

Problem Statement:

In this assessment, you are going to build a **Global Education** application which can be used to register a student for applying to different courses in various countries and view all the students by country and year of intake. You will have to implement the following functionalities in the application:

1. Register a student
2. Get all students by country and intake year

Project Structure provided:

File Name	Status
application.properties	
com.infy.globaleducation.GlobalEducationApplication.java	
com.infy.globaleducation.exception.GlobalEducationException.java	
com.infy.globaleducation.service.StudentService.java	
com.infy.globaleducation.utility.ErrorInfo.java	
log4j2.properties	Implemented
GlobalEducation_TableScript.sql	
ValidationMessages.properties	
com.infy.globaleducation.dto.StudentDTO.java	

FILE	INSTRUCTIONS	PROJECT INSTRUCTIONS	PROBLEM STATEMENT
GlobalEducation_TableScript.sql			
ValidationMessages.properties			
com.infy.globaleducation.dto.StudentDTO.java			To be modified
com.infy.globaleducation.entity.Student.java			To be modified
com.infy.globaleducation.utility.ExceptionControllerAdvice.java			
com.infy.globaleducation.api.StudentAPI.java			
com.infy.globaleducation.repository.StudentRepository.java			To be implemented
com.infy.globaleducation.service.StudentServiceImpl.java			
com.infy.globaleducation.validator.StudentValidator.java			

1. Entity Class

com.infy.globaleducation.entity.Student.java

- This class has been implemented for you. You have to map this class to the **student** table with **studentId** as the **primary key**
- Generate the **studentId** using the **IDENTITY** strategy

The screenshot shows a code editor with the following code:

```

G Student
com.infy.globaleducation.entity
    □ studentId: Integer
    □ studentName: String
    □ emailId: String
    □ interestedCountry: String
    □ intakeYear: Integer
    □ studyLevel: String
    □ courseInterested: String
    ● getStudentId(): Integer
  
```



Mozilla Firefox [workspace - ht...]

Go to top

04:23

PROBLEM STATEMENT

```
courseInterested: String  
getStudentId(): Integer  
setStudentId(studentId: Integer): void  
getStudentName(): String  
setStudentName(studentName: String): void  
getEmailId(): String  
setEmailId(emailId: String): void  
getInterestedCountry(): String  
setInterestedCountry(interestedCountry: String): void  
getIntakeYear(): Integer  
setIntakeYear(intakeYear: Integer): void  
getStudyLevel(): String  
setStudyLevel(studyLevel: String): void  
getCourseInterested(): String  
setCourseInterested(courseInterested: String): void
```

2. DTO Class

com.infy.globaleducation.dto.StudentDTO.java

This is a DTO (Data Transfer Object) class that holds Student details such as student id, student name, emailID, interested country, intake year, study level and the interested course.

- Use appropriate annotations from the Java Bean Validation API for validating student details attributes of this class. (Refer description of registerStudent() method of StudentAPI class for validation rules).

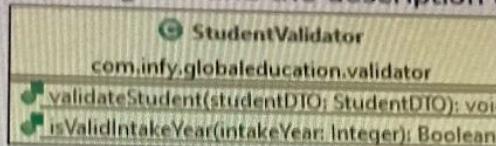
3. Validator Class

com.infy.globaleducation.validator.StudentValidator.java

Implement this class according to the class diagram and the description given below:

```
StudentValidator  
com.infy.globaleducation.validator  
validateStudent(studentDTO: StudentDTO): void
```

Implement this class according to the class diagram and the description given below:



Method Description:

1. **void validateStudent(StudentDTO studentDTO)**

This method receives an object of StudentDTO and calls the respective methods to validate the values and throws an object of GlobalEducationException with the message, based on the violation, as given below,

Violation for	Message
intakeYear	StudentValidator.INVALID_INTAKE_YEAR

2. **Boolean isValidIntakeYear(String intakeYear)**

- This method is to validate the **intakeYear** received
- **intakeYear** should not be a past year
- If the above condition fails return false, else return true

