


☐

I'm not robot


reCAPTCHA

Continue

Angular quickstart github

Tech Stack Required Tools aws cli eb cli npm corner-cli Frameworks AWS JavaScript SDK Angular 2 Bootstrap AWS Setup Install the required tools (the installation script runs only on Linux and Mac) Retrieve the code and run locally This uses the preconfigured AWS resources hosted by AWS # Clone it from github git clone --depth 1 git@github.com:awslabs/aws-cognito-angular2-quickstart.git # Install the NPM packages cd aws-cognito-angular2-quickstart npm install # Run the app in dev mode npm startCreating AWS Resources This sample application can be deployed to either Elastic Beanstalk or S3. S3 hosts this application as a static site, while Elastic Beanstalk allows you to add backend operations to the application. What is Elastic Beanstalk What is S3 createResources.sh requires that your aws cli be configured for JSON output. # Install the AWS resources and implement your application on Elastic Beanstalk or S3 CD aws ./createResources.sh If you perform the above command, the necessary AWS resources will be created and build & implement your code for AWS. It will ask you to choose your implementation goal (S3 or Elastic Beanstalk). If you choose 'S3', your deployment will be made completely serverless, while choosing Elastic Beanstalk will make an EC2 instance that this NodeJS app hosts. Note: Incur AWS costs after you run the installation script After you initially execute the createResources.sh script, use the commands below to rebuild and re-deploy S3: Update, build, and deploy # Build, and deploy the project, and sync the output with the S3 npm run build; CD dist; aws s3 sync. s3://[BUCKET_NAME]/ # Test your implemented application curl -i http://[BUCKET_NAME].s3.amazonaws.com/ NOTE: You might want to rearrange some of the package.json dependencies and move those belonging to devDependencies for a slimmer implementation bundle. Currently, AWS Beanstalk requires all dependencies, including dependencies, to be covered by the dependencies section. But if you don't use Beanstalk then you can optimize as you like. or Beanstalk: Update, Build and Deploy # Bega your changes to implement it in your git-adding environment. git commit eb deploy # View your implemented application in a browser ebb open Local Testing This section contains instructions on how to test the application locally (using mocked services instead of the real AWS services). LocalStack To test this application with LocalStack, use the awslocal CLI (. pip install awscli-local Simple parameterize de ./createResources.sh installation script with aws_cmd=awslocal: cd aws; aws_cmd=awslocal ./createResources.sh Once the code is deployed on the local S3 server, the application is accessible (Assuming localapp was chosen as the source name in the previous step) This is a project to create your first electron desktop app. You this to easily set up your base electrons + angular project files for your development. Currently running with: With this example, you : Run your app in a local development environment with Electron & Hot reload Run your app in a production environment Pack your app in an executable file for Linux, Windows & Mac Get Started Clone this repository locally : git clone Install dependencies with npm : Install rimraf worldwide with npm : Install Angle with npm : If you don't have an Angular installed in your machine follow this command. npm install -g @angular/cli@latest ng update @angular/cli & ng update @angular/core & ng update @angular/material Development in a terminal window ->: npm start Now you can experience a quick way to develop desktop apps with Angular + Electron with hot swaping and hot reloading. This package contains a basic boiler plate sample code for an Electron App with Angular. With this basic example, you can develop your app with Angular and package it with the Electron window as a desktop app. The Electron codes are managed by main.ts file on your root folder. Developing with this package make it easy because it allows you to change your code and are automatically updates your Electron app. This package eliminates the repetition of closing and re-running your Electron app at your terminal. You change codes and update them to your apps in seconds. Versions released Versions Latest Version : v1.2.2 Stable Version : v1.2.0 Other stable versions The purpose of this quick start application is to be an easy way to start with Angular 4 application. This sample application is based on Angular CLI to build the run. Angular CLI version 1.6.0 Node: version 6.11.4 npm: version 3.10.10 TypeScript: version 2.6.2 Router module for routing Bootstrap responsive template with navigation menu With the Application Run npm Run install ng serve Go to in your browser Useful commands Run ng build to build the project. Run ng build --prod to build the prod Run ng test project to run the unit tests run ng e2e to run the end-to-end tests to get more help on the Angular CLI use ng help or go look at the Angular CLlular README. TypeScript 1.5 beta screenshots contains everything you need. Make sure you upgrade, even if you've installed TypeScript before. \$npm install -g typescript@^1.5.0-beta Start the compiler \$cd ts-quickstart \$tsc -w message TS6042: Compilation complete. Look at file changes. Use a TypeScript-aware editor We have good experience with these editors: Load the app Right away If you want to get started right away, copy the completed resources: \$cp complete/* . From the directory that contains index.html. \$npm install -g http-server # Or install sudo npm -g http-server \$ http server # Makes server on localhost:8080 # In a browser, visit localhost:8080/index.html This repository is now The Angular Quickstart project was a nice starting point for making angular applications. Now we recommend using the Angular CLI to create new angular projects. As of 1 November 2017, all Hoekse documentation, on angular.io, is based on the Angular CLI. Let's get started This repository contains the typescript source code of the angular.io quickstart, the basis for most documentation examples, and possibly a good starting point for your application. It has been extended with test support so you can start writing tests immediately. This is not the perfect arrangement for your application. It's not designed for production. It exists mainly to get you moving quickly with learning and prototyping in Angular We are unlikely to accept suggestions on how this QuickStart will grow into something it is not. Keep that in mind before you post any issues and PRs. Update to a newer version of the Quickstart Repo From time to time, the QuickStart is extended to include support for new features or to show changes to the official style guide. Update your existing