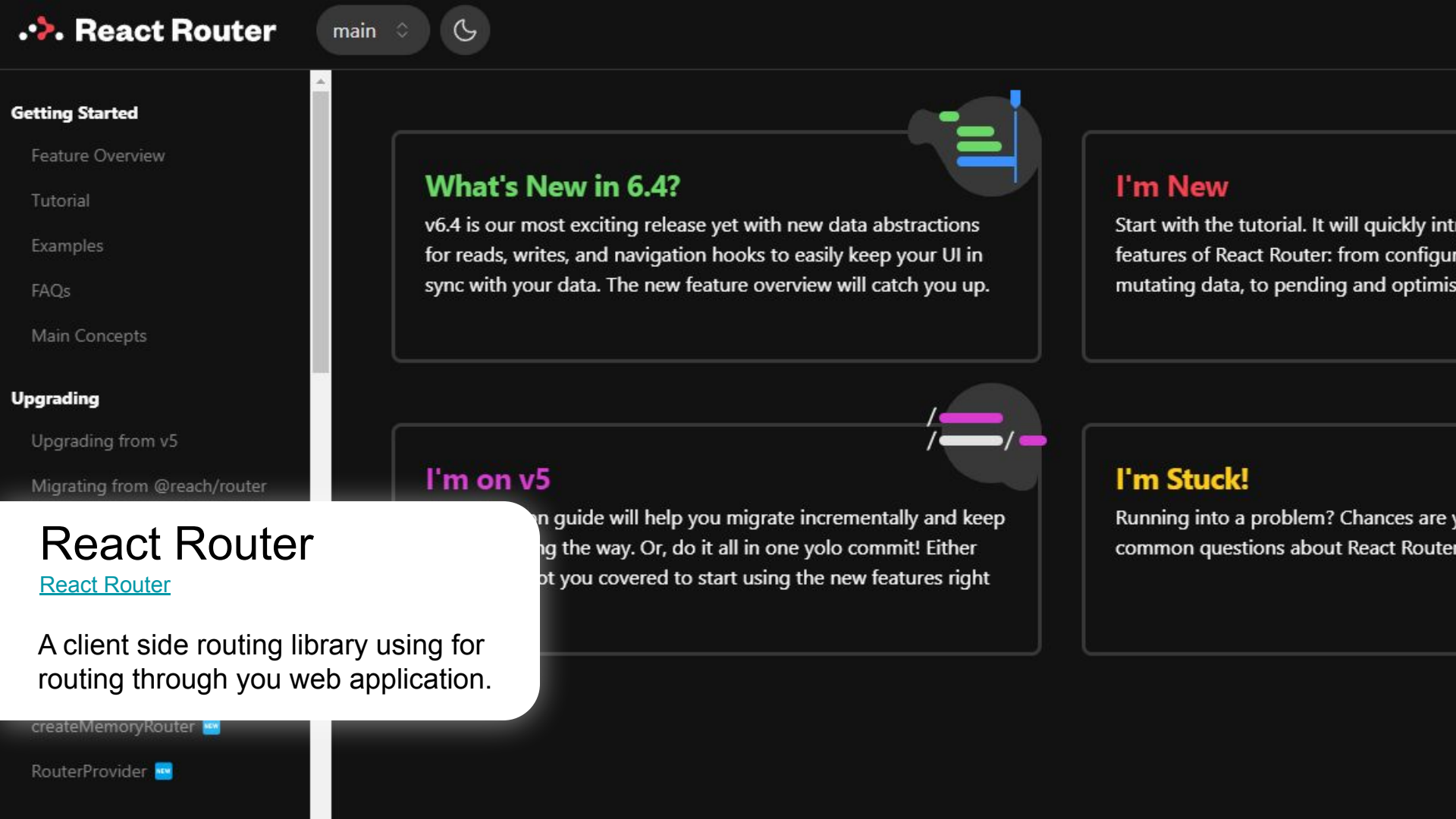




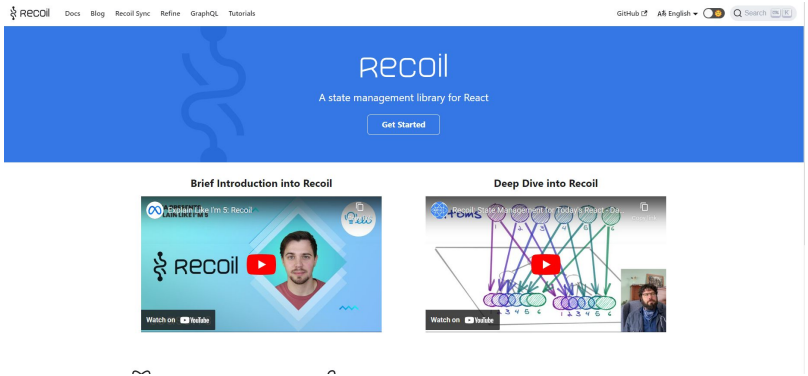
TanStack Query

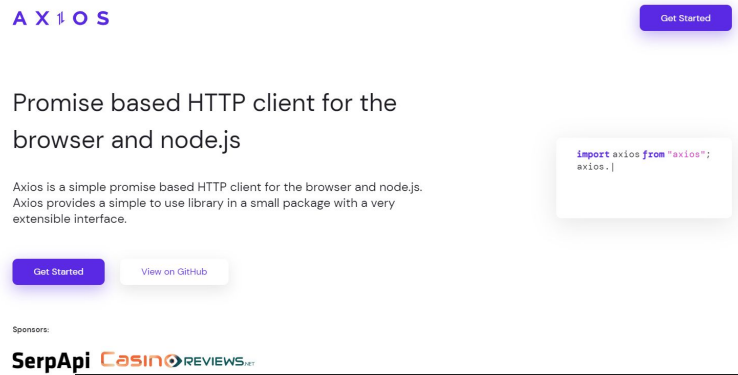
TanStack Query

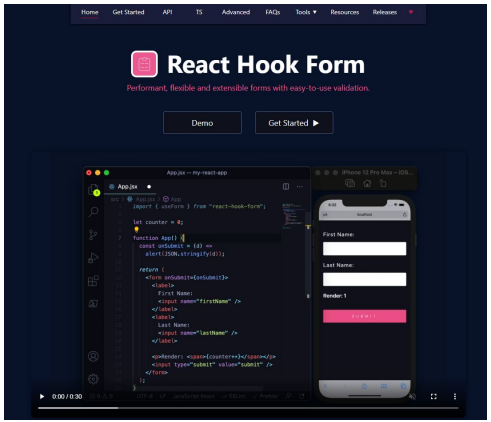
A state management library for managing data sent by an API.

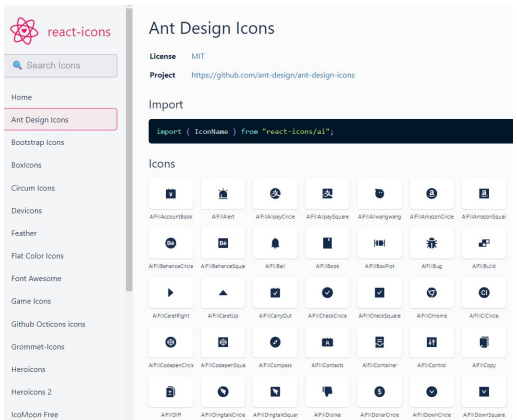

React Router
main
Getting Started
Feature Overview
Tutorial
Examples
FAQs
Main Concepts
Upgrading
Upgrading from v5
Migrating from @reach/router
createMemoryRouter
RouterProvider
What's New in 6.4?
v6.4 is our most exciting release yet with new data abstractions for reads, writes, and navigation hooks to easily keep your UI in sync with your data. The new feature overview will catch you up.
