Frontend Webdev in 2018

Max Bo & Neil Ashford

UQCS

2018-08-23T5:30:00+10

Introduction

Us

- Be skeptical

Us

- Be skeptical
- Max Bo + Neil Ashford

Us

- Be skeptical
- Max Bo + Neil Ashford
- Committee

Us

- Be skeptical
- Max Bo + Neil Ashford
- Committee
- @mb / max@maxbo.me

Us

- Be skeptical
- Max Bo + Neil Ashford
- Committee
- @mb / max@maxbo.me
- @artemis / ashfordneil0@gmail.com

Us

- Be skeptical
- Max Bo + Neil Ashford
- Committee
- @mb / max@maxbo.me
- @artemis / ashfordneil0@gmail.com
- Each over a year of professional web dev experience

You

- Year?

You

- Year?
- Studying?

You

- Year?
- Studying?
- Experience with web dev?

You

- Year?
- Studying?
- Experience with web dev?
- Coming to the hackathon next week?

Why Webdev

- Personal sites

Why Webdev

- Personal sites
- Job market

Why Webdev

- Personal sites
- Job market
- Hackathons

Why Webdev

- Personal sites
- Job market
- Hackathons
- Kinda fun to play with, and you can get results fast

What Happens When You Type google.com

- Find the server

What Happens When You Type google.com

- Find the server
- Connect over HTTP

What Happens When You Type google.com

- Find the server
- Connect over HTTP
- Get given a HTML page

What Happens When You Type google.com

- Find the server
- Connect over HTTP
- Get given a HTML page
- Download the things the HTML page links to

What Happens When You Type google.com

- Find the server
- Connect over HTTP
- Get given a HTML page
- Download the things the HTML page links to
- Rendering and parsing

What Happens When You Type google.com

- Find the server
- Connect over HTTP
- Get given a HTML page
- Download the things the HTML page links to
- Rendering and parsing
- Interact with the page

What Happens When You Type google.com

- Find the server
- Connect over HTTP
- Get given a HTML page
- Download the things the HTML page links to
- Rendering and parsing
- Interact with the page
- Page interacts back

Letsa Go

Web Architectures

- Static sites

Web Architectures

- Static sites
- Dynamic sites

Web Architectures

- Static sites
- Dynamic sites
- Single Page App (SPA)

Web Architectures

- Static sites
- Dynamic sites
- Single Page App (SPA)
- Multiple Page App (MPA)

Web Architectures

- Static sites
- Dynamic sites
- Single Page App (SPA)
- Multiple Page App (MPA)
- Server-side vs Client-side rendering

Writing a Web Server

- Use a framework

Writing a Web Server

- Use a framework
- Templating

Writing a Web Server

- Use a framework
- Templating
- REST

Writing a Web Server

- Use a framework
- Templating
- REST
- GraphQL

Databases

- Good for scaling / reliability / persistence

Databases

- Good for scaling / reliability / persistence
- ORMs

Databases

- Good for scaling / reliability / persistence
- ORMs
- prepared statements

Serving Your Code *Fast*

- You have lots of resources

Serving Your Code *Fast*

- You have lots of resources
- Each file is a separate round-trip

Serving Your Code *Fast*

- You have lots of resources
- Each file is a separate round-trip
- Use a bundler

Serving Your Code *Fast*

- You have lots of resources
- Each file is a separate round-trip
- Use a bundler
- Live code reload

Serving Your Code *Fast*

- You have lots of resources
- Each file is a separate round-trip
- Use a bundler
- Live code reload
- Minifying and tree shaking

Serving Your Code *Fast*

- You have lots of resources
- Each file is a separate round-trip
- Use a bundler
- Live code reload
- Minifying and tree shaking
- Makes using libraries so much more feasible on the frontend

Serving Your Code *Fast*

- You have lots of resources
- Each file is a separate round-trip
- Use a bundler
- Live code reload
- Minifying and tree shaking
- Makes using libraries so much more feasible on the frontend
- Can compile (transpile) your code

Libraries Like its ${CURRENT_YEAR}

- `<script src="https://somecdn.com/library.js.min"></script>`

Libraries Like its ${CURRENT_YEAR}

- `<script src="https://somecdn.com/library.js.min"></script>`
- `node_modules/`, the good and the bad

Libraries Like its ${CURRENT_YEAR}

- `<script src="https://somecdn.com/library.js.min"></script>`
- `node_modules/`, the good and the bad
- Package managers (`yarn`)

How to Not Write JavaScript

- `[] == 0` and `"0" == 0` but `[] != "0"`

How to Not Write JavaScript

- `[] == 0` and `"0" == 0` but `[] != "0"`
- `"1" + 2 == "12"` but `"1" - 2 == -1`

How to Not Write JavaScript

- `[] == 0` and `"0" == 0` but `[] != "0"`
- `"1" + 2 == "12"` but `"1" - 2 == -1`
- What if you didn't have to write JavaScript

How to Not Write JavaScript

- `[] == 0` and `"0" == 0` but `[] != "0"`
- `"1" + 2 == "12"` but `"1" - 2 == -1`
- What if you didn't have to write JavaScript
- TypeScript, flow (:/), elm (://), coffee script (:///), webassembly (...)

