ABOUT APIs:

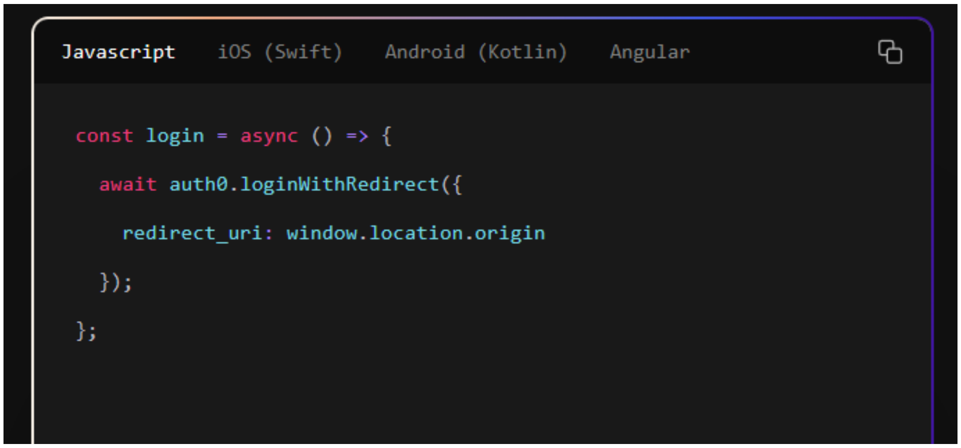
APIs, or Application Programming Interfaces, are a way for two pieces of software to communicate with each other. They allow developers to access and use data and functionality from other applications without having to build it themselves. This can save a lot of time and effort, and it can also help to create more innovative and user-friendly applications.

There are many different types of APIs, but they all share some common features. APIs typically have a set of endpoints, which are URLs that can be used to access the API's resources. Each endpoint has a corresponding set of methods, which are the actions that can be performed on the resource. For example, an API endpoint for a weather forecast might have a method for getting the current temperature in a particular city.

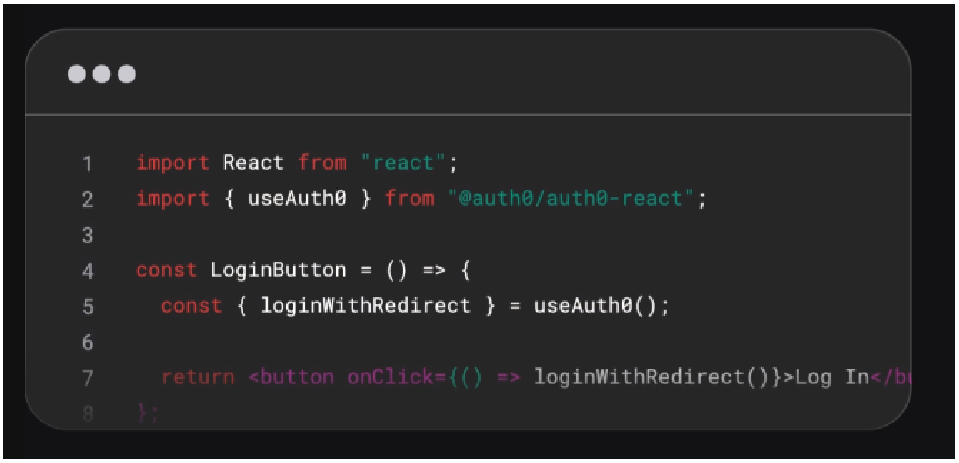
To use an API, a developer must first obtain an API key from the API provider. This key is used to authenticate the developer and to authorize them to access the API. Once the developer has an API key, they can use it to make requests to the API's endpoints. The API will then respond to the request with the requested data or functionality.

WHAT APIs TO USE:

- React Todo example App with secure NoSQL API using Auth0.com - Your Quick Backend Solution (codehooks.io)
    - example of building a to-do list (which we will want to implement using user authentication, back-end API w/ database, and react frontend interaction
    - super beneficial moving forward and hopefully helping us to connect all pieces together
- Recommends Auth0.com for user authentication which can be easily added into .js file using:



```
const login = async () => {
  await auth0.loginWithRedirect({
    redirect_uri: window.location.origin
  });
};
```

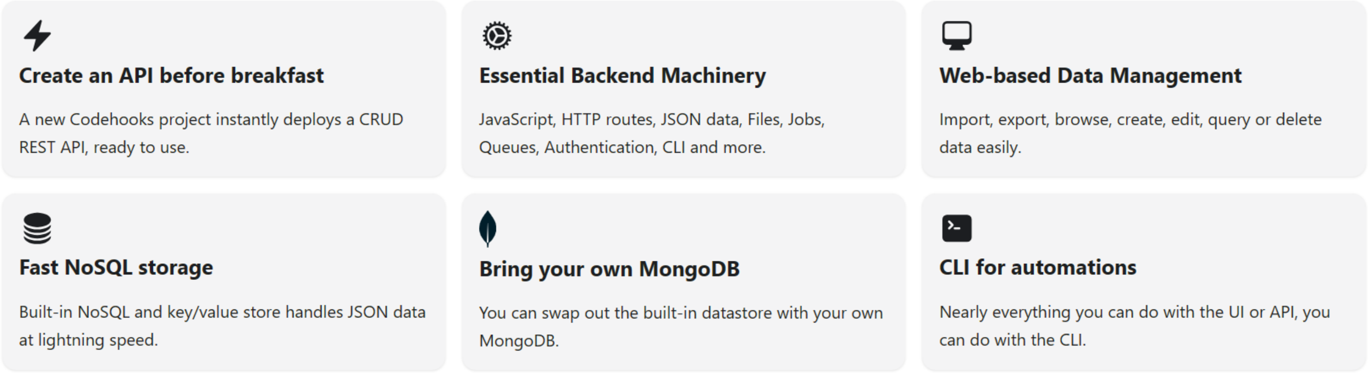


```
import React from "react";
import { useAuth0 } from "@auth0/auth0-react";

const LoginButton = () => {
  const { loginWithRedirect } = useAuth0();

  return <button onClick={() => loginWithRedirect()}>Log In</bu
};
```