I'm New
Start with the tutorial. It will quickly int
features of React Router: from configu
mutating data, to pending and optimis
I'm on v5
n guide will help you migrate incrementally and keep
g the way. Or, do it all in one yolo commit! Either
ot you covered to start using the new features right
I'm Stuck!
Running into a problem? Chances are
common questions about React Route
React Router
React Router
A client side routing library using for routing through you web application.

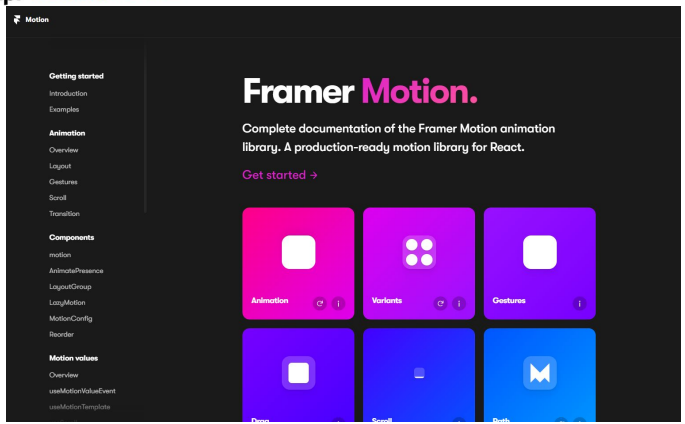
Other Libraries you might be interested.