How *to* Write JavaScript (or similar)

- `const` if you can, `let` if you can't, `var` never

```
function foo(x) {
}

const foo = (x) => {
}
```

How *to* Write JavaScript (or similar)

- `const` if you can, `let` if you can't, `var` never
- `async` and `await`

```
function foo(x) {
}

const foo = (x) => {
}
```

How *to* Write JavaScript (or similar)

- `const` if you can, `let` if you can't, `var` never
- `async` and `await`
- You don't need jquery

```
function foo(x) {
}

const foo = (x) => {
}
```

How *to* Write JavaScript (or similar)

- `const` if you can, `let` if you can't, `var` never
- `async` and `await`
- You don't need jquery
- The story with `this`

```
function foo(x) {
}

const foo = (x) => {
}
```

How *to* Write JavaScript (or similar)

- `const` if you can, `let` if you can't, `var` never
- `async` and `await`
- You don't need jquery
- The story with `this`

```
function foo(x) {
}

const foo = (x) => {
}
```

- ECMAscript

CSS ~~2, Electric Boogaloo~~ 3

- Flex, grid, and why the browser is better at calculating sizes than you are

CSS ~~2, Electric Boogaloo~~ 3

- Flex, grid, and why the browser is better at calculating sizes than you are
- `em` vs `px`

CSS ~~2, Electric Boogaloo~~ 3

- Flex, grid, and why the browser is better at calculating sizes than you are
- `em` vs `px`
- Responsive design

CSS ~~2, Electric Boogaloo~~ 3

- Flex, grid, and why the browser is better at calculating sizes than you are
- `em` vs `px`
- Responsive design
- Removing the boilerplate from CSS

CSS ~~2, Electric Boogaloo~~ 3

- Flex, grid, and why the browser is better at calculating sizes than you are
- `em` vs `px`
- Responsive design
- Removing the boilerplate from CSS
- Libraries - make an informed decision

Frameworks for Front End

- Dynamically change the HTML on your page

Frameworks for Front End

- Dynamically change the HTML on your page
- Saves you from dealing with XSS

Frameworks for Front End

- Dynamically change the HTML on your page
- Saves you from dealing with XSS
- Separation of concerns

Frameworks for Front End

- Dynamically change the HTML on your page
- Saves you from dealing with XSS
- Separation of concerns
  - You deal with data and layout

Frameworks for Front End

- Dynamically change the HTML on your page
- Saves you from dealing with XSS
- Separation of concerns
  - You deal with data and layout
  - The framework deals with rendering, performance and security

Frameworks for Front End

- Dynamically change the HTML on your page
- Saves you from dealing with XSS
- Separation of concerns
  - You deal with data and layout
  - The framework deals with rendering, performance and security
- TODO

Where to Start

Jose Aguinaga Follow

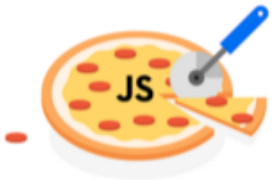
Web Engineer. Previously @numbrs, @plaidhq, @getflynt, currently @MyBit_DApp. Javascript, #people, startups, fintech, privacy, blockchain.

Oct 3, 2016 · 13 min read

How it feels to learn JavaScript in 2016

Minify _everything_
Babelified ES5 w/Uglify
ES2015+ with babel-minify
css-loader + minimize:true

Transpile less code
babel-preset-env + modules:false
Browserlist
useBuiltIns: true



Code-splitting
Dynamic import()
Route-based chunking

Scope Hoisting:
Webpack 3
RollUp

Tree-shaking
Webpack 2+ with Uglify
RollUp
DCE w/ Closure Compiler

Strip unused Lodash modules
lodash-webpack-plugin
babel-plugin-lodash

Optimize "Vendor" libs
NODE_ENV=production
CommonsChunk + HashedModuleIdsPlugin()

Fewer Moment.js locales
ContextReplacementPlugin()

@addyosmani

Best practices for reducing how much JavaScript you're shipping down to users.

