## **Overview**

Your code is configured to run on a node-server compatible deployment set-up. This guide will show you how to deploy your code using [Github](https://github.com/) and [Heroku](https://www.heroku.com/). You're certainly welcome to explore any other deployment methods but we recommend this one for its ease in speed and deployment.

## **Requirements**

1. A [GitHub](https://github.com/) account
2. An OrderCloud.ioAccount & Organization
3. A GitHub repo with your OrderCloud.io app code. You can get this one of two ways:
   1. Using the [Accelerator Generator](http://localhost:3000/docs/angularjs/accelerators)
   2. Forking the [AngularJS Seed](https://github.com/ordercloud-api/angular-seed) and adding your desired [Components](https://github.com/ordercloud-api/angular-components)
4. A [Heroku](https://www.heroku.com/) account

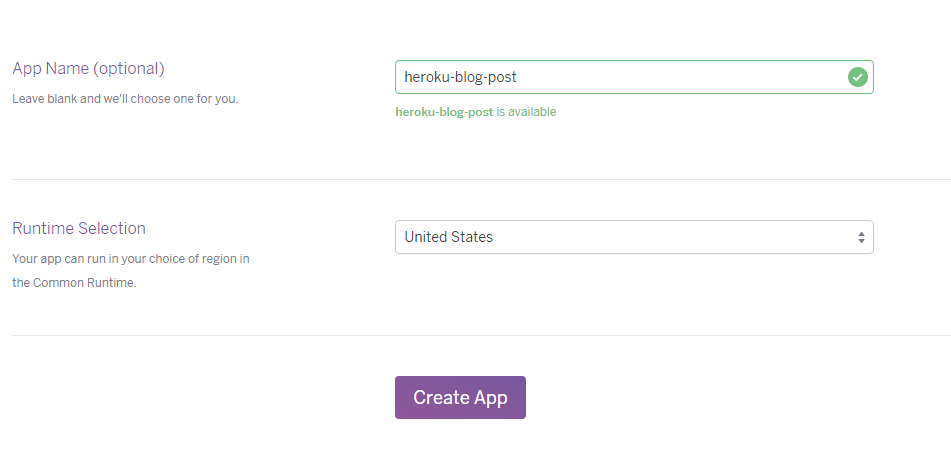
## **Heroku Set-Up Overview**

If you're familiar with the inner workings of Heroku and AngularJS, then you know they don't necessarily mesh or work together by themselves. AngularJS is a Single Page Application which means that it doesn't need a server to render each page on your application. The entire application gets loaded into the browser, and AngularJS handles the routing itself. Heroku, on the other hand, requires a server (Java, Node, PHP, Python, etc.) in order to run properly.

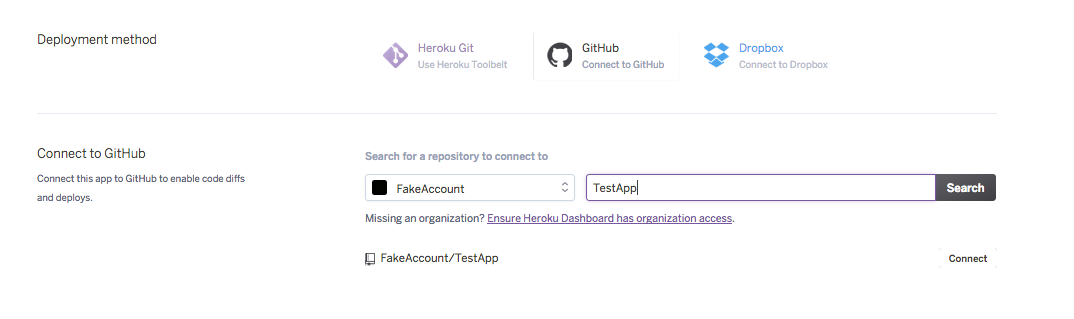
To overcome this, we have added one file, server.js into the OrderCloud.io AngularJS Seed app that allows the app to be served up via a Node server, while still allowing Angular to do it's job in the browser. This file acts as the entry point for the Node server, and we have configured a "catch-all" route for the server that serves up the OrderCloud.io AngularJS Seed to your browser and lets AngularJS do the rest. On the flip side, if you didn't want to run your app in Heroku, or a Node Server for that matter, you can do so by putting your compiled code folder on [AWS](https://aws.amazon.com/) or some Content Delivery Network (CDN) that has a URL to access the index.html file, and your app would work just the same. We give you the flexibility to deliver your app however you want.