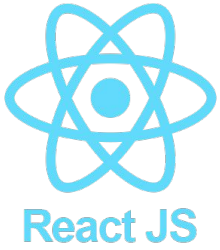






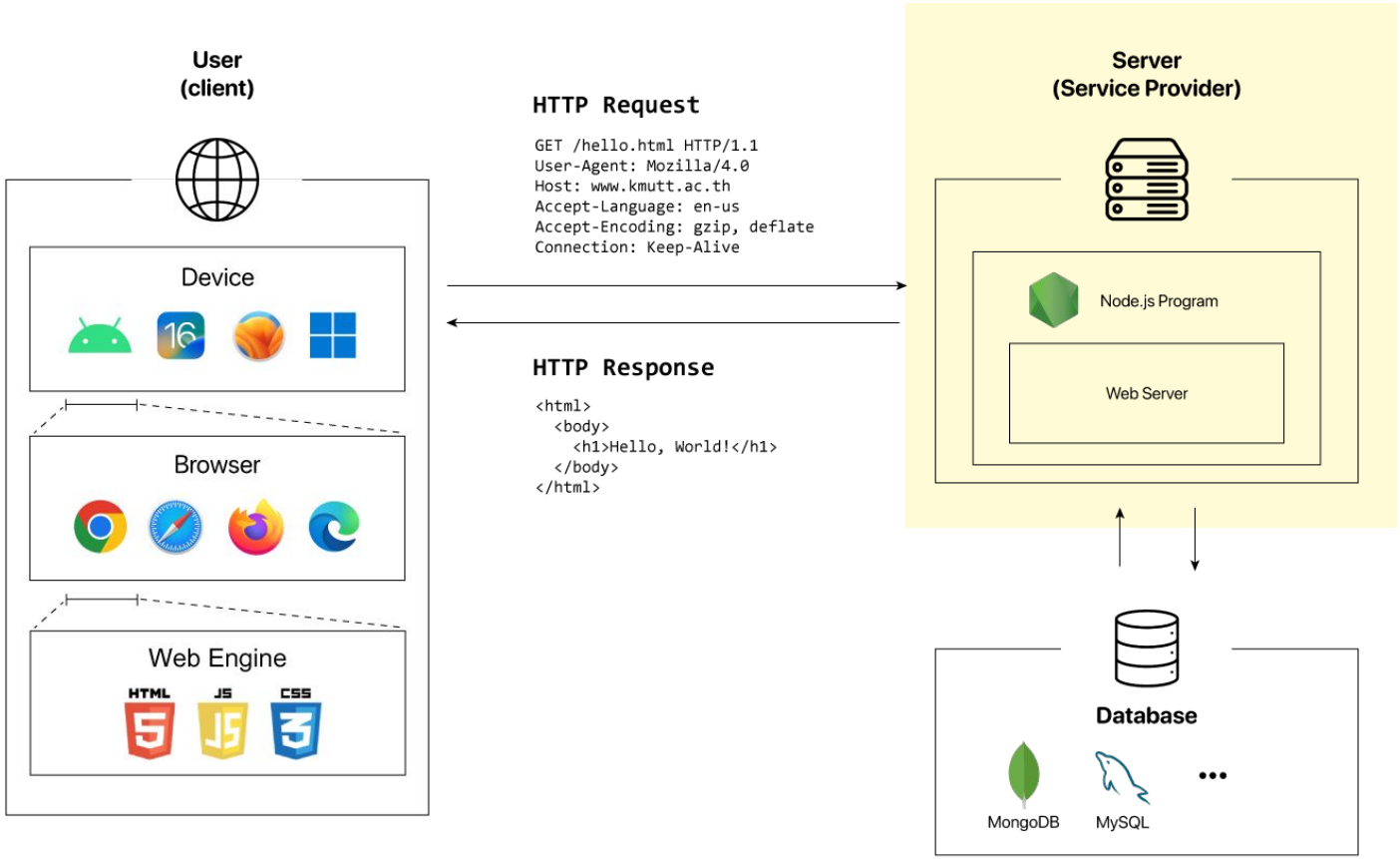


Here is some of web framework (Frontend) you can take some look.



Each web framework has their own style and community, so you can choose what you like.

Backend of web


User
(client)
Device
Browser
Web Engine
HTML
JS
CSS
HTTP Request
GET /hello.html HTTP/1.1
User-Agent: Mozilla/4.0
Host: www.kmutt.ac.th
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
HTTP Response
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
Server
(Service Provider)
Node.js Program
Web Server
Database
MongoDB
MySQL

What does backend do ?

- Receive, process and response data for client to use.
- Communicate to database and make CRUD (create, read, update, delete) operation.
- Process user credential for authentication and authorization.

How can I write a backend ?

