How to make a new project in spring boot , new spring project, spring initializr, initializer , initilijer

->just search spring initializr, : start.spring.io || [Spring Initializr](https://start.spring.io/)

* Dependency ,
  + web dependency (spring web),
  + jpa (spring data jpa) to connect with database
  + Mysql driver
* How to open file / folder to intellije directly using cmd , command : idea64 .

**While making ExamPortal**

* **Make spring boot project**
* **Create schema in mysql workbench “quiz” ke naam se**
* **Now we have to do database configuration**
* # database configuration  
  spring.datasource.url=jdbc:mysql://localhost:3306/appname  
  #?serverTimeZone=UTC  
  spring.datasource.username=root  
  spring.datasource.password=password  
    
  # may be not compulsory to write the driver  
  spring.datasource.driver-class=com.cj.jdbc.Driver  
    
  #hibernate configuration/jpa configuration  
  spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect  
  spring.jpa.hibernate.ddl-auto=update   
  # isse first time jab project chalega to khud ba khud saari entities ban jayegi orkhud table column ban jayenge  
    
  spring.jpa.show-sql=true   
  # iss hamare query fire hone ke baad terminal pe dikh jayega  
  spring.jpa.properties.hibernate.formate\_sql=true   
  # it will formate the query which will be seen in terminal , it will make query easily readable
* **We will make another package inside com.Appname with name “model” to store all database related things like users detail and all**
* **How to create angular project**
  + **Search Angular cli,** 
    - **npm i -g @angular/cli**
    - **“ng help” to know more commands related to angular**
    - **“Ng serve” to run angular app**
* **Angular material** 
  + **ng add @angual/material**
  + **source:** [Getting started | Angular Material](https://material.angular.io/guide/getting-started)

**How to use angular material:**

**For using component: import that component api code to src/app/app.module.ts**

**And add this line in app.module.ts file**

import {MatButtonModule} from '@angular/material/button';

**also write “MatButtonModule” inside @ngModule > imports**

* **for adding bootstrap grid only css in angular project**

**write**

"./node\_modules/bootstrap-grid-only-css/dist/css/bootstrap-grid.min.css"

This above line in “angular.json” inside styles[]

**Source:** [dmhendricks/bootstrap-grid-css: The grid and responsive utilities classes extracted from the Bootstrap 4 framework, compiled into CSS. (github.com)](https://github.com/dmhendricks/bootstrap-grid-css)

* **how to run typescript file,**

**when we run ts code it gets converted to js code then it gets run successfully,**

Typescript code -> compile to javascript -> Run javascript

**Note: Make sure u have node, and npm install**

**sol: make sure nodejs is installed in system then u need to install typescript compile also by writing this line of code in termina**

npm install -g typescript

**and there u go for running a typescript file make sure its name ends with ‘.ts’ and then u can run this file by “tsc filename.ts”**

**it will generate .js file then run this file by “node filename.js”**

**source:** [How to run typescript files from the command line. | Cloudhadoop](https://www.cloudhadoop.com/typescript-execute-files-command-line/)

by the way there are two way to run typescript file

* **one by tsc file.ts, it will generate js file then u need to run this also**

**so basically its two way process**

* **other way is to run by “ts-node file.ts” it will directly run the typescript file, and for this u need to do, “npm i -g ts-node” in terminal.**
* **Make angular components , angular shortcut , pages || html+css+ts**
  + **mkdir components**
  + **ng g c navbar(name that we want to give to the new folder), -> it will create four files in a navbar folder**
* **ng: angular command**
* **g : generate**
* **c : components**

**AWS**

* **create a aws account,**
* **after successfuly** creating **then click EC2 on home page**
* then launch local instance and select ubuntu 18.04
* note down security group name then launch(**launch-wizard-1**)
* make a key name, and save keypair file then again launch

**link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstances.html?icmpid=docs\_ec2\_console**

* **on the instance dashboard u will see instance is running now there is connect button also click that to see how to connect with that instance,**
  + **to connect u need that keypair file also which we download while making the instance**
* **now when**  our instance is created, instance is typically a virtual computer hosted on server, so to download some app in that virtual computer or instance we need to connect to it first , for that we’ll use ssh client method btw there are 3 ways connect to the instance(virtual computer)
* **to**  use ssl way to connect u need a another special cmd, for windows download gitbash or putty, I already have gitbash, and for linux user or mac not need to install another terminal,
  + **as our** virtual computer is ubuntu that’s why need different terminal for windows
* **w**rite all command to connect in gitbash ,
* where to open gitbash, where u have downloaded that ‘keypair file’ .pem file
* **Where** u will get all commands , click on connect button on aws instance page, and click ‘ssh’ to connect by ssh
* **If** everything goes right and u are connect, now u can download applications/software to your virtual computer/ instance from this gitbash terminal
* **We are** doing this aws, thing so that we can deploy our java application on server, we’ll upload our application to this ec2 instance before it we need to download java,tomacat, all the application that are installed in our computer to run the java project locally, we need to install same all software to that virtual computer also
* **How** to update linux ‘apt-get update’