4. Repository

com.infy.globaleducation.repository.StudentRepository.java

- This interface should extend the appropriate repository interface
- Add the following two methods to the interface

4. Repository

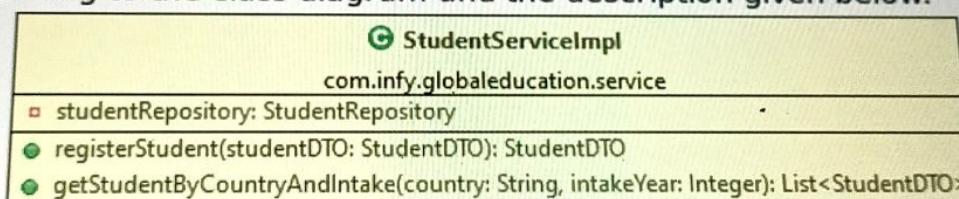
com.infy.globaleducation.repository.StudentRepository.java

- This interface should extend the appropriate repository interface
- Add the following two methods to the interface
 - The first method should accept an emailId and return an optional student object with the given emailId as **Optional<Student>**
 - Use **Query Creation based on method name approach** for this method
 - The second method should accept an interestedCountry and intakeYear and return all the students registered for the given country and intake as **List<Student>**
 - Use **Query Creation based on method name approach** for this method

5. Service Class

com.infy.globaleducation.service.StudentServiceImpl.java

Implement this class according to the class diagram and the description given below:



- Inject the **StudentRepository** object using the proper annotation
- Mark it with the appropriate stereotype annotation with the value **studentService**
- Mark this class with necessary annotation to manage transactions

Method Description:

1. StudentDTO registerStudent(StudentDTO studentDTO)

- This method is used to register a student to the application and add the Student details to the dat

Go to top

Exe

04:24



Method Description:

1. **StudentDTO registerStudent(StudentDTO studentDTO)**

- This method is used to register a student to the application and add the Student details to the database
- Invoke the validateStudent() method of the StudentValidator class to validate the object of the StudentDTO class received as parameter
- Invoke the appropriate method of the StudentRepository interface by passing the emailId, which will then return an Optional object of Student
- If student details are already present, throw an object of the **GlobalEducationException** with the message **StudentService.STUDENT_ALREADY_EXISTS**
- Else, create and populate a new object of **Student** with the appropriate values from the **StudentDTO** object (Hint: You can use the prepareEntity() method of the StudentDTO class for this)
- Then invoke the appropriate method of the StudentRepository to add the **Student** object to the database which will return the same object with the auto-generated **studentId**
- Set the **studentId** of the **Student** object obtained in the previous step to the existing **StudentDTO** object
- Return the modified **StudentDTO** object

2. **List<StudentDTO> getStudentByCountryAndIntake(String country, Integer intakeYear)**

- This method is used to get all the students applying to a given country for a given intake year
- Invoke the appropriate method of the StudentRepository interface which will return all the Students for a given country and intake
- If no Student details are available, throw an object of the **GlobalEducationException** with the message **StudentService.NO_STUDENTS_FOUND**
- Else, populate a list, **List<StudentDTO>** with all the Student details in the ascending order of the