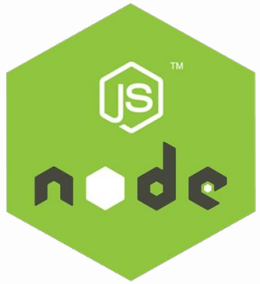
There are a lot of **tools**, **languages** and **libraries** you can choose.

Here is what you might want to learn:

mongoose

elegant mongodb object modeling for node.js

Express



Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine.

Basically, The thing that make your Javascript can be run outside browser just like Java or python.

We use Node.js to create a server to listen requests from client.

Express.js

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

Basically, A library for nodejs to provide simpler server development in Node.js.

We use Express to create server, provide middleware or even do a server side rendering.

Mongoose

Mongoose is a JavaScript object-oriented programming library that creates a connection between MongoDB and the Node.js JavaScript runtime environment

We use mongoose to communicate with database (Mongodb). It provided type structure that help you developing.

So, here is some question that you might need to ask to yourself first.

- What kind of **client** are we communicating with ?
- What kind of **Api** are we building ?
- What kind of **database** are we using ?

But that's a topic of the backend session…..

But backend might be a little complicated to virtualized.

So why don't we see it from the real project?

https://codesandbox.io/p/sandbox/eloquent-frost-s2e5vx

package.json README.md **index.js** Sandbox Info

```
1  import express from "express";
2
3  const app = express();
4
5  app.get("/", (req, res) => {
6    return res.send("Hello This is our webpage !");
7  });
8
9  app.get("/cat", (req, res) => {
10   return res.send({ name: "Meow", breed: "scottish fold", age: 35 });
11 });
12
13 app.post("/cat", (req, res) => {
14   return res.send({
15     name: "Post Meow",
16     description: "You won't be able to access this on normal webpage",
17     breed: "scottish fold",
18     age: 35,
19   });
20 });
21
22 app.listen(3000, () => {
23   console.log("This app is listening on port 3000 !");
24 });
25
```

Look outside Javascript.

So…. you don’t like Javascript but you want to be a web developer ?

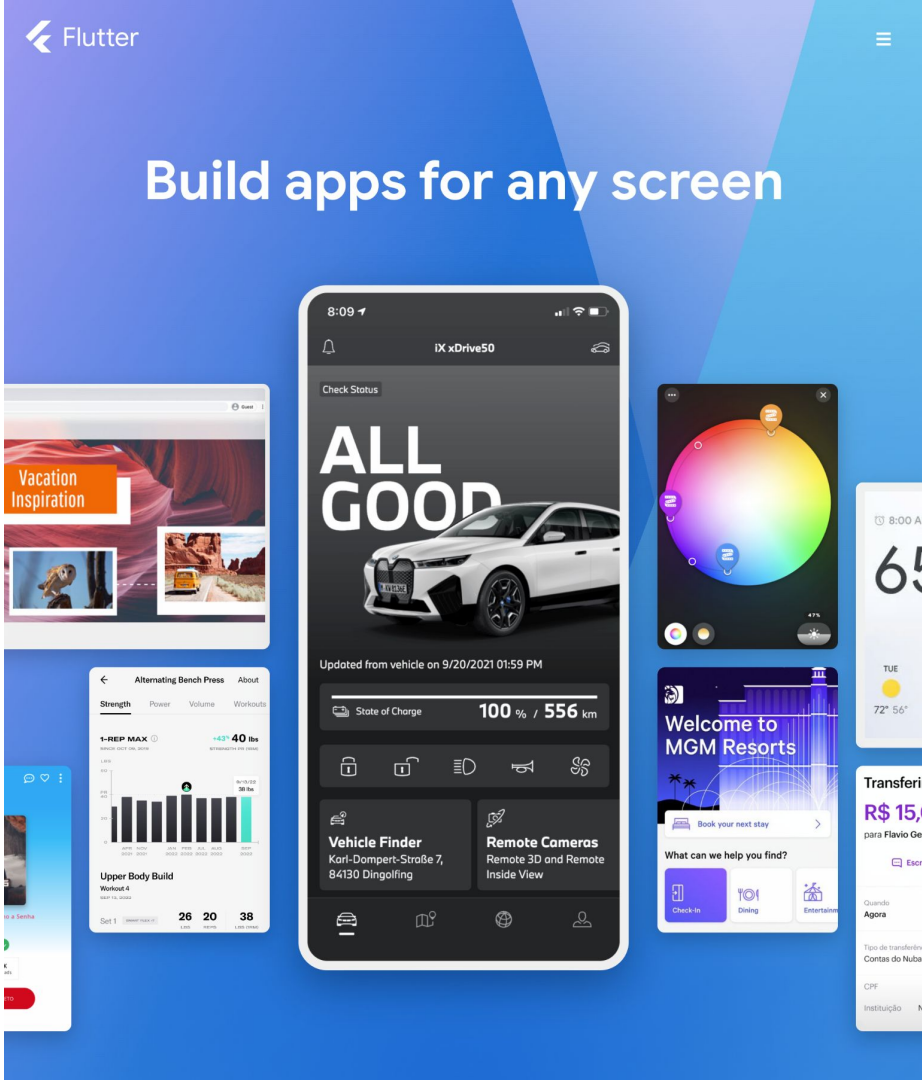
Don’t worry, we can develop website without using any of Javascript !


