**Configuration Spring Boot Project:**

1.First create spring boot project using the spring.io or as in STS/Eclipse using spring stater project

2.Add the all the dependencies in the pom.xml

pom.xml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<project

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns=*"http://maven.apache.org/POM/4.0.0"*>

<modelVersion>4.0.0</modelVersion>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.5.2</version>

<relativePath />

<!-- lookup parent from repository -->

</parent>

<groupId>com.infy.employee.finance</groupId>

<artifactId>Infy-EmployeeFinance</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Infy-EmployeeFinance</name>

<description>Demo project for the microservice employee finance details</description>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-log4j2</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

<exclusions>

<exclusion>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-validation</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

<!-- Swagger -->

<!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger2 -->

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.9.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/io.springfox/springfox-swagger-ui -->

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger-ui</artifactId>

<version>2.9.2</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-actuator</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Note: Simply copy paste and enter the only change is <groupId>org.springframework.boot</groupId> -> this is for the package name

<artifactId>spring-boot-starter-parent</artifactId> -> project name

<version>2.5.2</version> -> version of the project

3.Add application.properties file

spring.application.name=EmployeeAndTrainingManagementSystemApplication

server.port=8080

# Database properties

spring.datasource.url=jdbc:mysql://localhost:3306/infytel\_employee

spring.datasource.username=root

spring.datasource.password=7775

spring.jpa.show-sql=true

spring.jpa.properties.hibernate.format\_sql=true

server.error.whitelabel.enabled=false

logging.level.org.hibernate.type=info

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.output.ansi.enabled=always

#For actuator

#management.security.enabled = false

#management.port = 8765

General.EXCEPTION\_MESSAGE= Internal error please check the code

#----------------------------API/Controller Messages Environment ----------------------------------------