- For the API themselves we could use: **Codehooks.io**
- some pros of codehooks.io



Create an API before breakfast

A new Codehooks project instantly deploys a CRUD REST API, ready to use.

Essential Backend Machinery

JavaScript, HTTP routes, JSON data, Files, Jobs, Queues, Authentication, CLI and more.

Web-based Data Management

Import, export, browse, create, edit, query or delete data easily.

Fast NoSQL storage

Built-in NoSQL and key/value store handles JSON data at lightning speed.

Bring your own MongoDB

You can swap out the built-in datastore with your own MongoDB.

CLI for automations

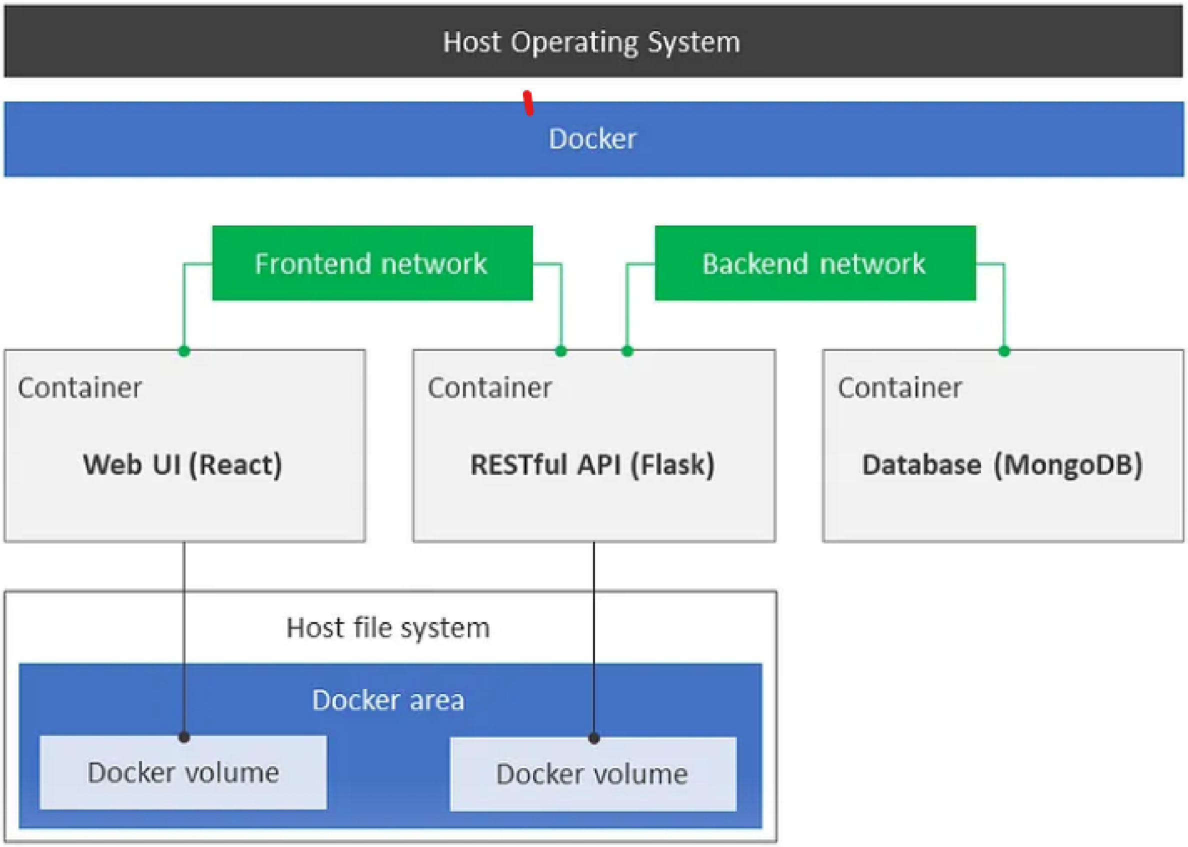
Nearly everything you can do with the UI or API, you can do with the CLI.

- some cons: in javascript and not python

Other option that I saw Gabi was researching was Flask and MongoDB

- are compatible with react
- found a website that shows how to connect all 3 so we won't have to find too much time figuring that out

How to set up a React app with a Flask and MongoDB backend using Docker | by Emil Johansson | The Startup | Medium



Flask:

- **pros: in python (we are more familiar with that), has basic authentication pre-built in, is a great link between the front end and back end that we want to use.**