**Assignment by APURVA LALCHETA**

Frame Work Used-Spring boot

IDE-Spring Tool Suite

Dependencies used-Spring JPA, Spring Web, H2 Database.

Starting with project:

1. Created Model package and added 3 java files:

* Model For Employee and their fields are:

**private** **int** eid; // (Employee id set as primary key for employee table);

**private** String fullname(); // (Employee name)

**private** String designation; // (Employee designation)

**private** String assignedobject; // (Asset name if any asset is assigned else field value is NULL)

**private** **int** assignedassetid; // (Id of the assigned asset if asset assigned else set to 0).

(generated getters and setters and toString for fields)

* Model For Category:

**private** **int** cid; // (Category Id set as Primary key for category table)

**private** String cname; // (Category name)

**private** String description; // (Description about category)

(generated getters and setters and toString for datatypes).

* Model For Asset:

**private** **int** aid; // (Asset id set as primary key for asset table)

**private** String name; // (Asset name)

**private** Date date; // (date on which asset was purchased)

**private** String conditionnote; // (Conditionnote on asset)

**private** String category; // (Category of asset)

**private** String status; // (Status of Asset: ”Assigned” or ”Available” or “Recovered”)

(generated getters and setters toString for datatypes).

1. Wrote connection code with h2 database in application.properties.

**spring.h2.console.enabled=True**

**spring.datasource.platform=h2**

**spring.datasource.url=jdbc:h2:mem:asset**

1. Created 3 interfaces which extends JPA repository in new package repository:

* For Employee
* For Category
* For Asset

1. Created data.sql in resources and inserted some values in all tables of database to save time when starting again or after refereshing STS.
2. Created Controller package and then java file for controller supported with @RestController and attached it with 3 repositories by using @Autowired and formed objects of interfaces in controller class:

* Object name repo for asset:

**@Autowired**

**Assetrepo repo;**

* Object name erepo for employee:

**@Autowired**

**Employeerepo erepo;**

* Object name crepo for categories:

**@Autowired**

**Categoryrepo crepo;**

1. APIs for Category:

* POST-

Link-“**localhost:8080/category**"-Use **POST** method in Postman to add data(use JSON format) to database for category .

* GET-

Link-“**localhost:8080/categories**” – Use **GET** method in Postman to retrieve all records from CATEGORY table.

* PUT-

Link-“ **localhost:8080/category**”-Use **PUT** method in Postman to update data(JSON format) if present or add data if not present.

1. APIs for Employee:

* POST-

Link-“**localhost:8080/employee**"-Use **POST** method in Postman to add data(use JSON format) to database for employee .

* GET-

Link-“**localhost:8080/employees**” – Use **GET** method in Postman to retrieve all records from EMPLOYEE table.

* PUT-

Link-“ **localhost:8080/employee**”-Use **PUT** method in Postman to update data(JSON format) if present or add data if not present.

1. APIs for Asset:

* POST-

Link-“**localhost:8080/asset**”-Use **POST** method in Postman to add data(use JSON format) to database for asset .

* GET-

Link-“**localhost:8080/assets**” – Use **GET** method in Postman to retrieve all records from ASSETMODEL table.

* PUT-

Link-“ **localhost:8080/asset**”-Use **PUT** method in Postman to update data(JSON format) if present or add data if not present.

* Get Assets by names-

Link-“ l**ocalhost:8080/asset/{name}**”-Use **GET** method in Postman and give name( for which you want to extract data), in link in place of {name}.

**NOTE**- **you have to define findByName() method in Assetrepo interface.**

1. API for assigning asset:

Link-“**localhost:8080/asset/{name}/status/{status}/employee/{eid}"-**Use **Put** Method in Postman.

Here you want to assign the asset to the employee just base on name (e.g Laptop),So write name of asset in place of {Asset} then you have to give status here “Available” or “Recovered” in place of {status} for finding available assets and then give ID of Employee you want to give asset in place of {eid}.

**NOTE-In {status} don’t use “Assigned” because you can give assets that are “Available” or “Recovered”.**

It would do following changes-

* Now here if here asset is available then it would update Status in ASSETMODEL table from “Available” or “Recovered ” to “Assigned” so that it is not available to assign again to other employee till it is not recovered .
* It will also update the name of the assigned asset and id of assigned asset in Employee Table and would return message “Asset Assigned Successfully”.

Else if asset of that name is not available, it will return the message “Asset Not Available”.

1. Api for recovering asset:

Link- "**localhost:8080/asset/{eid}"** Use **PUT** method in Postman.

Here you have to give the id of the employee for which you want to recover the asset in place of {eid}.

It will first check whether employee has any asset assigned or not.

If no asset is there with the employee, then it will return a message“Employee has no asset assigned”.

If asset is there :

* It will recover the asset and will update value of status of asset from “Assigned” to “Recovered”, so that it is again available to assign to other employees.
* It will also update the Asset id for assigned object to 0 and name of assigned asset to “NULL” in Employee table and will return a message ”Asset Recovered successfully”.

1. Api for Deleting assets which are not assigned.

Link- "**localhost:8080/asset/{aid}"** Use **DELETE** method in Postman.

Here you have to give id of asset you want to delete.

Then it will check ,if status of asset of given id is “Assigned” or not.

If assets are not assigned, it will be present in only employee database and will have nothing to do with employee table, So it will delete the asset in ASSETMODEL table and return the message ”Asset deleted successfully”.

Else if asset is assigned then it will just return the message ”Sorry, asset is in assigned state”.

**Unit Testing(using JUNIT5 and Mockito)**

1. Created a new class for unit testing and named file in it as AssetControllerTest.

Used @WebMvcTest(AssetController.class) because we have to test for our controller.

1. Used @Autowired for MockMVC (for creating mock test ) and ObjectMapper (for mapping JSON data)
2. Used @MockBean for inserting all dependencies of other class (i.e. model class and repo class).
3. Used @Test for testing each API created by us and I have printed all APIs result on console.