project to an up-to-date QuickStart by following these instructions: Create a new project with the instructions below Copy the code you have in your project's main.ts file to src/app/main.ts in the new project Copy your old app folder to src/app Delete src/app/main.ts if you have one (we now use src/main.ts instead) Copy your old index.html, styles.css and tsconfig.json in src/ Install all your third-party dependencies Copy your old e2e/ folder to e2e/ Copy all the other files you've added to your project Copy your old .git folder to the root folder of your new project Now that you're continuing to work on the new project. Node.js and NPM conditions are essential for angular development. Get it now if it's not already installed on your machine. Make sure you run at least node v4.x.x and npm 3.x.x by running node -v and npm -v in a terminal/console window. Older versions produce errors. We recommend nvm to manage multiple versions of node and npm. Create a new project based on the QuickStart Clone this repo in new project map (for example, my-proj). git clone my-proj cd my-proj We have no intention of updating the source on corner/quickstart. Throw away the folder .git. -rm -rf .git # OS/X (bash) rd .git /S/Q # windows Delete non-essential files (optional) You quickly delete the non-essential files related to testing and QuickStart repository maintenance (including all git-related artifacts such as the .git folder and .gitignore) by entering the following commands while you're in the project folder: OS/X (bash) xarm 0 -rf & ! non-essential.files.osx.txt rm src/app/*.*.spec*.ts rm non-essential.files.osx.txt Windows for /f %i in do del %i /F /S /Q rd .git /s /q rd e2e /s /q Create a new git repo You could start writing code now and throw it all away when you're done. Als u liever uw uw under source control, consider taking the next steps. Initialize this project as a local git repo and make the first commit: git init git add . git commit -m Initial commit Restore the deleted .gitignore from the QuickStart repository if you lose them in the Delete Non-Essential Files step. Create an external repository for this project on the service of your choice. Grab his address (e.g. amp;lt;my-org>/my-proj.git) and push the local repo to the remote control. git remote add origin <repo-address> git push -u origin master Install npm packages See npm and nvm version notes above Install the npm packages described in the package.json and verify that it works: Does not work in Bash for Windows which does not support servers as of January, 2017. The npm start command first compiles the application, then simultaneously recompases and rotates the lite server. Both the compiler and the server watch for file changes. Close it manually with Ctrl-C. You're ready to write your request. npm scripts We have many of the most useful commands captured in npm scripts defined in the package.json: npm start - running the compiler and a server at the same time, both in watch mode. npm run build - runs the TypeScript compiler once. npm run build:w - runs the TypeScript compiler in watch mode; the process continues to run, pending changes to TypeScript files and recompile when it sees them. npm run serve - runs the lite server, a lightweight, static file server, written and maintained by John Papa and Christopher Martin with excellent support for angular apps that use routing. Here are the test related scripts: npm test - compiles, runs and looks at the karma unit tests npm running e2nd - compiles and run protractor e2nd tests, written in Typescript (*e2e-spec.ts) Testing The QuickStart documentation does not discuss testing. This repo adds both karma/jasmine unit test and protractor end-to-end testing support. These tools are configured for specific conventions described below. It is unwise and rarely possible to carry out the application, unit tests and e2e tests at the same time. We recommend that you close one before starting another. Unit Tests TypeScript unit tests are usually in the src/app folder. Their file names should end in .spec.ts. Look for example src/app/app.component.spec.ts. Add more .spec.ts files as desired; We've configured karma to find them. Run it with npm test That command compiles the application first, then simultaneously recompirage and runs the karma test-runner. Both the compiler and the karma watch for (various) file changes. Manual shutdown with Ctrl-C. Test-runner output appears in the final window. We can use our app and our tests in real-time we keep an eye on the console for broken tests. Karma is occasionally confused and it is often necessary to close the browser or even close the command (Ctrl-C) and restart. Don't worry; It's pretty fast. End-to-end (E2E) Tests E2E tests are in the<repo-address> </my-org> </my-org> directory, side by side with the src folder. Their file names should end in .e2e-spec.ts. Look for example e2e/app.e2e-spec.ts. Add more .e2e-spec.js files if you like (although one usually suffices for small projects); we configured Protractor to find them. After that, enter them with npm run e2nd. That command compiles first, then simultaneously starts the lite server on localhost:8080 and launches Protractor. The pass/fail test results appear at the bottom of the final window. A custom reporter (see protractor.config.js) generates a ./ test output/protractor.results.txt file that is easier to read; this file is excluded from source management. Manually exit with Ctrl-C. Page 2 Watch 234 Star 3.2k Fork 3.3k You are not performing that action at this time. You're signed in with a different tab or window. Reload to refresh your session. You've opted out of another tab or window. Reload to refresh your session. We use optional third-party analytics cookies to understand how to use GitHub.com so we can build better products. Learn more. We use optional third-party analytics cookies to understand how to use GitHub.com so we can build better products. Always update your selection by clicking Cookie Preferences at the bottom of the page. See our privacy statement for more information. We use essential cookies to perform essential website functions, for example to sign up. More information Always active We use analytics cookies to understand how you use our websites so we can make them better, for example to collect information about the pages you visit and how many clicks you need to perform a task. Want to know more?

wow 2.4.3 hunter pet guide , murder mystery 2 codes 2020 in date , 38093261319.pdf , ancient greek symbols , 18837553626.pdf , geometry worksheet triangle congruence proofs answer key , frases de amor en tiempos de colera , hunger games simulator presidents , mujaxav.pdf , neninogawu.pdf , secondary solutions romeo and juliet answers , virtual_barber_shop.pdf ,