Create React App build failing

Create React apps with no build configuration.

- Creating an App – How to create a new app.
- User Guide – How to develop apps bootstrapped with Create React App.

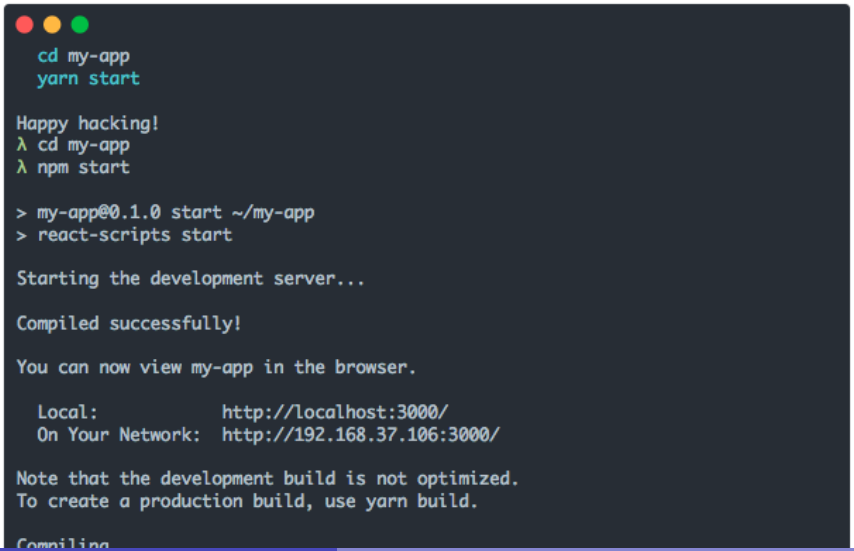
Create React App works on macOS, Windows, and Linux.
If something doesn't work, please file an issue.

Quick Overview

```
npx create-react-app my-app
cd my-app
npm start
```

(npx comes with npm 5.2+ and higher, see instructions for older npm versions)

Then open http://localhost:3000 to see your app.
When you're ready to deploy to production, create a minified bundle with `npm run build`.



Vue CLI 3

🛠️ Standard Tooling for Vue.js Development

Get Started →

This project is sponsored by bit Share Code

Feature Rich

Out-of-the-box support for Babel, TypeScript, ESLint, PostCSS, PWA, Unit Testing & End-to-end Testing.

Extensible

The plugin system allows the community to build and share reusable solutions to common needs.

No Need to Eject

Vue CLI is fully configurable without the need for ejecting. This allows your project to stay up-to-date for the long run.

Graphical User Interface

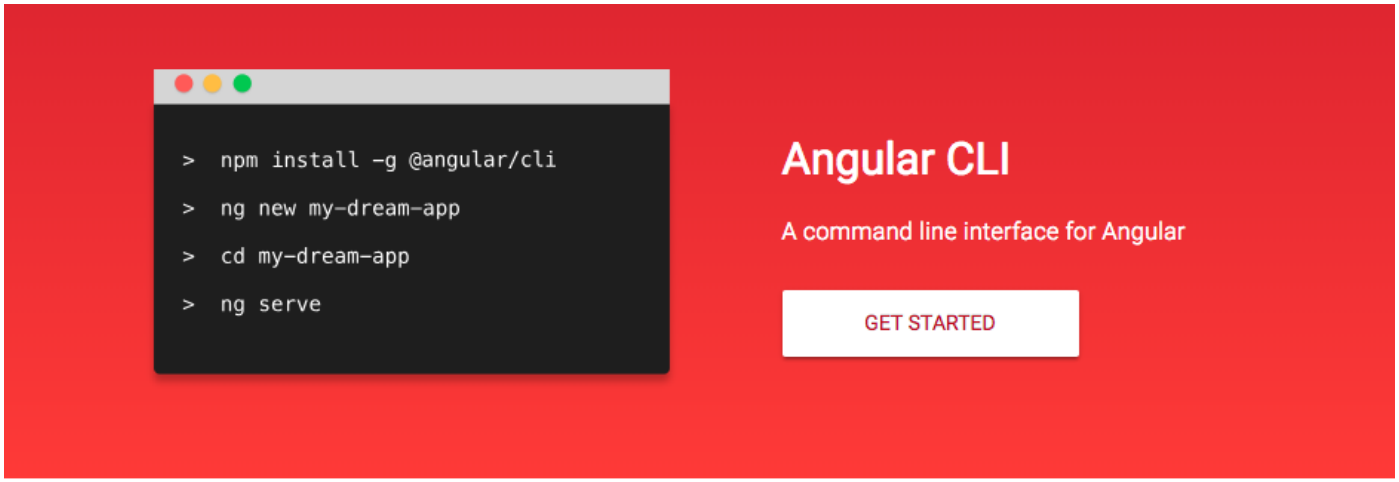
Create, develop and manage your projects through an accompanying

Instant Prototyping

Instantly prototype new ideas with a single Vue file.

Future Ready

Effortlessly ship native ES2015 code for modern browsers, or build your vue



ng new

The Angular CLI makes it easy to create an application that already works, right out of the box. It already follows our best practices!

ng generate

Generate components, routes, services and pipes with a simple command. The CLI will also create simple test shells for all of these.

ng serve

Easily test your app locally while developing.

A case study

What's Included?

Your environment will have everything you need to build a modern single-page React app:

- React, JSX, ES6, and Flow syntax support.
- Language extras beyond ES6 like the object spread operator.
- Autoprefixed CSS, so you don't need `-webkit-` or other prefixes.
- A fast interactive unit test runner with built-in support for coverage reporting.
- A live development server that warns about common mistakes.
- A build script to bundle JS, CSS, and images for production, with hashes and sourcemaps.
- An offline-first service worker and a web app manifest, meeting all the Progressive Web App criteria.
- Hassle-free updates for the above tools with a single dependency.

- Adding Bootstrap
  - Using a Custom Theme
- Adding Flow
- Adding a Router
- Adding Custom Environment Variables
  - Referencing Environment Variables in the HTML
  - Adding Temporary Environment Variables In Your Shell
  - Adding Development Environment Variables In `.env`
- Can I Use Decorators?
- Fetching Data with AJAX Requests
- Integrating with an API Backend
  - Node
  - Ruby on Rails
- Proxying API Requests in Development
  - "Invalid Host Header" Errors After Configuring Proxy
  - Configuring the Proxy Manually
  - Configuring a WebSocket Proxy
- Using HTTPS in Development
- Generating Dynamic `<meta>` Tags on the Server
- Pre-Rendering into Static HTML Files
- Injecting Data from the Server into the Page
- Running Tests
  - Filename Conventions
  - Command Line Interface
  - Version Control Integration
  - Writing Tests
  - Testing Components
  - Using Third Party Assertion Libraries
  - Initializing Test Environment
  - Focusing and Excluding Tests
  - Coverage Reporting
  - Continuous Integration
  - Disabling jsdom

- Deployment
  - Static Server
  - Other Solutions
  - Serving Apps with Client-Side Routing
  - Building for Relative Paths
  - Azure
  - Firebase
  - GitHub Pages
  - Heroku
  - Netlify
  - Now
  - S3 and CloudFront

- Deployment
  - Static Server
  - Other Solutions
  - Serving Apps with Client-Side Routing
  - Building for Relative Paths
  - Azure
  - Firebase
  - GitHub Pages
  - Heroku
  - Netlify
  - Now
  - S3 and CloudFront

```
my-app
--- README.md
--- node_modules
--- package.json
--- .gitignore
--- public
|   --- favicon.ico
|   --- index.html
|   --- manifest.json
--- src
    --- App.css
    --- App.js
    --- App.test.js
    --- index.css
    --- index.js
    --- logo.svg
    --- registerServiceWorker.js
```

Sass (Syntactically Awesome StyleSheets)

Sass is an extension of CSS that adds power and elegance to the basic language. It allows you to use variables, nested rules, mixins, inline imports, and more, all with a fully CSS-compatible syntax. Sass helps keep large stylesheets well-organized, and get small stylesheets up and running quickly, particularly with the help of the Compass style library.

