CLI Deep Dive & Troubleshooting

we're going to build our first little app!

If the CLI prompts you to**answer some questions**(some versions do that), you can simply hit **ENTER** for all questions. This will accept the default settings which are fine for this course.

The CLI generates a different welcome screen than you're going to see in my video though. No worries, you'll still be able to follow along without issues! Just make sure to code along **so that your code equals mine** - Angular itself didn't change a bit :)

Depending on the CLI version you're using, you might also need to add the FormsModule  to the imports[]  array in your app.module.ts  file (add it if you don't see it there). You might not fully understand what that all means but we're going to cover that in this course, no worries.

If you don't have FormsModule  in imports[]  in AppModule , please do add it and also add an import at the top of that file: import { FormsModule } from '@angular/forms';

-----

If you want to **dive deeper into the CLI** and learn more about its usage, have a look at its official **documentation**: <https://github.com/angular/angular-cli/wiki>

**You encountered issues during the installation of the CLI or setup of a new Angular project?**

A lot of problems are solved by making sure you're using the latest version of NodeJS, npm and the CLI itself.

**Updating NodeJS:**

Go to nodejs.org and download the latest version - uninstall (all) installed versions on your machine first.

**Updating npm:**

Run [sudo] npm install -g npm  (sudo  is only required on Mac/ Linux)

**Updating the CLI**

[sudo] npm uninstall -g angular-cli @angular/cli

npm cache clean

[sudo] npm install -g @angular/cli

**Here are some common issues & solutions:**

1. **Creation of a new project takes forever (longer than 3 minutes)**  
   That happens on Windows from time to time => Try running the command line as administrator
2. **You get an EADDR error (Address already in use)**  
   You might already have another ng serve process running - make sure to quit that or use ng serve --port ANOTHERPORT  to serve your project on a new port
3. **My changes are not reflected in the browser (App is not compiling)**  
   Check if the window running ng serve  displays an error. If that's not the case, make sure you're using the latest CLI version and try restarting your CLI
4. Bindable Properties and Events
5. Section 2, Lecture 27
6. How do you know to which Properties or Events of HTML Elements you may bind? You can basically bind to all Properties and Events - a good idea is to console.log()  the element you're interested in to see which properties and events it offers.
7. **Important**: For events, you don't bind to onclick but only to click (=> (click)).
8. The MDN (Mozilla Developer Network) offers nice lists of all properties and events of the element you're interested in. Googling for YOUR\_ELEMENT properties  or YOUR\_ELEMENT events  should yield nice results.
9. Important: FormsModule is Required for Two-Way-Binding!
10. Section 2, Lecture 29
11. Important: For Two-Way-Binding (covered in the next lecture) to work, you need to enable the ngModel  directive. This is done by adding the FormsModule  to the imports[]  array in the AppModule.
12. You then also need to add the import from @angular/forms  in the app.module.ts file:
13. import { FormsModule } from '@angular/forms';

Important: FormsModule is Required for Two-Way-Binding!

Section 2, Lecture 29

Important: For Two-Way-Binding (covered in the next lecture) to work, you need to enable the ngModel  directive. This is done by adding the FormsModule  to the imports[]  array in the AppModule.

You then also need to add the import from @angular/forms  in the app.module.ts file:

import { FormsModule } from '@angular/forms';

Click Button Event:

html

<button

class="btn btn-primary" (click)="onCreateServer()">Add Server</button>

ts

allowNewServer = true;

serverCreationStatus = 'No server was created!';

serverName = 'Testserver';

onCreateServer() {

alert(this.serverCreationStatus = 'Server was created! Name is ' + this.serverName);

}

[Evrytrainingbatch01@gmail.com](mailto:Evrytrainingbatch01@gmail.com)

AbCd1234#

JdbcConnection db = **new** JdbcConnection();

db.conect();

**try**{

Connection conn= **null**;

Statement stmt = **null**;

stmt = conn.createStatement();

