**Problem Statement**

ABC Solutions provides online business consulting and outsourcing services. They want to automate the project allocation process. The application has following two modules:

1. Add new projects
2. Generate report of each project

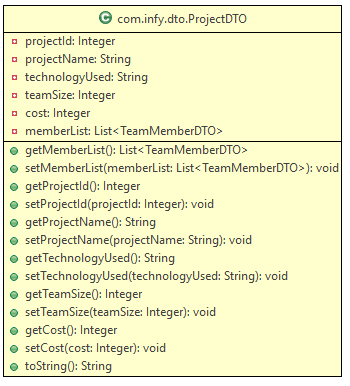
You have to develop the service layer of this application. Follow the following instructions before starting coding:

1. Click [here](https://academy.onwingspan.com/common-content-store/Shared/Shared/Public/lex_auth_013068227510665216192_shared/web-hosted/assets/ABCToTrainee1640929274848.zip) to download​​​​​​​ the project.
2. Import the downloaded project in your workspace.
3. This project is partially implemented. You have to implement the DTO classes, validator class and service class.

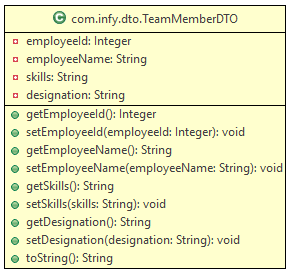
The description of different classes is given as follows:

**1. DTO Classes**

* **ProjectDTO (already implemented)**
  + The following is the class diagram for this class:

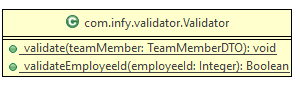


* **TeamMemberDTO (already implemented)**
  + The following is the class diagram for this class:



**2. Validator class (to be implemented)**

This class is used for validating the input values. Implement it using class diagram and description given below:



Note : The methods are static.

**Method description**

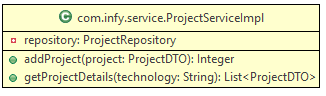
* **validate(TeamMemberDTO teamMember)**
  + This method will receive a TeamMemberDTO object. It should call the respective validator methods (described below) to validate the team member data and in case of failed validation, throw exceptions with the corresponding messages as shown in the table below:

|  |  |
| --- | --- |
| Violation For | Message |
| employeeId | Validator.INVALID\_EMPLOYEEID |

* **validateEmployeeId(Integer employeeId)**
  + The method validates employeeId.
  + The employeeId should be a six digit positive number.
  + If the above condition is satisfied, return true, otherwise return false.

**3. ProjectServiceImpl class (to be implemented)**

This is the service class of this application. Implement it using class diagram and description given below:



Mark it with appropriate stereotype annotation with value as "projectService".

Instance variable description

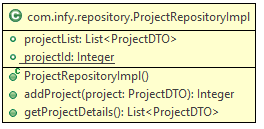
* repository should be injected using autowiring.

**Method description**

* **addProject(ProjectDTO project)**
  + This method receives a ProjectDTO object, validates it and send to Repository to be added to database.
  + This method invokes validate() of Validator class to validate the employeeId of each employee.
  + If all the details are valid, it should invoke addProject() of ProjectRepository, which in turn returns an integer value.
  + It returns the value received in previous step.
* **getProjectDetails(String technology)**
  + This method return a list of all projects that belong to the received technology.
  + It should invoke the getProjectDetails() of ProjectRepository which returns a list of Project object.
  + Filter the list obtained in previous step to get a list of only those projects whose technology is same as the technology received by this method.
  + Create and populate a new List<ProjectDTO> using the filtered list, obtained in previous step.
  + If the filtered list is empty, then throw AbcException with message “Service.PROJECTS\_NOT\_FOUND”, else return the List<ProjectDTO> obtained in previous step.

**4. ProjectRepositoryImpl class**

This is the Repository class of this application and is already implemented.

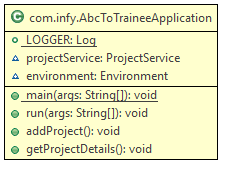


**Method description**

* **addProject(ProjectDTO project)**
  + This method is used to save the project details in database.
* **List<ProjectDTO> getProjectDetails()**
  + The method return details of all the projects.