Frontend without Javascript.

Flutter - Frontend Alternative

Flutter use Dart instead of HTML,CSS and Javascript. If you write your web application in Flutter, you can convert it into Mobile Application or a computer program!

[Flutter - Build apps for any screen]



WebAssembly

Alternative of Javascript that’s even faster than Javascript !

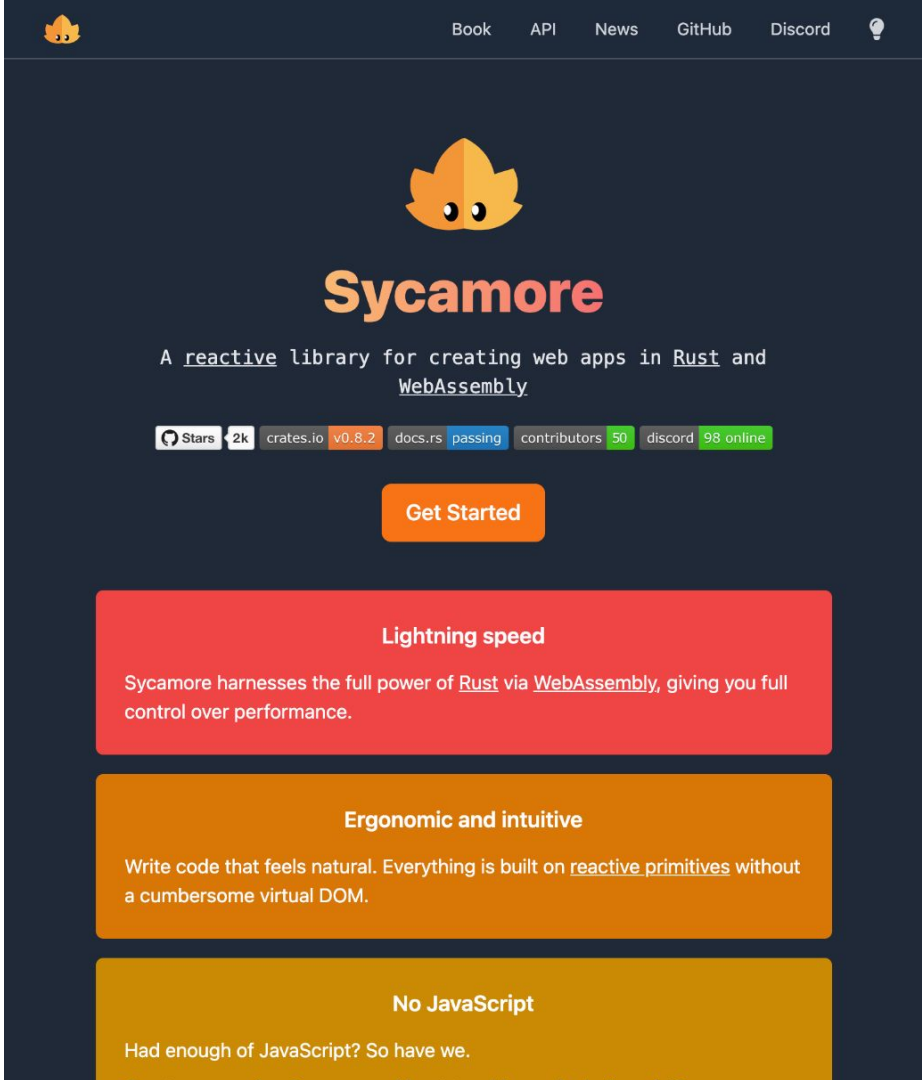
Web assembly allows you to run binary instruction in web browser ! you can write an application in Rust, Python,Go and other languages then run it (almost) natively in browser.

WebAssembly

Sycamore - Writing web application in Rust !

Sycamore is Rust library that let you write webAssembly instead of javascript.