ResultSet rs =stmt.executeQuery("select username,password from customer");

**while** (rs.next()) {

String user = rs.getString("username");

String pass=rs.getString("password");

System.***out***.println(username);

**if**(username!=user && password!=pass) {

System.***out***.println("Welcome "+user);

}**else** {System.***out***.println("Invalid UserName and Password");

}

}

} **catch**(SQLException ex){

ex.getMessage();

}

**MAVEN Training**:

Git clone “path”

Git add \*

Git commit -m “new files add”

Git push origin master

mvn archtype:generate -DgroupId=com.evryindia -Dartifact=SampleProject -DarchtypeArtifactId=maven-archtype-quickstart -DinteractiveMode=false

-----------------------------------------------------

This command is working:

mvn archetype:generate -DgroupId=com.evryindia -DartifactId=SampleProject -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

go to project folder

>mvn package

>java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app

> java -cp target/SampleProject-1.0-SNAPSHOT.jar com.evryindia.App

**Joins:fetching 2 tables data**

SELECT \* FROM evrybank\_customer\_details.customer AS c

Join evrybank\_customer\_details.user AS u

on u.id=c.customer\_user\_fk;

SELECT c.id,c.firstname,c.lastname,c.age,c.country,c.mobile,c.email FROM evrybank\_customer\_details.customer AS c

Join evrybank\_customer\_details.user AS u

on u.id=c.customer\_user\_fk;

**Tomcat issues:**

**netstat -aon |find /i "listening" |find "8080"**

# taskkill /F /PID 15280

ALTER USER 'student'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'pass123';

# This is Payment Application Assignment which used JSP,Servlet and Hibernate .In this User Can Register and Login then Once enterd into project we cad Add money to wallet ,send Money and view the balance these are the models contained.

**C:\Users\Renuka.Prasada\Documents\dumps\Dump20190405**

POST http://localhost:3759/addCustomer

Input: id means customer\_id

{

"id": 2,

"name": "Ramesh",

"emailId": "rms.gs@gmail.com",

"dateOfBirth": "2011-11-01T18:30:00.000+0000",

"address": "Madiwala",

"city": "Bengaluru",

"country": "Ind",

"mobileNo": 909900,

"loginId": 2894993,

"password": "ZLBH8WLLG2TYRLNBT",

"userType": 0,

"accountStatus": 0,

"lastUpdated": null,

"account": null

}

GET http://localhost:3759/getCustomer/1

Input :GET http://localhost:3759/getCustomer/{customer\_id}

Output:Customer Details with particular {customer\_id}

{

"id": 1, {id means customer\_id}

"name": "Arp",

"emailId": "Arp@gmail.com",

"dateOfBirth": "2011-11-01T18:30:00.000+0000",

"address": "tumkure",

"city": "Bengaluru",

"country": "Ind",

"mobileNo": 909900,

"loginId": 3492119,

"password": "XAMBENONH5AV444VM",

"userType": 0,

"accountStatus": 0,

"lastUpdated": null,

"account": null

}

GET http://localhost:3759/getAllCustomers

@GetMapping("/getAllCustomers")

Input:{getAllCustomers}-->getting all Customers

Output:list of Customers

{

"id": 3, {id means customer\_id}

"name": "sushma",

"emailId": "sushma@gmail.com",

"dateOfBirth": "2001-02-01T18:30:00.000+0000",

"address": "Gubbi",

"city": "Bengaluru",

"country": "Ind",

"mobileNo": 909900,

"loginId": 4305770,

"password": "DLX26L5CIYWSX08BQ",

"userType": 0,

"accountStatus": 0,

"lastUpdated": null,

"account": null

}

PUT http://localhost:3759/updateCustomer/3

Input:

@PutMapping("/updateCustomer/{Customer\_id}")

