Building a Web Application with JavaScript:

Planning Your App and Setting up the Project

Steve Buchanan
DevOps Architect

@buchatech | www.buchatech.com

Overview

Tools Used to Develop a JavaScript Web App

Deciding on a Runtime for a JavaScript App

Describing the Files We Will Create for the JavaScript App

- Demo: Creating the File/Directory Structure for a JavaScript App

Initializing a Node.js Project for the JavaScript App

- Demo: Initializing a Node.js Project for the JavaScript App

Tools Used to Develop a JavaScript Web App

Tools Used to Develop a JavaScript Web App

IDEs and Editors

Software for devs used to build apps, it combines common dev tools into a single GUI & tool, has a source code editor, terminal, build automation tools, debugger, & source control integration:
VS Code, Atom, Eclipse, Sublime Text

Frameworks

A structure to hold applications:
Angular, Express, Emberjs, VueJS, React

Package Managers

Software that installs, upgrades, configures, & removes packages maintained in repositories:
npm, yarn, gulp

Libraries

A collection of functions to perform an operation for quick implementation. Used for animations, data visualization, data handling, user interface 7 more:
mocha, socket.io, webpack, jQuery

Linting

Used to maintain the quality of code, readability of code, identify structural problems, & discover ugly syntax errors:
ESLint, Prettier, Flow

Top 5 IDEs

Rank (as of Aug 2022)	IDE	Share	Trend
1	Visual Studio	28.24 %	-0.4 %
2	Visual Studio Code	12.73 %	+1.4 %
3	Eclipse	12.65 %	-1.4 %
4	Android Studio	9.04 %	-0.4 %
5	pyCharm	8.45 %	+0.4 %

Source: Top IDE index:
https://pypl.github.io/IDE.html

Deciding on a Runtime for a JavaScript App

What Is a Runtime?

A runtime is a program or environment that can execute code

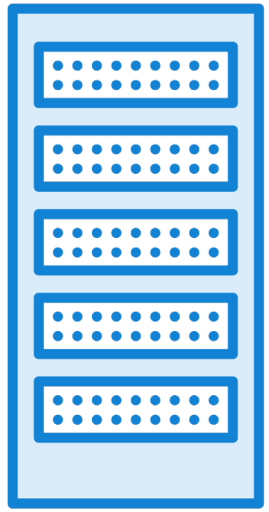
A runtime for JavaScript is anything that can interpret & execute JavaScript code & provide features as well as capabilities for it

Think of a runtime as a layer between your code and the hardware or software that runs it taking care of translating your code into instructions that the underlying system can understand and execute

The runtime also handles errors, memory management, security, and other aspects of running your code

Runtimes for JavaScript

JavaScript code may be executed in one of two runtime environments:



A browser is a runtime environment

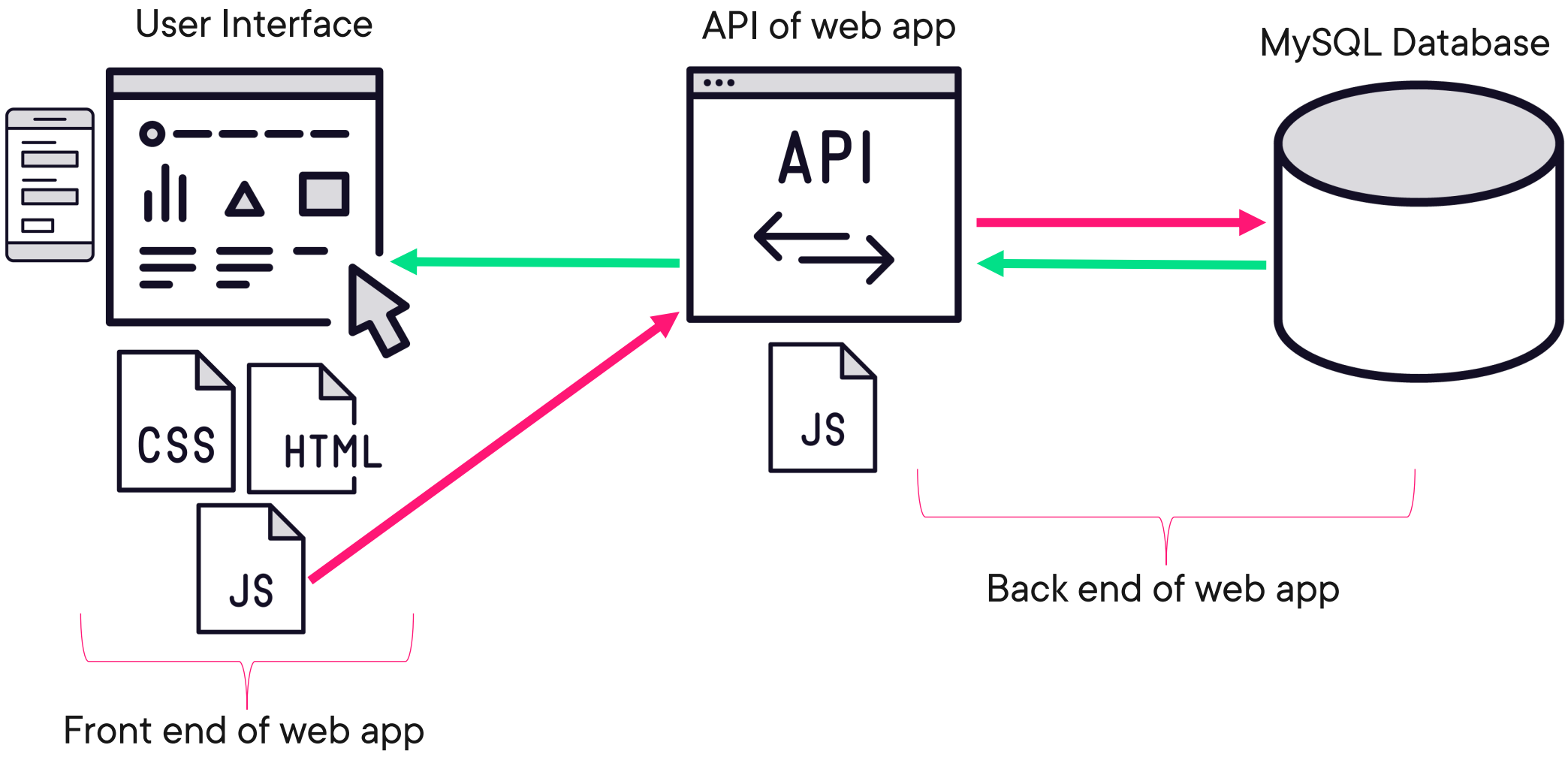
A browser can provide a runtime for JavaScript. It has built-in objects and functions that you can use to manipulate the web page, access the network, handle events, etc.

A Node is a runtime environment

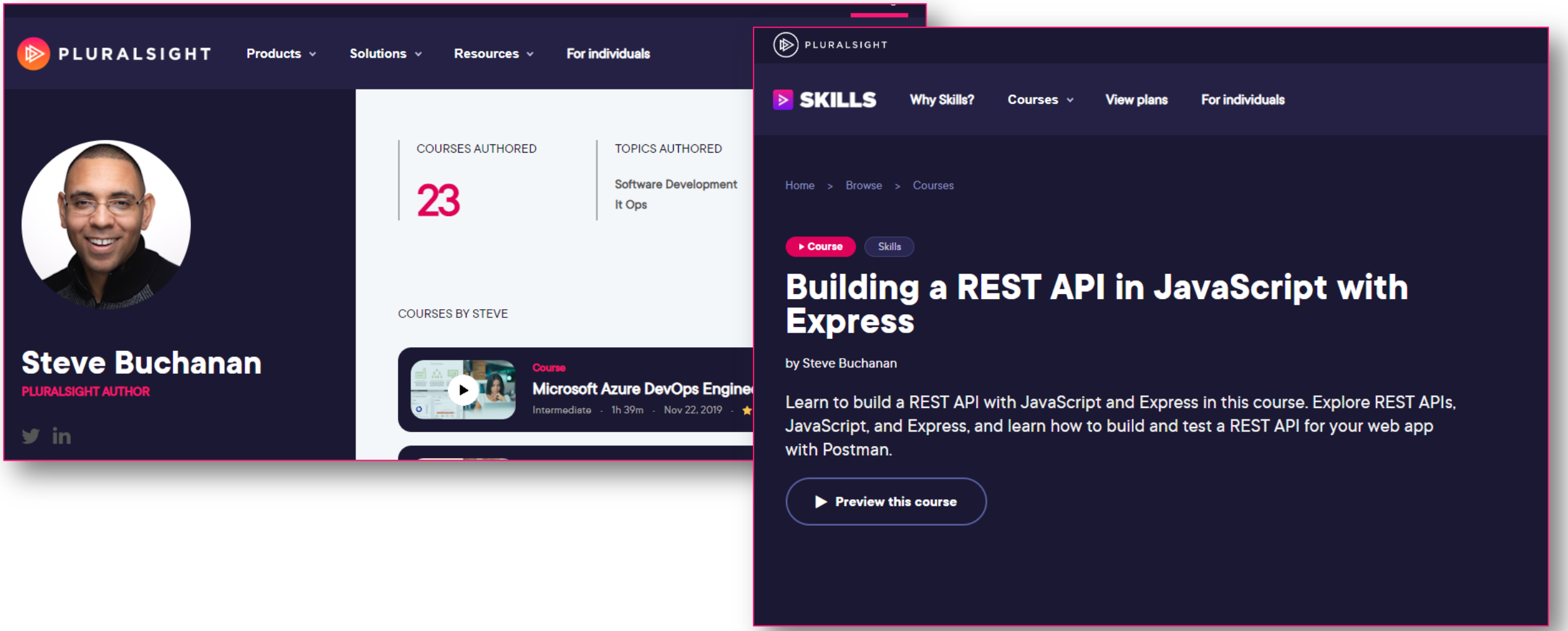
JavaScript is not limited to web browsers, it can also run on other platforms, such as Node.js. Node.js is a runtime for JavaScript on servers. Node.js has its own set of objects & functions that can be used to create web apps, access databases, work with files, etc.

Describing the Files We Will Create for the JavaScript App

Basic JavaScript Web App Architecture

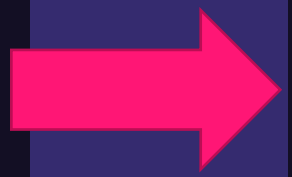


Building a REST API in JavaScript



pluralsight.com/courses/javascript-express-building-rest-api

Feature Based File/Directory Structure for Our Web App



```
- app/
  - config/
- features/
    - feature1/
      - feature1.html
      - feature1.js
    - feature2/
      - feature2.html
      - feature2.js
- views/
    - index.html
-index.js
```

```
#Create directories, sub directories, and files for our web app
sudo -s mkdir -p app/{config,features/{addjoke,jokelist},views} && touch "app/index.js" "app/apiserver.js"
"app/config/dbinfo.js" "app/views/index.html" "app/features/addjoke/addjoke.html" "app/features/addjoke/addjoke.js"
"app/features/jokelist/jokelist.html" "app/features/jokelist/jokelist.js"
```

Create Directories, Sub directories, and Files for our Web App

We can use bash to create our directories, sub directories, and files with a single command.

Demo

Demo: Creating the File/Directory Structure for a JavaScript App

Initializing a Node.js Project for the JavaScript App

To initialize a Node.js project, use ‘npm’ to generate the initial project

Do this by navigating to the project directory & run the following command `npm init’

This command will ask a series of questions & then create a ‘package.json’ file for the project

To skip the questions, use the ‘-y’ flag & automatically say “yes” placing the defaults in the ‘package.json’ file

Initialize a Node.js project

Demo

Demo: Initializing a Node.js Project for the JavaScript App

Summary

In this module we covered:

- Explored the various tools developers use to build web apps in JavaScript
- Decided on a JavaScript runtime for our web app
- Built a general architecture for our JavaScript web app
- Learned about initializing Node.js
- And then we created the files, directories, and initialized Node.js for our project

Why this is important:?

- This is the first step in building a web app with JavaScript. It is ideal to plan out the general architecture of your web app & plan the file/directory structure. Getting the file/directory structure correct up front will save you time later on so that you won't have to do re-work.

Up Next:

Building the Layout, Style, and CRUD for Your App with HTML and CSS