[Sycamore (sycamore-rs.netlify.app)]

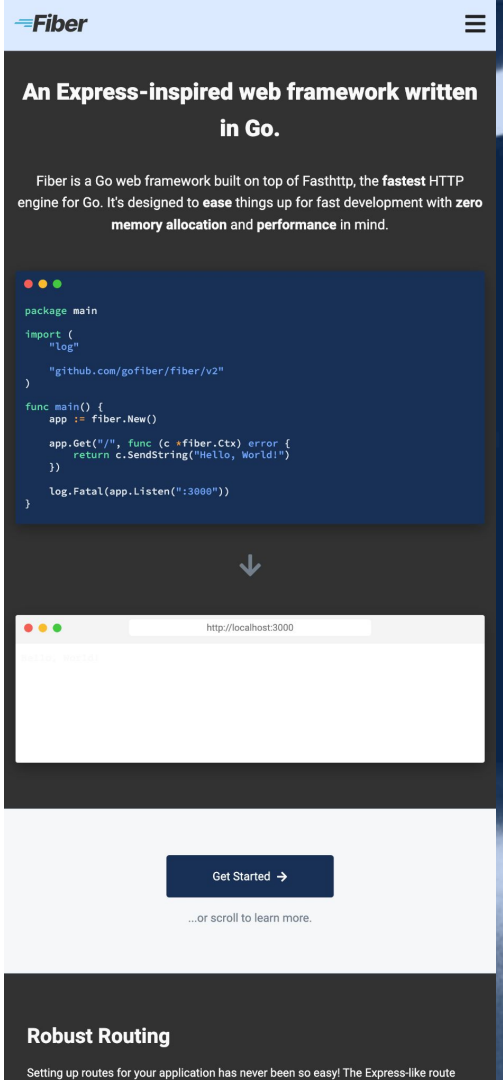



Backend without Javascript.

Gin, Fiber: Go's alternative web framework.

Gin and Fiber is a web framework that let you write Go instead of Javascript in Node.js. These two are very popular in modern web development and provide more performance than Nodejs!

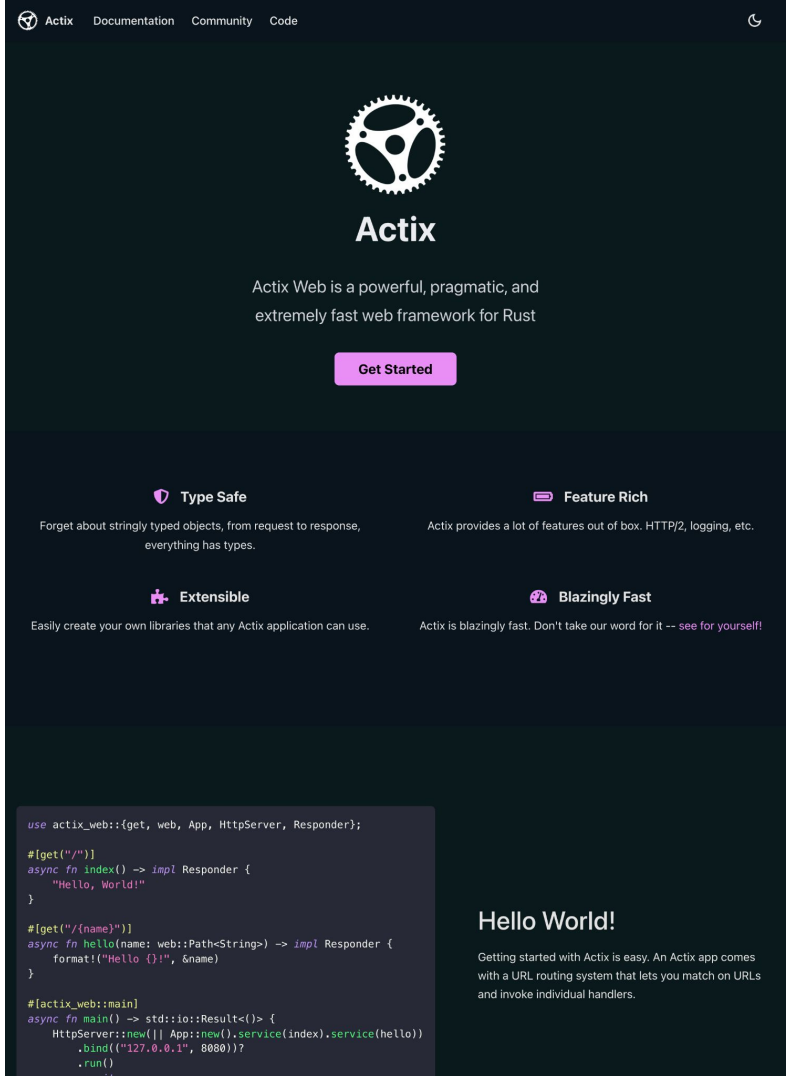
[Gin Web Framework (gin-gonic.com)]
[Fiber (gofiber.io)]