## **Creating Your Heroku App**

Log into your [Heroku](https://www.heroku.com/) account, and click **new** at the top right, then click on **Create a new app**. Name your app, and pick your region.



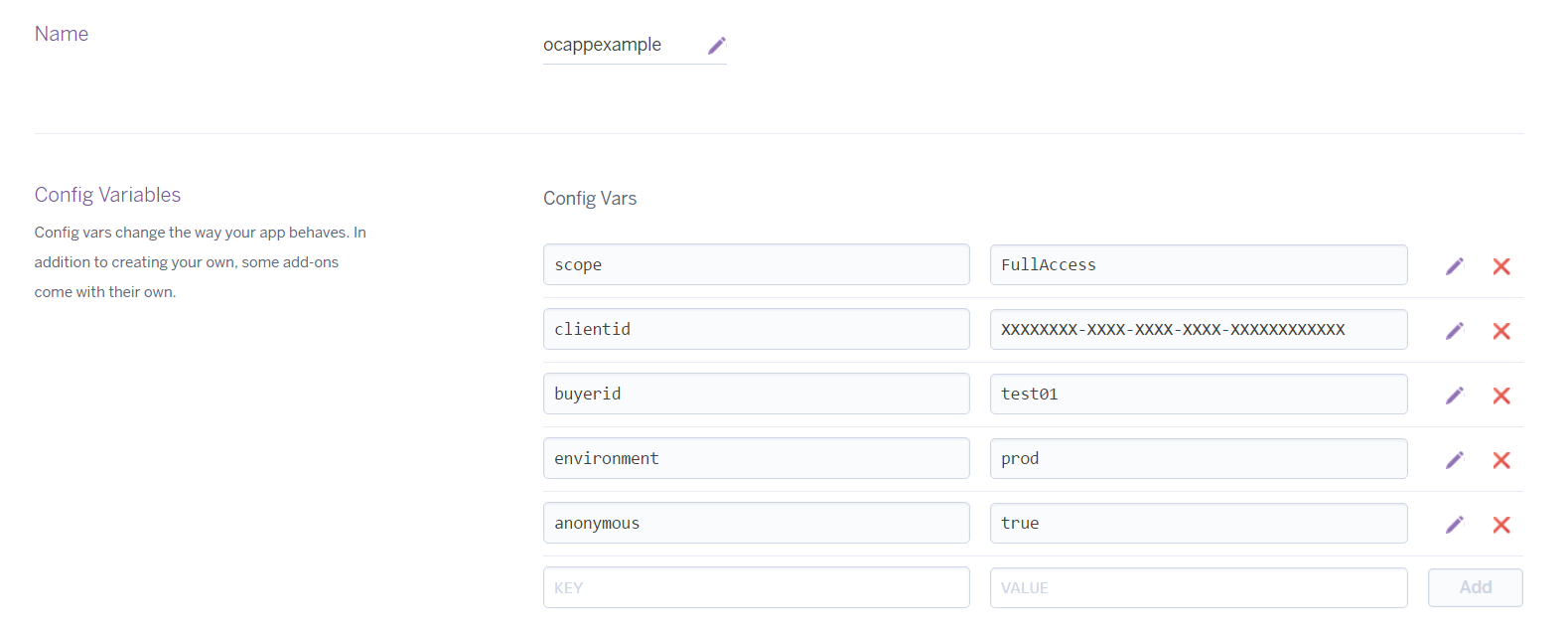
Once you've created the Heroku app, you'll need to connect your GitHub repo. In the Heroku app, select the **Deploy** tab, and scroll down until you see **Deployment Method**. Select **GitHub**. You'll be prompted to pick a repository name. Find your repo and select **Connect**.