{

"id": 3, {id means customer\_id}

"name": "sushma",

"emailId": "sushma@gmail.com",

"dateOfBirth": "2001-02-01T18:30:00.000+0000",

"address": "Gubbi",

"city": "Tumkure",-------->City updating

"country": "Ind",

"mobileNo": 909900,

"loginId": 4305770,

"password": "DLX26L5CIYWSX08BQ",

"userType": 0,

"accountStatus": 0,

"lastUpdated": null,

"account": null

}

Output:

{

"id": 3, {id means customer\_id}

"name": "sushma",

"emailId": "sushma@gmail.com",

"dateOfBirth": "2001-02-01T18:30:00.000+0000",

"address": "Gubbi",

"city": "Tumkure", ----City details updated

"country": "Ind",

"mobileNo": 909900,

"loginId": 4305770,

"password": "DLX26L5CIYWSX08BQ",

"userType": 0,

"accountStatus": 0,

"lastUpdated": null,

"account": null

}

DELETE http://localhost:3759/deleteCustomer/3

Input:@DeleteMapping("/deleteCustomer/{Customer\_id}")

{id means customer\_id}

THYMELEAF TRAINING:

<h2>Languages spoken</h2>

<p>You speak <span th:text="${model.languages.size() >=3} ? '3 or more' : ${model.languages.size()}" >x</span> languages</p>

<ul th:if="${model.languages.size() >=3}" >

<li th:each="language : ${model.languages}" th:text="${language}">language1</li>

</ul>

<form action=*"#"* th:action=*"@{/books/searchBooks}"* method=*"POST"*>

<select id=*"bookId"* name=*"bookName"* th:value=*"\*{books}"*>

<option value=*"bookName"* selected=*"selected "* >Book Name</option>

<option th:each=*"book : ${books}"* th:text=*"${book.Book\_name}"*

th:value=*"\*{book.id}"*>Book Name</option>

</select> <input type=*"submit"* value=*"Search Books"* class=*"button"* />

</form>

**Spring Validations:**

Pattern p = Pattern.compile("(0/91)?[7-9][0-9]{9}");

**Applying Validations in Spring RestFull Web Services**

@Size(min=5, max=10, message="Your name should be between 5 - 10 characters.")

    private String name;

    @Min(value=5, message="Please insert at least 5 characters")

    private String lastname;

    @NotNull(message="Please select a password")

    @Length(min=5, max=10, message="Password should be between 5 - 10 charactes")

    private String password;

    @Pattern(regexp="[0-9]+", message="Wrong zip!")

    private String zip;

    @Pattern(regexp=".+@.+\\..+", message="Wrong email!")

    private String email;

    @Range(min=18, message="You cannot subscribe if you are under 18 years old.")

    private String age;

@NotNull(message="is required")

@Size(min=1, message="is required")

private String lastName;

// Applying validation rule and regular expression on freePasses and postalCode

@NotNull(message="is required")

@Min(value=0, message="must be greater than or equal to zero")

@Max(value=10, message="must be less than or equal to 10")

//private int freePasses; Because If u enter alphatics it will throw Number Format Expections to avoid this I am using Integer

// private int freePasses; this converted into string ,but I dont want string I want int dats why I am using Integer

private Integer freePasses;

@Pattern(regexp="^[a-zA-Z0-9]{5}", message="only 5 chars/digits")

private String postalCode;

https://www.tutorialspoint.com/json/json\_java\_example.htm

Session currentSession = sessionFactory.getCurrentSession();

CriteriaBuilder builder = currentSession.getCriteriaBuilder();

CriteriaQuery<User> criteriaquery = builder.createQuery(User.**class**);

Root<User> personRoot = criteriaquery.from(User.**class**);

criteriaquery.select(personRoot);

criteriaquery.where(builder.like(personRoot.get("email"), "%" + theEmail + "%"));

Query<User> users = currentSession.createQuery(criteriaquery);

users.setMaxResults(500);

/\*

\* Criteria cr = currentSession.createCriteria(User.class);

\*

\* cr.add(Restrictions.ilike("email",theEmail,MatchMode.ANYWHERE));

\*/

// cr.add( Restrictions.On<User>(c => c.Field).IsInsensitiveLike(theEmail,

// MatchMode.ANYWHERE));

/\*

\* Query theQuery = null;

\*

\* if (theEmail != null && theEmail.trim().length() > 0) {

\*

\* // search for Email ... case insensitive theQuery=currentSession.