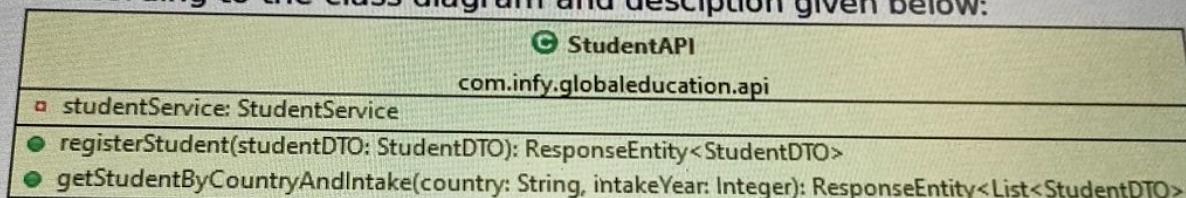


- If no Student details are available, throw an object of the **GlobalEducationException** with the message **StudentService.NO_STUDENTS_FOUND**
- Else, populate a list, **List<StudentDTO>** with all the Student details in the ascending order of the student name and return the sorted list
(Hint: You can use the **prepareDTO()** method of the **StudentDTO** class to convert to **StudentDTO**)

6. API Class

com.infy.globaleducation.api.StudentAPI.java

Implement this class according to the class diagram and description given below:



- Annotate this class with appropriate annotation to declare it as REST controller class
- Annotate this class with appropriate annotation so that all its methods are mapped with **global-edu** as base URL
- Annotate this class with appropriate annotation to enable the bean validation support
- Inject **StudentService** object with appropriate annotation

Method Description:

1. **ResponseEntity<StudentDTO> registerStudent (StudentDTO studentDTO)**

- This method registers a student to the GlobalEducation application
- It accepts **POST** request with URI **student**
- It receives Student details in the form of JSON. The JSON data should be populated in **studentDT** parameter of this method after successful validation

- This method registers a student to the GlobalEducation application
- It accepts **POST** request with URI **student**
- It receives Student details in the form of JSON. The JSON data should be populated in **studentDTO** parameter of this method after successful validation.
- The Student details will consist of the student name, emailID, interested country, intake year, study ^{level} and course interested. Using **Bean Validation API** validate the details according to the description given below (**Note:** The validation message should not be hard-coded. It should be read from the **ValidationMessages.properties** file),

- **studentName:**

If name is not provided then set **Student name is mandatory** as the validation message

If it is provided, it should consist of only letters of the English alphabet and spaces. If invalid, set **Student name should have only letters and spaces** as the validation message

- **emailId:**

If emailId is not provided then set **Email ID is mandatory** as the validation message.

If it is provided, it should be a valid Email ID.

If invalid, set **Email ID should be valid** as the validation message

- **interestedCountry:**

If interestedCountry is not provided then set **Interested country is mandatory** as the validation message.

- **intakeYear:**

If intakeYear is not provided then set **Intake year is mandatory** as the validation message.

- **studyLevel:**

If studyLevel is not provided then set **Study level is mandatory** as the validation message.

- **courseInterested:**

If courseInterested is not provided then set **Course interested is mandatory** as the validation message.

- **studyLevel:**

If studyLevel is not provided then set **Study level is mandatory** as the validation message.

- **courseInterested:**

If courseInterested is not provided then set **Course interested is mandatory** as the validation message.

- Then invoke the appropriate service class method to register the student by passing the **StudentDTO** object which will return the same object with the auto-generated **studentId**.
- Return an object of **ResponseEntity** with the StudentDTO object obtained in the previous step as the response and the HTTP response status as **CREATED**.

2. **ResponseType<List<StudentDTO>> getStudentByCountryAndIntake (String country, Integer intakeYear)**

- This method is to retrieve all the students registered for a given country and intake
- It accepts **GET** request with URI **student/{country}/{intakeYear}**
- The path variable intakeYear should be validated as below (use **Bean Validation API** for validation),
 - **intakeYear** should be a 4-digit number. If invalid, set **Intake year should be of 4 digits** as the validation message

Note: The validation message should not be hard-coded. It should be read from the

ValidationMessages.properties file

- Once validated, invoke the appropriate service class method by passing the country and intakeYear to get the **List<StudentDTO>**
- Return an object of **ResponseType** with the List<StudentDTO> object obtained in the previous step as a response and the HTTP response status as **OK**.

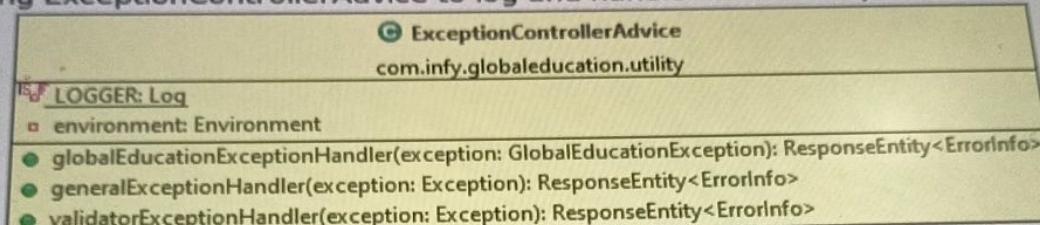
7. Utility Classes



7. Utility Classes

com.infy.globaleducation.utility.ExceptionControllerAdvice.java

Annotate the following ExceptionControllerAdvice to log and handle all the exceptions of the API class,



- Inject Environment objects using appropriate annotation
- The body of the methods have been implemented for you

Method Description:

1. ResponseEntity<ErrorInfo> globalEducationExceptionHandler(GlobalEducationException exception)

- This method is for handling **GlobalEducationException** exception
- Add appropriate annotation to make this method as exception handler for the GlobalEducationException

2. ResponseEntity<ErrorInfo> generalExceptionHandler(Exception exception)

- This method is for handling general exceptions
- Add appropriate annotation to make this method as exception handler for any type of exceptions

3. ResponseEntity<ErrorInfo> validatorExceptionHandler(Exception exception)

- This method is for handling **MethodArgumentNotValidException** and **ConstraintViolationException** exceptions



Mozilla Firefox



[workspace - ht...]

Go to top

04:26

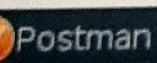
**ConstraintViolationException** exceptions

- Add appropriate annotation to make this method as exception handler method for **MethodArgumentNotValidException** and **ConstraintViolationException** exceptions

Please find in the table below sample input and output for REST end points.

Note: The intake year value has been used considering the current year to be 2021.

REST end point	Request Type(valid/invalid)	End point URI with sample input	Sample Response
Get all the students for given country and intake year	Valid test data	global-edu/student/UK/2021	[{ "studentId": 1, "studentName": "Adam Scott", "emailld": "adams@gmail.com", "interestedCountry": "UK", "intakeYear": 2021, "studyLevel": "Postgraduate-Taught", "courseInterested": "Artificial Intelligence" }, { "studentId": 11 }





ASSESSMENT INSTRUCTIONS

PROJECT INSTRUCTIONS

PROBLEM STATEMENT

```
{  
    "studentId": 11,  
    "studentName": "Penelope  
Wyatt",  
    "emailId":  
    "penny.w28@hotmail.com",  
    "interestedCountry": "UK",  
    "intakeYear": 2021,  
    "studyLevel": "Postgraduate-  
Taught",  
    "courseInterested": "Big Data  
Analytics"  
}  
]
```

Invalid test data 1

global-edu/student/India/2021

```
{  
    "errorMessage": "No student  
details were found for given  
country",  
    "errorCode": 400  
}
```

Invalid test data 2

global-edu/student/UK/202

```
{  
    "errorMessage": "Intake year  
should be of 4 digits",  
    "errorCode": 400  
}
```

Invalid test data 3

global-edu/student/UK/20211

```
{  
    "errorMessage": "Intake year  
should be of 4 digits",  
    "errorCode": 400  
}
```

