Writing Custom Dash Components

> This article is an updated version of the Creating Your Own Components section of the official Dash Documentation.

Table of Contents

Part 1 Getting Started

If You’re Experienced with React

1. Follow the instructions in the Dash Component Boilerplate README. This will generate folder with sample React code, build scripts and publishing scripts.
2. View the *generated* `README.md` file for instructions on how to build the files. Here’s a sample (the `{{ & }}` are variables that are templated in when creating the boilerplate).

If You’re New to React

Introduction

If you’re a Dash developer, at some point you probably have thought about writing your own set of components for Dash. You might have even taken a peek at some of our source code, or taken the `dash-component-boilerplate` for a spin.

However, if you’ve never programmed in JavaScript and/or used React before, you might feel slightly confused. The following will give you some resources to help get you started.

JavaScript

JavaScript is the language of the web - all modern browsers can run it, and most modern web pages use it to make their pages interactive. It is the de-facto standard of front end development and has come a long way since its inception. Today, modern JavaScript has a rich set of features, designed to create a development experience perfectly suited for the web.

If you’re new to JavaScript, we recommend learning the basics first. There are many good tutorials available, such as this one by Mozilla: [JavaScript basics - Learn web development | MDN]

React

React is JavaScript library for building user interfaces, written and maintained by Facebook. It has been very popular over the last few years because it brings the power of reactive, declarative programming to the world of front end development.

React has made it easier to think about user interface code, and its programming model encourages code that’s modular and reusable. It also has a huge, vibrant open-source community that has published all sorts of reusable UI components, from sliders to data tables, dropdowns to buttons.

It is important to realize that React is just JavaScript. React is not a language on its own, nor is it a domain-specific framework that takes years to master. It has a relatively small API, with just a few functions and paradigms to learn before you, too, can use it to write applications for the web.

Dash uses React under the hood to render the user interface you see when you load a web page created with Dash. Because React allows you to write your user interface in encapsulated components that manage their own state, it is easy to split up parts of code for Dash too.

For now, the important thing to know is that Dash components are often simple wrappers around existing React components. This means the entire React ecosystem is potentially usable in a Dash application.

If you’re new to React, we recommend playing around with React a bit before diving into the Component Library. Here are some 3rd party guides:

1. Use create-react-app https://reactjs.org/docs/create-a-new-react-app.html
2. Follow official React tutorial https://reactjs.org/docs/hello-world.html

NPM

NPM is the “Node Package Manager” and it is used to install packages and run scripts. Besides being a package manager (like pip for Python), `npm` also allows you to run scripts and perform tasks, such as creating a project for you (`npm init`), starting up a project (`npm start`), or firing custom scripts (`npm run custom-script`). These scripts are defined in a `package.json` file, which every project that uses `npm` has.

The `package.json` file holds your requirements and `devRequirements` , which can be installed using `npm` install, the same way pip has a `requirements.txt` option you can use in `pip install -r requirements.txt` .

It is usually a good idea to check out a new project’s `package.json` file to see which scripts the project uses.

Installing NPM & Node

a. How to Install Node.js and npm on Windows (makeuseof.com)

b. How to install Node.js on Mac OS (fosstechnix.com)

c. Verify that node is installed by running: `node -v` . Verify that npm is installed by running: `npm -v`

Next, we will create a custom Dash component using the `dash-component-boilerplate` library. We will follow the instructions for setting up a React development environment, complete with the necessary scripts for building our React component for Dash. Let’s get started!

Up Next Part 2 Architecture Overview