**->we’ll download jdk, tomcat, mysql on ec2 instance to run our future project**

* **How to login to gitbash**

**->sudo su**

**-> now install java/tomcat on that instance , search ubuntu commands which u will write on gitbash to download on virtual computer**

-> to download java on instance: -> write on gitbash teminal after connecting to instance

-> sudo su , it will give administrative privilege and u’ll see # on teminal’s directory now

-> apt-get install openjdk-11-jre openjdk-11-jdk

* If download then check java -version,
* There is java shell also u can write ‘jshell’ on gitbash terminal to write java code in teminal
* **To install tomcat and run tomcat on instance/virtual computer**
* **Check if u are connected with instance or not**
* **Then check again that u are in administrative mode or not, for this u’ll see # where u write command,**
  + **If not in sudo mode then write: sudo su**
  + **Install tomcat by: apt-get install tomcat8**

**//below start/restart doesn’t work**

**-> when** tomcat is downloaded then start by writing command

-> /etc/init.d/tomcat9 start

-> u can check status of tomcat also , to see if it is running or not

-> /etc/init.d/tomcat9 status

-> to stop

->/etc/init.d/tomcat9 stop

->to restart

->/etc/init.d/tomcat9 restart

**->Error while installing tomcat8**

**Unable to locate package tomcat8,**

**Sol: because my ubuntu version was 22.04 for this tomcat8 was not present tomcat9 got installed easily**

My Stackoverflow article: [java - Puppet: unable to locate package tomcat - Stack Overflow](https://stackoverflow.com/questions/26456460/puppet-unable-to-locate-package-tomcat/72478191#72478191)

* **To** check port of tomcat from terminal write : ss -ltn
* **To enable tomcat: systemctl enable tomcat9**
* **To check if tomcat is running: systemctl status tomcat9**
* **To disable tomcat: systemctl disable tomcat9**

**Now our tomcat is running we want to check this in our instance/virtual computer , also tomcat is running on port 8080( check by ‘ss -ltn’)**

* **On aws dashboard create a elastic ip address and associate this ip with our instance and edit inbound rules for our instance and create a new inbound rule to access from anywhere**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**Install mysql server in instance(virtual computer)**

**->connect to your instance again ( connection url gets changed time to time )**

**Now to download mysql write below commands:**

**-> sudo su**

**->apt-get update**

**->apt-get install mysql-server**

**->mysql\_secure\_installation, (after executing this command u have to set password also)**

**->after successuly installation of mysql write to enter in mysql,**

**-> mysql -u root -p**

**-> enter password**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**->for creating new user for mysql**

**-> first enter in mysql using mysql -u root -p**

**-> then write,**

**create user ‘username’@’%’ identified by ‘newpsd’;**

**grant all privileges on \*.\* to ‘username’@’%’;**

**flush privileges;**

**exit**

**mysql -u username -p**

**above command will create a new user % means user can enter from anywhere in desktop into database**

* **Java installation - done**
* **Mysql - done**
* **Tomcat - done**

**Now we have to upload our project to this ec2 instance,**

* **To upload we need ftp client to send data from our local computer to ec2 virutal computer**
* **Download: filezilla ftp client**
* **After installing we will connect this ftpclient to ec2 instance**
* **For connecting open ftpclient and create new site and add some port and localhost detail**
* **Select ssl in protocol while making new site (under general)**
* **You will get host from instance dashboard, host is url where our tomcat is hosted , and port for ssl is always 22**

**-> user is ‘ubuntu’**

**->Then select keypair file under logon type**

**->then connect**

**You will find the tomcat in below directory:**

**File path of my tomcat in my ubuntu instance,**

**/var/lib/tomcat9/webapps**

**->we have to upload all our project to this folder**

**To give read/write permission to filezilla**

**Write below command in termminal**

**->chmod 777 /var/lib/tomcat9/webapps,**

**Now transfer .war file of your project to /webapp location and restart your server and run on that public host there u go…..**