Go to top

Execute

04:26



		INSTRUCTIONS	PROBLEM STATEMENT
	Invalid test data 3	global-edu/student/UK/20211	<pre>{ "errorMessage": "Intake year should be of 4 digits", "errorCode": 400 }</pre>
Register a student	Valid test data	<p>global-edu/student</p> <p>Sample JSON:</p> <pre>{ "studentName": "Andrew Hart", "emailId": "andrew_h@ymail.com", "interestedCountry": "USA", "intakeYear": 2022, "studyLevel": "Postgraduate- Research", "courseInterested": "Data Science" }</pre>	<pre>{ "studentId": 17, "studentName": "Andrew Hart", "emailId": "andrew_h@ymail.com", "interestedCountry": "USA", "intakeYear": 2022, "studyLevel": "Postgraduate- Research", "courseInterested": "Data Science" }</pre>
	Invalid test data 1	<p>global-edu/student</p> <p>Sample JSON:</p> <pre>{ "studentName": null, "emailId": null, "interestedCountry": null, "intakeYear": null, "studyLevel": null,</pre>	<pre>{ "errorMessage": "Student name is mandatory, Interested country is mandatory, Course interested is mandatory, Intake year is mandatory, Study level is mandatory, Email ID is mandatory". "errorCode": 400 }</pre>



	<pre> "intakeYear": null, "studyLevel": null, "courseInterested": null } } </pre>	<pre> "errorCode": 400 } } </pre>
Invalid test data 2	<p>global-edu/student</p> <p>Sample JSON:</p> <pre> { "studentName": "Andrew123 Hart", "emailld": "andrew_h@ymail.com", "interestedCountry": "USA", "intakeYear": 2022, "studyLevel": "Postgraduate- Research", "courseInterested": "Data Science" } </pre>	<pre> { "errorMessage": "Student name should have only letters and spaces", "errorCode": 400 } </pre>
Invalid test data 3	<p>global-edu/student</p> <p>Sample JSON:</p> <pre> { "studentName": "Andrew Hart", "emailld": "andrew_hymail", "interestedCountry": "USA", "intakeYear": 2022, "studyLevel": "Postgraduate- Research" } </pre>	<pre> { "errorMessage": "Email ID should be valid", "errorCode": 400 } </pre>

Go to

04:2



Mozilla Firefox

[workspace - ht...]

ASSESSMENT INSTRUCTIONS	PROJECT INSTRUCTIONS	PROBLEM STATEMENT
	<p>Sample JSON:</p> <pre>{ "studentName": "Andrew Hart", "emailld": "andrew_hymail", "interestedCountry": "USA", "intakeYear": 2022, "studyLevel": "Postgraduate-Research", "courseInterested": "Data Science" }</pre>	<p>should be valid", "errorCode": 400 }</p>
Invalid test data 4	<p>global-edu/student</p> <p>Sample JSON:</p> <pre>{ "studentName": "Andrew Hart", "emailld": "andrew_h@ymail.com", "interestedCountry": "USA", "intakeYear": 2020, "studyLevel": "Postgraduate-Research", "courseInterested": "Data Science" }</pre>	<p>{ "errorMessage": "Intake year cannot be a past year", "errorCode": 400. }</p>
Invalid test data 5	<p>global-edu/student</p> <p>Sample JSON:</p>	<p>{ "errorMessage": "Student with the same email id already exists" }</p>



04:27

Go to top

Ex



ASSESSMENT INSTRUCTIONS

PROJECT INSTRUCTIONS

PROBLEM STATEMENT

```
studentName": "Andrew Hart",  
"emailId":  
"andrew_h@ymail.com",  
"interestedCountry": "USA",  
"intakeYear": 2020,  
"studyLevel": "Postgraduate-  
Research",  
"courseInterested": "Data  
Science"  
}
```

Invalid test data 5

global-edu/student

Sample JSON:

```
{  
    "studentName": "Andrew Hart",  
    "emailId": "adam.s@gmail.com",  
    "interestedCountry": "USA",  
    "intakeYear": 2022,  
    "studyLevel": "Postgraduate-  
Research",  
    "courseInterested": "Data  
Science"  
}
```

```
{  
    "errorMessage": "Student with  
the same email id already exists",  
    "errorCode": 400  
}
```

PREV

Go to

04