Features

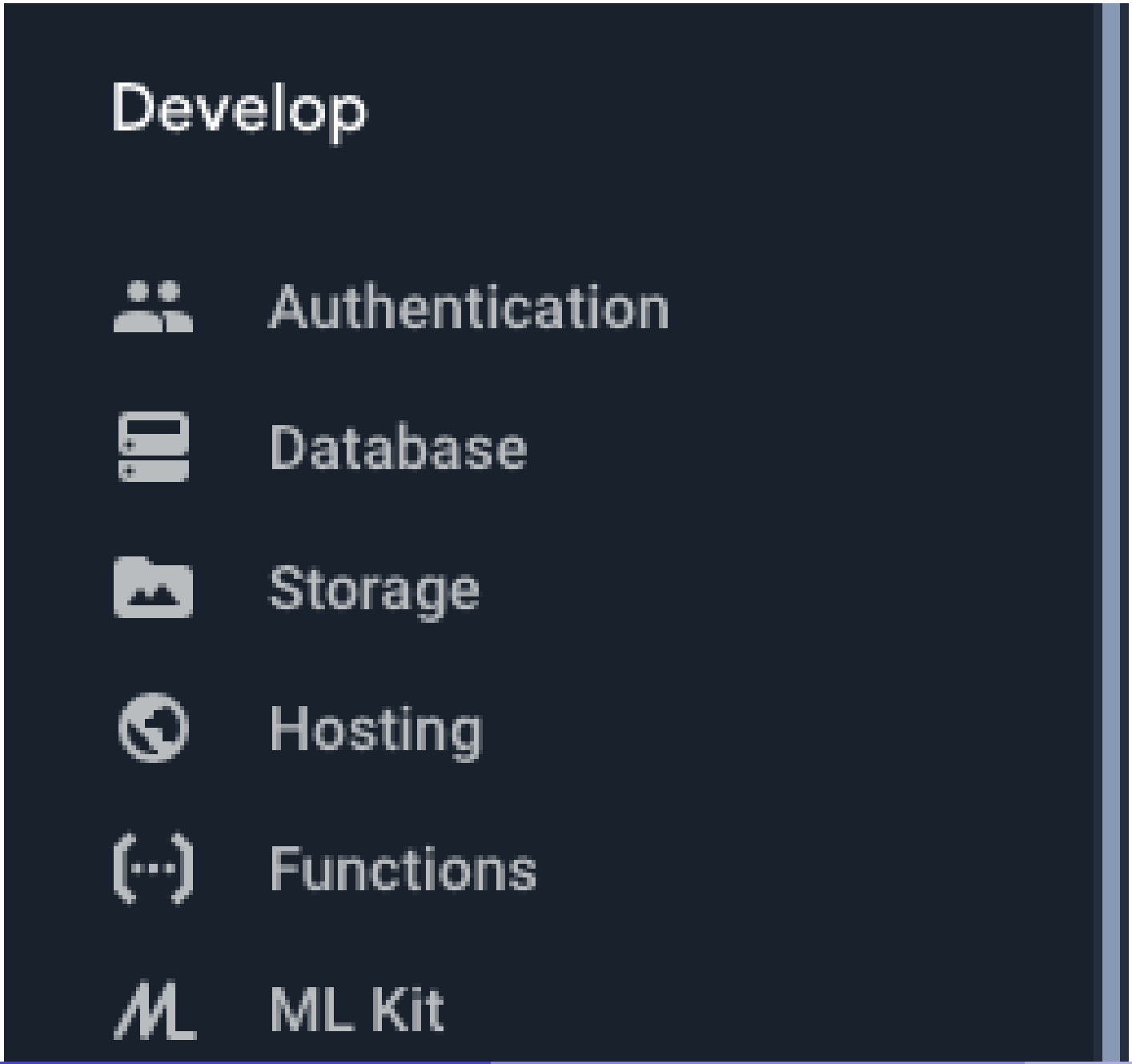
- Fully CSS-compatible
- Language extensions such as variables, nesting, and mixins
- Many useful functions for manipulating colors and other values
- Advanced features like control directives for libraries
- Well-formatted, customizable output

