<https://www.concretepage.com/spring-boot/spring-boot-rest-angular-2-jpa-hibernate-mysql-crud-example>

1. <http://start.spring.io/> check

Spring Boot Development Tools

Full-stack web development with Tomcat and Spring MVC

Document RESTful services by combining hand-written and auto-generated documentation

Java Persistence API including spring-data-jpa, spring-orm and Hibernate

H2 database (with embedded support)

1. create mvn project in Intellij with folder created from step 1
2. create Entity, DAO, Service and Controller interfaces and classes
3. do mvn clean build
4. deploy jar by java –jar –Dserver.port=9090 target/xxxx.jar
5. Postman <http://localhost:9090//user/all-articles> “GET”

[

{

"articleId": 2,

"title": "Spring Boot Getting Started",

"category": "Angular"

},

{

"articleId": 1,

"title": "Angular 2 Tutorial using CLI",

"category": "Angular"

}

]

# Angular 2+: Creating CRUD Apps

https://www.lynda.com/Angular-tutorials/Angular-2-Creating-CRUD-Apps/616738-2.html

# What is Angular?

From[Learning Angular](https://www.lynda.com/Angular-tutorials/Learning-Angular/677480-2.html)

- [Narrator] Before we get started, let's take a quick look at what Angular is so that you'll get an idea of what you'll be learning in this course. Angular is an application framework, which means a way of structuring your code that makes it easier to build applications using web technologies. There are two versions of the Angular framework: Angular One, also known as Angular JS, and this version of Angular, which means anything after Angular One. Now that means anything greater than Angular Two, and currently Angular is on version five.

Now, Angular itself, not Angular JS, is a component-based framework, which means that the way you build code is by creating small pieces of code and then connecting them together.Each piece can contain its own scripts, templates and CSS, and can talk to other components.Now, the Angular framework itself is considered to be a modular framework, and that means that you don't just build small components yourself, but that in order to use different features of the framework you load up only the pieces that you need.

And that makes this framework super fast and efficient. One of the key features of the framework is something called data binding. This involves learning how to connect your templates, which are sometimes called views, and your information, which is sometimes called models or data. Angular has a rich templating system that looks like HTML with some additional formats for attributes and even tags that you build and define yourself. It's pretty easy to pick up and work with. Making the connections between components can be a bit challenging, so Angular provides a CLI or command line interface.

This is a little application that you install and load with your terminal that makes it easier to create the connections between components. In the rest of this chapter, we're going to take a look at how the CLI works. So if you're ready to get started, let's take a look at installing the CLI in the next video.

# Using the Angular CLI

From[Learning Angular](https://www.lynda.com/Angular-tutorials/Learning-Angular/677480-2.html)

- [Speaker] Something pretty common in modern frameworks is to provide a quick way to set up your projects. Angular does this using something called the Angular CLI. So let's take a look at what it is and how it works. A CLI is a command line interface, or a set of tools that you access using a terminal application and that allow you to perform a variety of common tasks.The Angular CLI is mainly used to scaffold an application. That means creating an app with common settings quickly using simple commands.

Angular CLI is highly opinionated, so the basic scaffolding command is going to set up an application that may not be the way you would set up your own application, but it's pretty common to use to get started, and I'll walk you through what it does and how to customize things. Using the CLI requires a separate installation, and you have to make sure that you've already installed Node.js, as well as Git. Within Node.js, you'll also be using NPM, or the Node Package Manager, which means you should be familiar with those technologies.

The CLI has several commands that you can use to create, serve, build, generate components, and much more. They all start with the NG prefix. So let's take a look at the most common commands, some of which we'll be using in this course. The first command that you'll get used to is the NG New command. It allows you to create or scaffold a new application. It creates all the files you're going to need for a basic project and then some. We'll cover that file structure in the next video. You use it by typing in NG New, and then the name of the project.

After NG New, you'll probably want to see what your application looks like while in development. For that, you use the NG Serve to run the application in development mode.This will use Webpack, which is a tool that processes the code in your application and runs a temporary live server that will add a update as you make changes to your project. Once you are done developing your application, you'll probably want to publish it to a server. The NG Build command will process your files and get them ready for deploying.

This is a slightly different process than using NG Serve because it produces files that you can push or pass onto a server. Finally, the Angular CLI provides a set of commands that will automatically generate different pieces of angular code for you. The NG G command can be used with different types to generate the code for things like comments, types, and other pieces that you can use in your application. Let's go ahead and install the CLI and then try some of these commands.

To install the Angular CLI, we're going to type in NPM install -G, because this will be a global installation, and then @angular, and then /CLI. Alright, let's go ahead and clear this out so we can have a clean screen. And then, we're going to create a new application by first switching to the desktop. So I'll do a CD and then a ~/Desktop. And then, I'm going to create a new app using the NG New command, and then typing in the name of my application.

I'm going to call this learnangular5. Once that's done, let's go ahead and clear out the screen again. And we'll open this folder up in my favorite editor. For this course, I'm using Visual Studio Code. And you'll notice that inside this Learn Angular folder, we have a lot of folders as well as other sub-folders. Now let's go ahead and see what this application looks like in a browser. To do that, I'm going to switch back up to my terminal.

And now I need to switch to the learnangular5 folder that I created. And then, I can issue any of the NG commands. So the one that you're going to use most often is the NG Serve command. Now that has a -o option, which will open up a browser while it's serving a copy of the application. Alright, so let's take a look at what we get when we install Angular JS.

So I'm going to open this up in my favorite text editor. I'm using Visual Studio Code for this course. You can see all the different folders are right here. And we're going to be covering those in the next video, but before I do that, I wanted to show you a slightly different command that you can run. Let's go ahead and hit Ctrl C and clear this out. And now we're going to run the NG Build command. NG Build is going to generate a new folder, and in that folder, this DIST folder, are going to live all the files that you would use to upload to a server if you were putting this somewhere on the Web.

The Angular CLI is going to save you a lot of time when creating projects. Now it is highly opinionated, so you can see that there are a lot of different folders as well as configuration files.

C:\Users\Edward>node -v

v8.11.3

C:\Users\Edward>npm -v

5.6.0

C:\Users\Edward>npm install -g @angular/cli

C:\Users\Edward\AppData\Roaming\npm\ng -> C:\Users\Edward\AppData\Roaming\npm\no

de\_modules\@angular\cli\bin\ng

> @angular/cli@6.0.8 postinstall C:\Users\Edward\AppData\Roaming\npm\node\_module

s\@angular\cli

> node ./bin/ng-update-message.js

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node\_modules\@an

gular\cli\node\_modules\fsevents):

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@

1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}

)

+ @angular/cli@6.0.8

added 246 packages in 31.231s

C:\Users\Edward>node -v

v8.11.3

C:\Users\Edward>npm -v

5.6.0

C:\Users\Edward>ng

Available Commands:

add Add support for a library to your project.

new Creates a new directory and a new Angular app.

generate Generates and/or modifies files based on a schematic.

update Updates your application and its dependencies.

build Builds your app and places it into the output path (dist/ by default).

serve Builds and serves your app, rebuilding on file changes.

test Run unit tests in existing project.

e2e Run e2e tests in existing project.

lint Lints code in existing project.

xi18n Extracts i18n messages from source code.

run Runs Architect targets.

eject Temporarily disabled. Ejects your app and output the proper webpack conf

iguration and scripts.

config Get/set configuration values.

help Help.

version Outputs Angular CLI version.

doc Opens the official Angular API documentation for a given keyword.

For more detailed help run "ng [command name] --help"