\* createQuery("from User where lower(email) like :theEmail " , User.class);

\* theQuery.setParameter("theEmail", "%" + theEmail.toLowerCase() + "%");

\*

\* }

\*

\* else { // theSearchName is empty ... so just get all customers

\* theQuery=currentSession.createQuery("from User", User.class); }

\*/

// execute query and get result list

**return** users.getResultList();

**Q:** xml and java annotations based  configuration 2 types we can configure but here I want  java annotations based configure and using JPa (Entity Manager ) crud operations

Answer:https://www.baeldung.com/the-persistence-layer-with-spring-and-jpa

public void createInsuranceProduct\_invalidAgeRange\_unsuccessful() {

    insuranceProduct.setAgeRestriction(true);

    insuranceProduct.setLowerAgeRange(55L);

    insuranceProduct.setHigherAgeRange(45L);

        insuranceProductService.createInsuranceProduct(insuranceProduct, edbHeader);

        assertThat(insuranceProduct.getAgeRestriction()).isEqualTo("true");

    }

@Mock

UserDAO userDao;

@InjectMocks

UserServiceImpl userServiceImpl;

@Test

**public** **void** getUserTest() {

*when*(userDao.getUser(2)).thenReturn(**new User()**);

User user = **new** User();

user.setId(2);

user.setEmail("arp@gmail.com");

user.setMobile(9343323451L);

user.setName("arp");

// assertThat(userServiceImpl.getUser(2), is(notNullValue()));

*assertEquals*(userDao.getUser(2),userServiceImpl.getUser(2));

}

Unit Testing with mockit with assert j:

unit-testing-mockito-assertj/

1. Place your cursor on the method’s name inside the test class. I’d recommend [using the keyboard to navigate to the method](http://www.eclipseonetips.com/2011/02/21/the-easiest-ways-to-navigate-methods-in-a-class-using-eclipse-keyboard-shortcuts/), especially if there are many tests.
2. Press **Alt+Shift+X,T** [to run the test](http://www.eclipseonetips.com/2010/01/14/rundebug-a-class-in-eclipse-as-a-junit-test-or-java-app-with-a-single-keystroke/) (or right-click, **Run As > JUnit Test**).
3. If you want to rerun the same test method, just press **Ctrl+F11**. For this to work, ensure that [you’ve told Eclipse to always run the last launched application](http://www.eclipseonetips.com/2010/01/15/always-rundebug-the-last-launched-class-instead-of-the-selected-one-in-eclipse/).

Here’s how to only run testForecastNormal() in the following example:

User user = **new** User();

user.setId(2);

user.setEmail("arp@gmail.com");

user.setMobile(9343323451L);

user.setName("arp");

userService.addUser(user);

userDao.saveUser(user);

*assertThat*(user.getId()).isEqualTo(**true**);

}

public void createInsuranceProduct\_invalidAgeRange\_unsuccessful() {

    insuranceProduct.setAgeRestriction(true);

    insuranceProduct.setLowerAgeRange(55L);

    insuranceProduct.setHigherAgeRange(45L);

        insuranceProductService.createInsuranceProduct(insuranceProduct, edbHeader);

        assertThat(insuranceProduct.getAgeRestriction()).isEqualTo("true");

    }

[‎5/‎15/‎2019 3:22 PM]  Sunil Kumar Gupta:

@Test(expected = AinException.**class**)  
**public void**createInsuranceProduct\_invalidAgeRange\_unsuccessful() {  
    **insuranceProduct**.setAgeRestriction(**true**);  
    **insuranceProduct**.setLowerAgeRange(55L);  
    **insuranceProduct**.setHigherAgeRange(45L);  
    **try**{  
        **insuranceProductService**.createInsuranceProduct(**insuranceProduct**, **edbHeader**);  
    } **catch**(AinException ainException) {  
        *assertThat*(ainException.getErrorEnumeration()).isEqualTo(ErrorEnumeration  
                .***IP\_LOWER\_AGE\_SHOULD\_NOT\_BE\_GREATER\_THAN\_HIGHER\_AGE***);  
        **throw**ainException;  
    }

**This test Case is working :**

@Mock

UserDAO userDao;

UserService userService;

@Before

**public** **void** setUp() **throws** Exception {

MockitoAnnotations.*initMocks*(**this**);

userService=**new** UserServiceImpl();

}

@Test

**public** **void** getUserTest() {

*when*(userDao.getUser(2)).thenReturn(**new** User());

User user = **new** User();

user.setId(2);

user.setEmail("arp@gmail.com");

user.setMobile(9343323451L);

user.setName("arp");

*assertThat*(user.getName()).isEqualTo("arp");

//assertThat(userService.getUser(2),is(notNullValue()));

// assertEquals(userDao.getUser(2),userService.getUser(2));

}

**Pakka code for Mockito unit testing :**

**package com.evry.springproject.TestService;**

**import static org.assertj.core.api.Assertions.assertThat;**

**import static org.hamcrest.CoreMatchers.any;**

**import static org.hamcrest.CoreMatchers.is;**

**import static org.hamcrest.CoreMatchers.notNullValue;**

**import static org.junit.Assert.assertEquals;**

**import static org.junit.Assert.assertThat;**

**import static org.mockito.Matchers.any;**

**import static org.mockito.Mockito.when;**

**import java.util.List;**

**import org.junit.Before;**

**import org.junit.Test;**

**import org.junit.runner.RunWith;**

**import org.mockito.InjectMocks;**

**import org.mockito.Mock;**

**import org.mockito.MockitoAnnotations;**

**import org.mockito.runners.MockitoJUnitRunner;**

**import com.evry.springproject.Dao.UserDAO;**

**import com.evry.springproject.Domain.User;**

**import com.evry.springproject.Service.UserService;**

**import com.evry.springproject.Service.UserServiceImpl;**

**@RunWith(MockitoJUnitRunner.class)**

**public class UserServiceImplTest {**

**UserService userService;**

**@Mock**

**UserDAO userDao;**

**User user = new User();**

**@Before**

**public void init() {**

**userService = new UserServiceImpl(userDao);**

**user.setId(2);**

**user.setEmail("arp@gmail.com");**

**user.setMobile(9343323451L);**

**user.setName("arp");**

**}**

**@Test**

**public void getUserTest() {**

**when(userDao.getUser(user.getId())).thenReturn(user);**

**User user1 = userService.getUser(user.getId());**

**assertEquals(this.user, user1);**

**}**

**@Test**

**public void addUserTest() {**

**User user1 = userService.addUser(user);**

**assertThat("arp").isEqualTo(user1.getName());**

**assertThat(9343323451L).isEqualTo(user1.getMobile());**

**assertThat("arp@gmail.com").isEqualTo(user1.getEmail());**

**assertThat(2).isEqualTo(user1.getId());**

**}**

**@Test**

**public void deleteUserTest() {**

**userService.deleteUser(user.getId());**

**assertThat(user, is(notNullValue()));**

**}**

**@Test**

**public void updateUserTest() {**

**user.setId(2);**

**user.setEmail("arp1@gmail.com");**

**user.setMobile(1343323451L);**

**user.setName("arp1");**

**User user1 = userService.updateUser(user);**

**assertThat(user1.getName()).isEqualTo("arp1");**

**}**

**}**------------------------------------------------------------------------------------