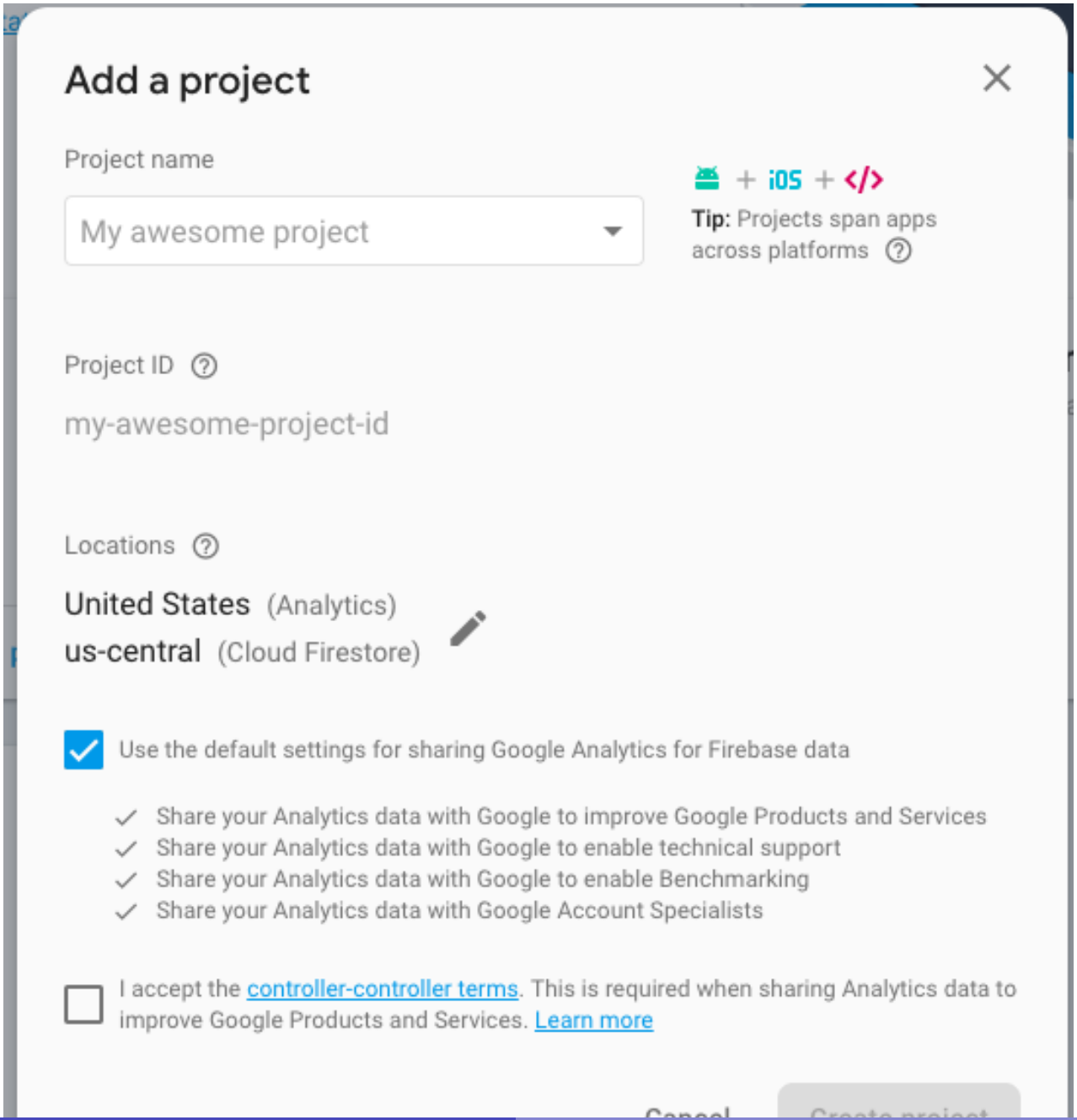
Adding a CSS Preprocessor (Sass, Less etc.)

```
diff --git a/src/App.tsx b/src/App.tsx
index c1bc27c..2d716ed 100644
--- a/src/App.tsx
+++ b/src/App.tsx
@@ -1,19 +1,47 @@
 import * as React from 'react';
 import './App.css';

-import logo from './logo.svg';
+interface ITodo {
+  userId: number
+  id: number
+  title: string
+  completed: boolean
+}

-class App extends React.Component {
+interface IState {
```


Develop
Authentication
Database
Storage
Hosting
Functions
ML Kit


Add a project
Project name
My awesome project
+ iOS + </>
Tip: Projects span apps across platforms
Project ID
my-awesome-project-id
Locations
United States (Analytics)
us-central (Cloud Firestore)
Use the default settings for sharing Google Analytics for Firebase data
Share your Analytics data with Google to improve Google Products and Services
Share your Analytics data with Google to enable technical support
Share your Analytics data with Google to enable Benchmarking
Share your Analytics data with Google Account Specialists
I accept the controller-controller terms. This is required when sharing Analytics data to improve Google Products and Services. Learn more

HEROKU

Jump to Favorites, Apps, Pipelines, Spaces...

Personal > uqcs-webdev-talk-2018

Open app | More

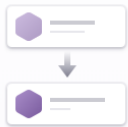
GITHUB MaxwellBo/uqcs-webdev-talk-2018-firebase-example

Overview | Resources | Deploy | Metrics | Activity | Access | Settings

Add this app to a pipeline

Create a new pipeline or choose an existing one and add this app to a stage in it.

Add this app to a stage in a pipeline to enable additional features



Pipelines let you connect multiple apps together and **promote code** between them. Learn more.

Pipelines connected to GitHub can enable **review apps**, and create apps for new pull requests. Learn more.