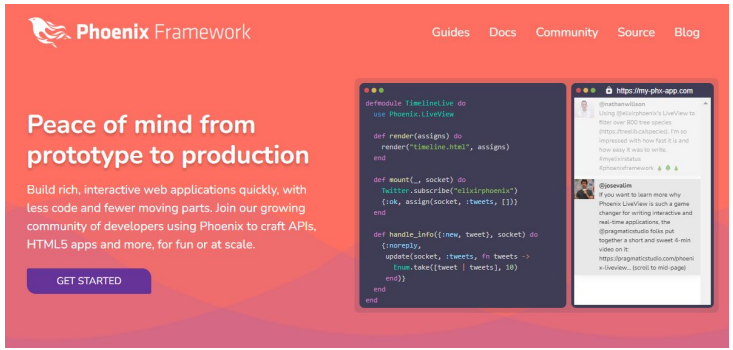
Rust - Actix a web framework in Rust

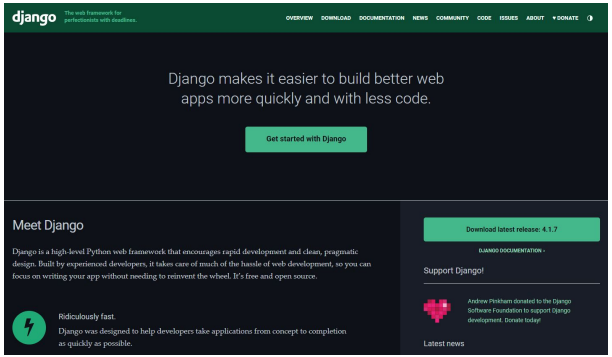
Gin and Fiber is a web framework that let you write Rust instead of Javascript in Nodejs. It even give more performance than Go !.

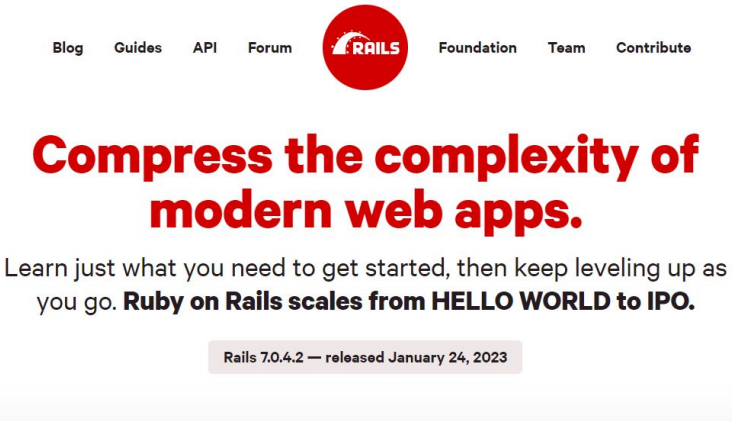
[Actix]