API.EMPLOYEE\_ADDED\_SUCCESSFULLY = Employee added successfully

API.EMPLOYEE\_UPDATED\_SUCCESSFULLY=Employee updated successfully

API.EMPLOYEE\_DELETED\_SUCCESSFULLY=Employee deleted successfully

4. Logging with the log4j

Lo4j.properties file

# Define the root logger with appender file

log4j.rootLogger = DEBUG, FILE, CONSOLE

# Define the file appender

log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender

log4j.appender.FILE=org.apache.log4j.FileAppender

log4j.appender.FILE.File=techTv.log

# Define the layout for file appender

#%5p - Priority of the logging event

#%t - Name of the thread that initiated the logging event

#%F- File name where the logging issue was requested

#%L - line number that caused the logging message to be generated

log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout

log4j.appender.CONSOLE.layout.conversionPattern=%5p [%F:%L] - %m%n

log4j.appender.FILE.layout=org.apache.log4j.PatternLayout

log4j.appender.FILE.layout.conversionPattern=%5p [%F:%L] - %m%n

name=LoggingFile

rootLogger.level=info

rootLogger.appenderRef.file.ref=LoggerAppender

appender.file.type=File

appender.file.name=LoggerAppender

appender.file.fileName=log/ErrorLog.log

appender.file.layout.type=PatternLayout

appender.file.layout.pattern=%d{dd-MMM-yyyy HH:mm:ss} %level - %m%n

# Console Appender

appender.console.name=ConsoleAppender

appender.console.type=Console

appender.console.layout.type=PatternLayout

appender.console.layout.pattern=%m%n

#Declaring logger for business logic

logger.infyacademy.name=com.infy.infyinterns.utility

logger.infyacademy.level=DEBUG

logger.infyacademy.appenderRef.file.ref=LoggerAppender

logger.infyacademy.additivity=false

logger.tester.name=com.infy.infyinterns

logger.tester.level=INFO

logger.tester.appenderRef.file.ref=ConsoleAppender

logger.tester.additivity=false

#name=LoggingFile

#rootLogger.level=info

#rootLogger.appenderRef.file.ref=LoggerAppender

#

#appender.file.type=File

#appender.file.name=LoggerAppender

#appender.file.fileName=log/ErrorLog.log

#appender.file.layout.type=PatternLayout

#appender.file.layout.pattern=%d{dd-MMM-yyyy HH:mm:ss} %level - %m%n

#

## Console Appender

#appender.console.name=ConsoleAppender

#appender.console.type=Console

#appender.console.layout.type=PatternLayout

#appender.console.layout.pattern=%m%n

#logging.pattern.console=%d{dd-MM-yyyy HH:mm:ss.SSS} [${spring.application.name}] [%thread] %-5relative %-5level %logger{35} - %msg%n

#

#

##Declaring logger for business logic

#logger.infyacademy.name=com.infy.infyinterns.utility

#logger.infyacademy.level=DEBUG

#logger.infyacademy.appenderRef.file.ref=LoggerAppender

#logger.infyacademy.additivity=false

#

#logger.tester.name=com.infy.infyinterns

#logger.tester.level=INFO

#logger.tester.appenderRef.file.ref=ConsoleAppender

#logger.tester.additivity=false

5. Validation for input DTO classes

**public** **class** EmployeeDTO {

@NotNull(message= "{employee.id}")

**private** Integer employeeId;

**private** String employeeName;

@NotNull(message= "{phoneNo.id}")

//@Pattern(regexp="[0-9]{10}",message="Phone Number Should be 10 digit number")

**private** Long phoneNo;

@NotNull(message= "{finance.id}")

**private** Integer financeId;

@NotNull(message= "{course.id}")

//@Pattern(regexp="[A-Z]{1}",message="Course Id must single capital letter from A-Z")

**private** Character courseId;

@NotNull(message= "{project.id}")

**private** Integer projectId;

}

Create the validation.properties

employee.id=Employee ID can not be null!.!Please Enter the employee ID....

phoneNo.id= phone number can not be empty!

project.id = Project id can not b empty!

course.id= course id can not ne empty!

finance.id= Finance ID can not be empty

Point to remember: Always the validation messages it takes the input from the separate file called as the validation messages.

we can not get the validation messages values if we created into the application properties

and how to know the spring boot project about validation.properties file as below:

@PropertySource("classpath:validationMessages.properties")

@SpringBootApplication

**public** **class** DemoApplication {

**public** **static** **void** main(String[] args) {

SpringApplication.*run*(DemoApplication.**class**, args);

}

}

6. Creating the main source code package

Creating the packages like as below:

* com.infy.employee -> main application
* Demoapplication.class
* com.infy.employee.controller -> In this package all the controller ApI classes
* EmployeeController/EmployeeAPI
* ProjectController/projectAPI
* com.infy.employee.dto -> data tranfer object use to tranfer the data from the one layer to the another layer
* EmployeeDTO
* ProjectDTO
* com.infy.employee.entity -> this package is used for to interact with DB so this is called as the entity classes and we annotate it to match with the db
* Employee
* Project
* com.infy.employee.exception -> creating the user defines exception classes
* EmployeeException
* ProjectException
* com.infy.employee.repository -> this layer/ package tis used to interact with data Hibernate Query Lang and persisitenace query lang is used in this layer to fetech the data from the DB in this layer we use the interface which extending to another built in interface JparRepository<Entity,PrimaryKey\_DataType
* EmployeeRepository -> Interface
* ProjectRepository ->Interface
* com.infy.employee.service -> This layer contains all oue business logic to fetech the data from the database also do the some validation before inserting the data to the database
* EmployeeService->interface
* EmployeeServiceImpl ->class Business logic here
* ProjectService->interface
* ProjectServiceImpl->class business logic here
* com.infy.employee.utility -> this layer consist of the AOP -Aspect Oriented Programming (Three cross cutting concern = Transcation ,Logging,Seurity)
* ErrorInfo
* ExceptionHandlerAdvice
* AspectLogging