## **Configuring Your App in Heroku**

If you followed the [Getting Started](http://localhost:3000/docs/angularjs/getting-started) guide you may have remembered configuring your application. This is just one way you can configure your app, but you can also configure it through Heroku using [Config Variables](https://devcenter.heroku.com/articles/config-vars), which are variables that change the way your app behaves. You can access these in your Heroku App under **Settings**. Click on **Reveal Config Vars**, and you'll be able to add the following:

* scope
* clientid
* buyerid
* environment
* anonymous
* NODE\_ENV

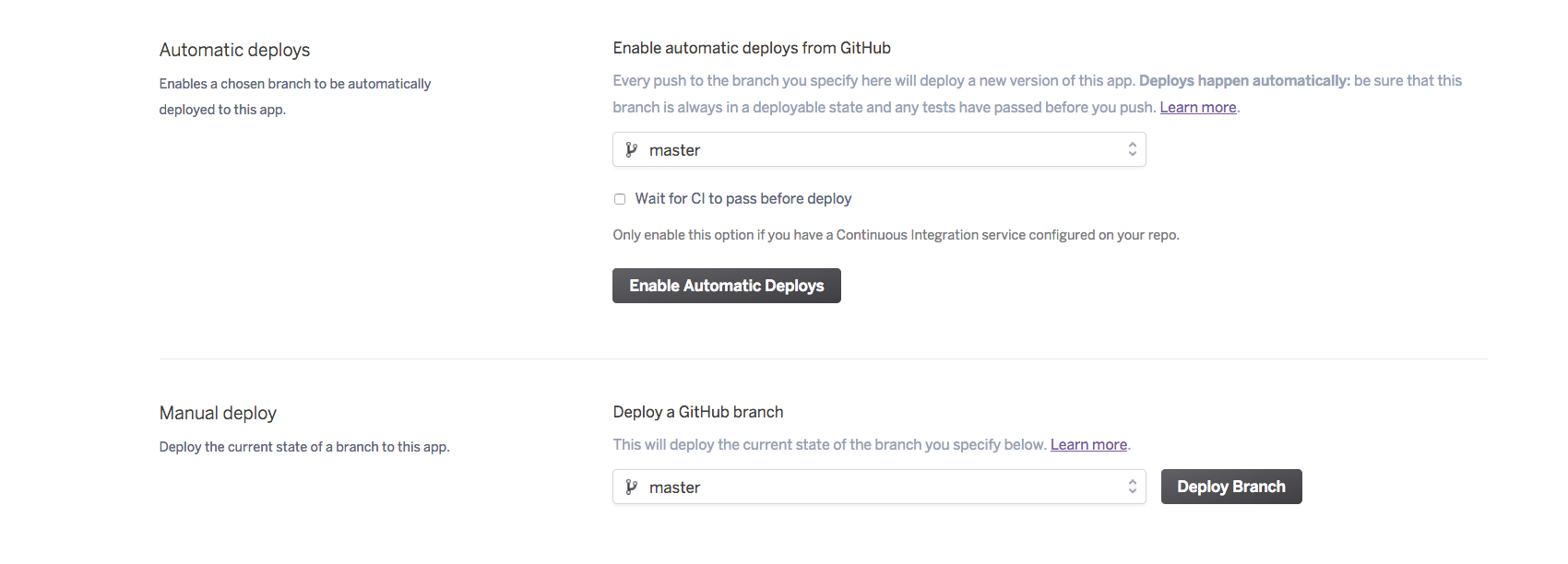


All of these variables, with the exception of NODE\_ENV, should be familiar to you from the [Configuring Your App](http://localhost:3000/docs/angularjs/getting-started#ConfigureApp) section in the Getting Started guide. If NODE\_ENV is set to production it will serve up the compiles files (minified files), excluding it or setting it to anything else will serve up the build files (unminified files). This can come in handy if you need to troubleshoot an error in your front-end code. If you add or change a Config Variable, you'll need to redeploy your app.

Generally, you will want to configure your app in the app.config.json file as demonstrated in the Getting Started Guide, but using config variables can come in handy when you're testing an app and want to more easily change things.

## **Deploy Your App**

You should now be ready to deploy your app. Navigate to the **Deploy** tab and scroll down to the bottom. You'll need to choose if you want to manually or automatically deploy. If you choose automatically, whenever you commit to the selected branch, Heroku will redeploy the app with your new changes.



You'll see the build log roll on through, and then as long as there are no build errors, Heroku will tell you your app is deployed