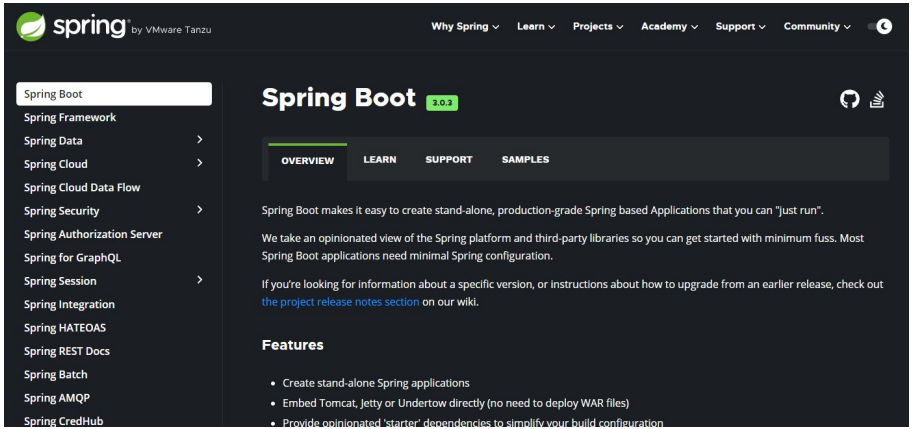


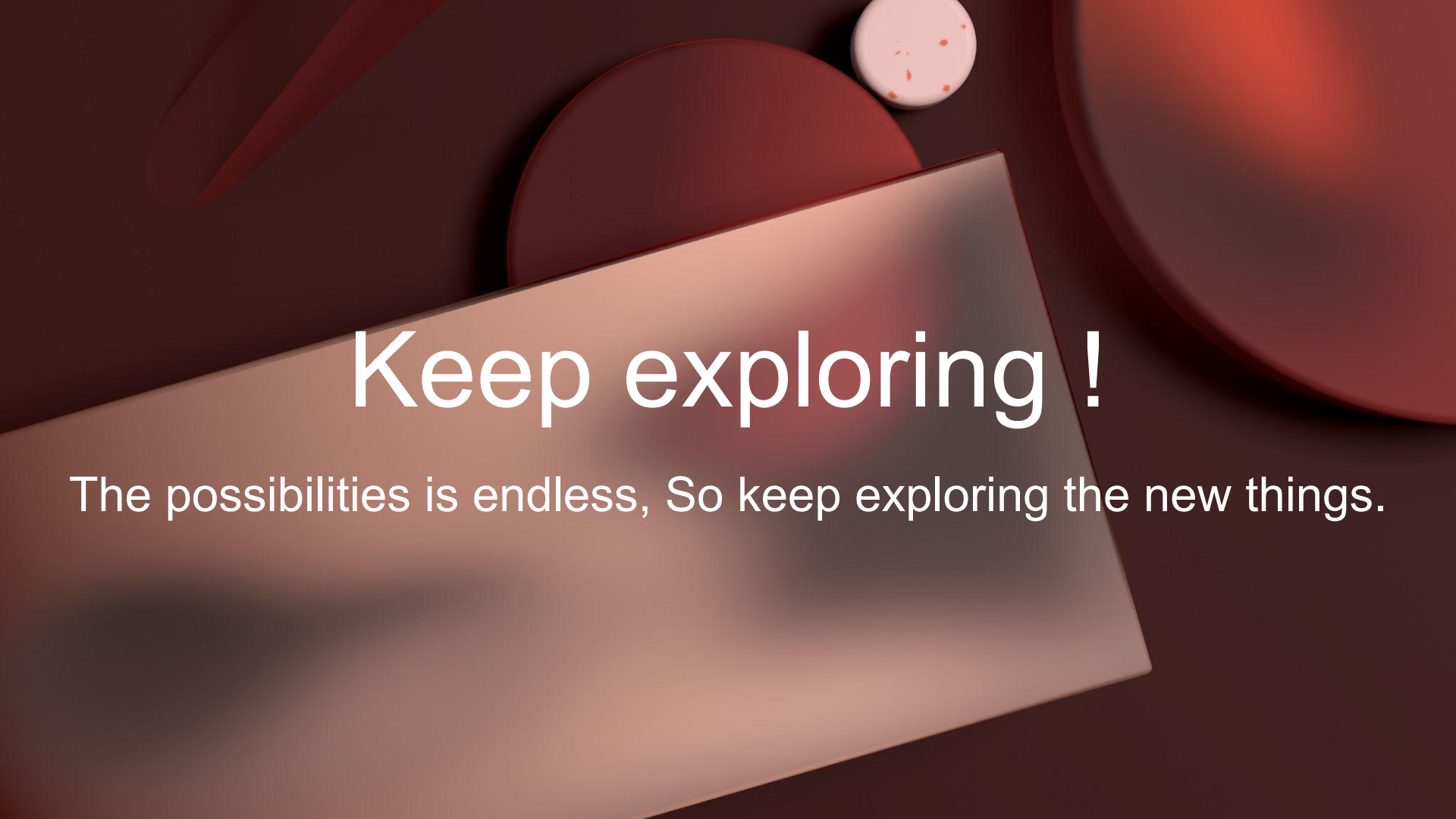
And Other….


Phoenix Framework
Guides
Docs
Community
Source
Blog
Peace of mind from prototype to production
Build rich, interactive web applications quickly, with less code and fewer moving parts. Join our growing community of developers using Phoenix to craft APIs, HTML5 apps and more, for fun or at scale.
GET STARTED


django
Django makes it easier to build better web apps more quickly and with less code.
Get started with Django
Meet Django
Ridiculously fast.
Support Django!
Latest news


Blog
Guides
API
Forum
RAILS
Foundation
Team
Contribute
Compress the complexity of modern web apps.
Learn just what you need to get started, then keep leveling up as you go. Ruby on Rails scales from HELLO WORLD to IPO.
Rails 7.0.4.2 — released January 24, 2023


spring by VMware Tanzu
Why Spring
Learn
Projects
Academy
Support
Community
Spring Boot
Spring Framework
Spring Data
Spring Cloud
Spring Cloud Data Flow
Spring Security
Spring Authorization Server
Spring for GraphQL
Spring Session
Spring Integration
Spring HATEOAS
Spring REST Docs
Spring Batch
Spring AMQP
Spring CredHub
Spring Boot 3.0.3
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
Create stand-alone Spring applications
Embed Tomcat, Jetty or Undertow directly (no need to deploy WAR files)
Provide opinionated 'starter' dependencies to simplify your build configuration


Keep exploring !
The possibilities is endless, So keep exploring the new things.