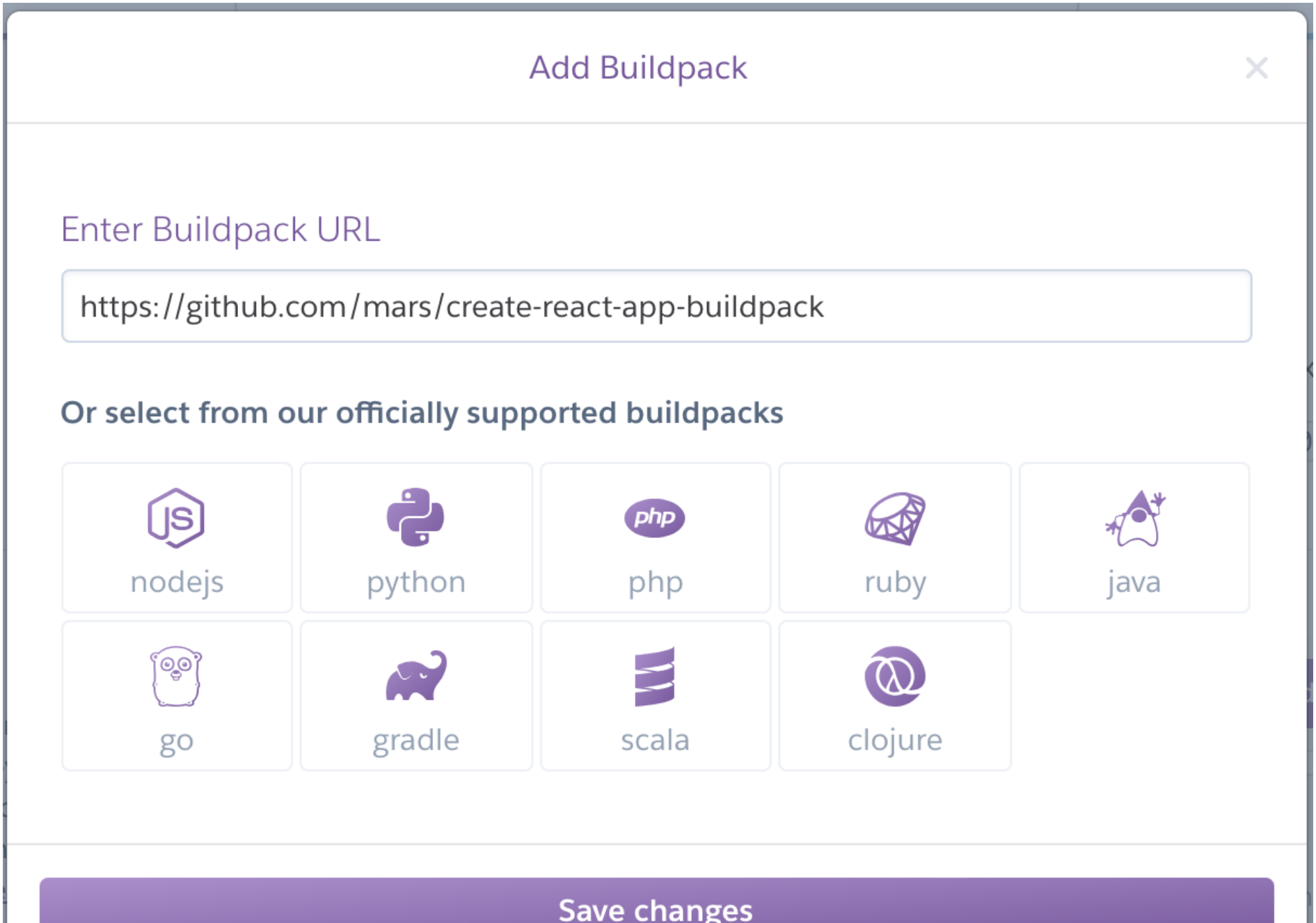
Choose a pipeline

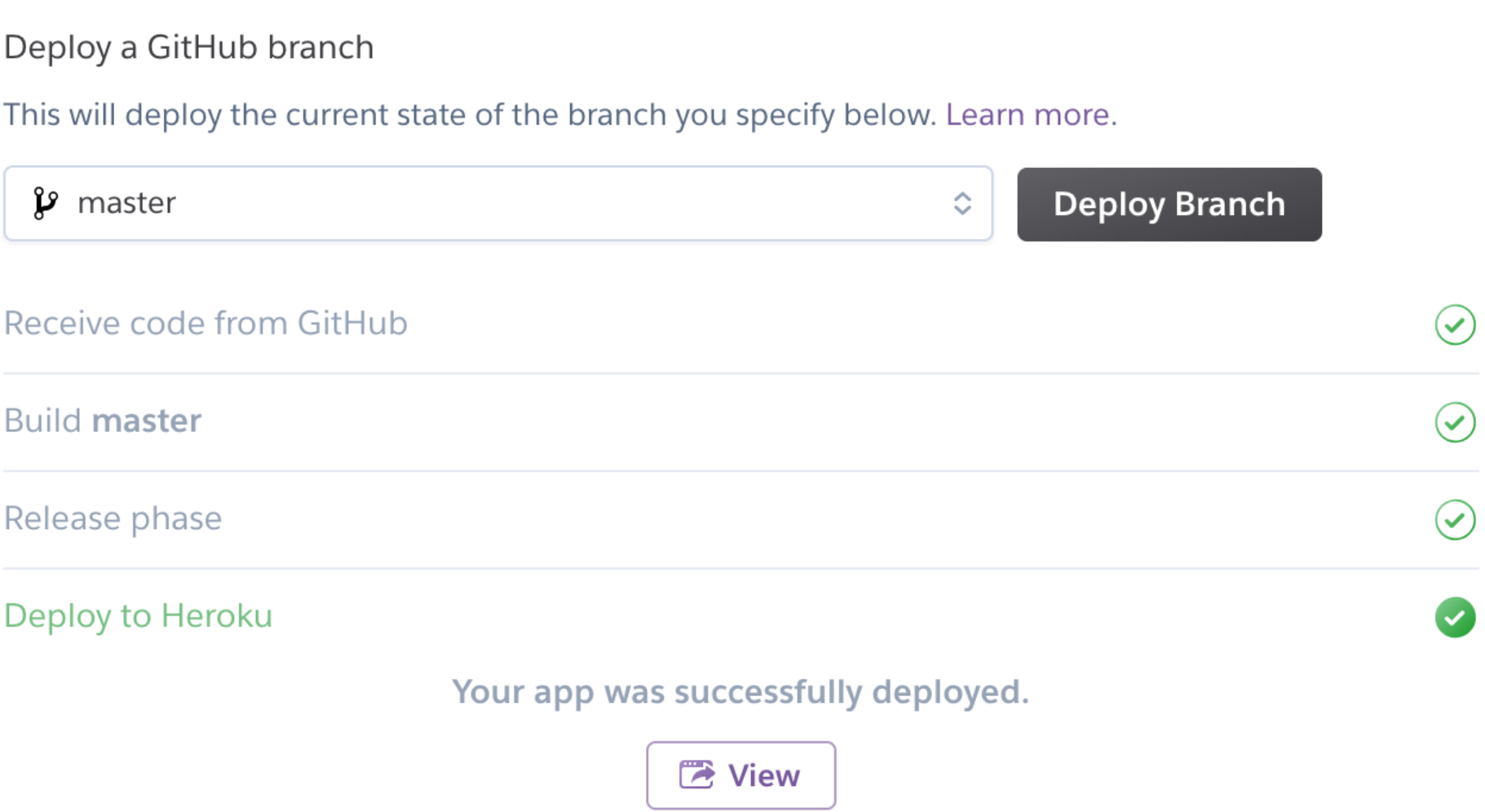
Deployment method

Heroku Git
Use Heroku CLI

GitHub
Connected

Container Registry
Use Heroku CLI


Add Buildpack
Enter Buildpack URL
https://github.com/mars/create-react-app-buildpack
Or select from our officially supported buildpacks
nodejs
python
php
ruby
java
go
gradle
scala
clojure
Save changes


Deploy a GitHub branch
This will deploy the current state of the branch you specify below. Learn more.
master
Deploy Branch
Receive code from GitHub
Build master
Release phase
Deploy to Heroku
Your app was successfully deployed.
View

Conclusion

Thanks

:realheart:

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot
- webpack, parceljs

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot
- webpack, parceljs
- react (create-react-app), angular, vue

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot
- webpack, parceljs
- react (create-react-app), angular, vue
- Mozilla Developer Network (MDN)

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot
- webpack, parceljs
- react (create-react-app), angular, vue
- Mozilla Developer Network (MDN)
- flexbox, grid

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot
- webpack, parceljs
- react (create-react-app), angular, vue
- Mozilla Developer Network (MDN)
- flexbox, grid
- caniuse.com

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot
- webpack, parceljs
- react (create-react-app), angular, vue
- Mozilla Developer Network (MDN)
- flexbox, grid
- caniuse.com
- ECMAscript, typescript, flow

Cheat Sheet and Resources

- github pages, vultr, AWS, GCP, azure, heroku
- Progressive Web App (PWA)
- flask, django, express, koa, ASP.net, SpringBoot
- webpack, parceljs
- react (create-react-app), angular, vue
- Mozilla Developer Network (MDN)
- flexbox, grid
- caniuse.com
- ECMAscript, typescript, flow
- #webdev