**5. AbcToTraineeApplication**

This class is used to run your Spring Boot application. Implement it using class diagram and description given below:



**Note :** The success and error messages should be logged using properties file.

**Instance Variable Description**

* Inject the projectService using autowiring.
* Inject the environment using autowiring.

**Method description**

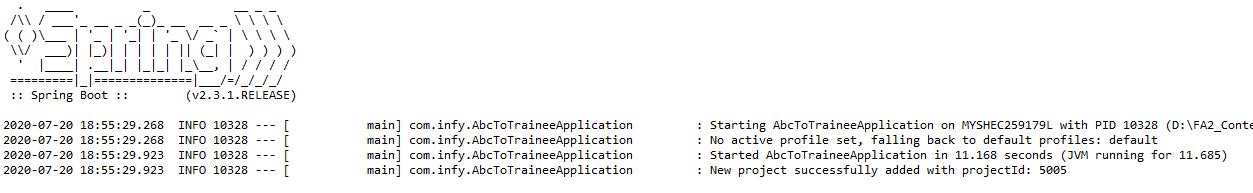
**1. addProject()**

* Invoke addProject() method of ProjectServiceImpl class, which returns an integer for valid data or throws an exception for invalid data.
* The success message should be logged using property as "UserInterface.PROJECT\_ADDED\_SUCCESS".
* For following valid values for project details :

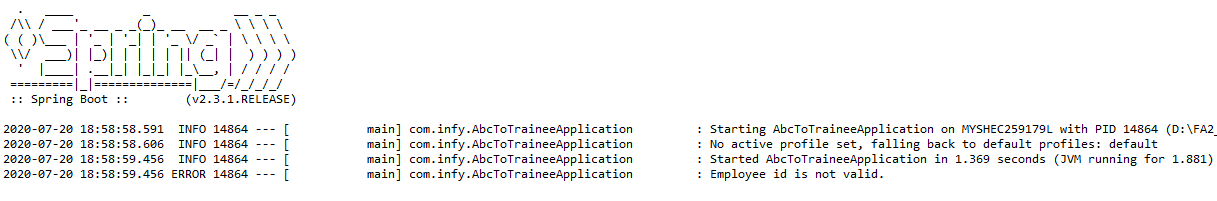
|  |
| --- |
| projectName: FSADM8 |
| cost: 200000 |
| teamSize: 5 |
| technologyUsed: Java |

|  |  |
| --- | --- |
| memberList | employeeId: 722009 employeeName: Robin designation: SSC skills: Java , Oracle |
| employeeId: 722019 employeeName: Monica designation: SSC skills: Java, Python |

The output should be logged in console as:

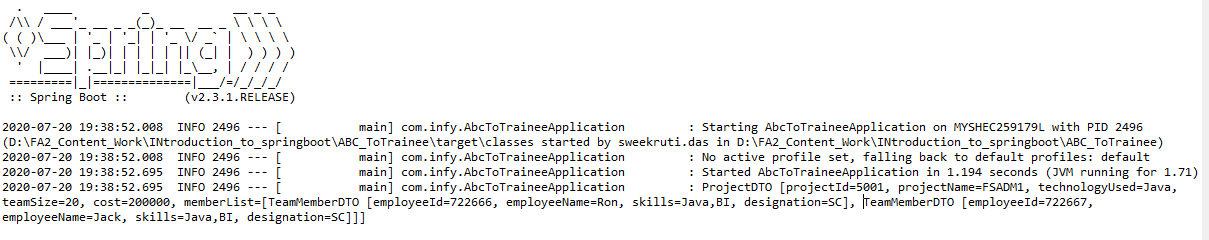


* For invalid **employeeId: 7220**, output should be logged in console as:



**2. getProjectDetails()**

* Invoke getProjectDetails() method of ProjectServiceImpl class.
* For valid **technology: “Java”**, output should be logged in console as:



* For invalid **technology: “Angular”**, output should be logged in console as:

