

Mozilla Firefox

file:///home/user/Desktop/user_repo/QP/index.html#

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

Infosys Certified Spring Associate Hands-On Assessment

Problem Statement:

In this assessment, you are going to build a **Meeting Scheduler** application which can be used to schedule meetings for a particular team and view all the meetings scheduled by a user. You will have to implement the following functionalities in the application,

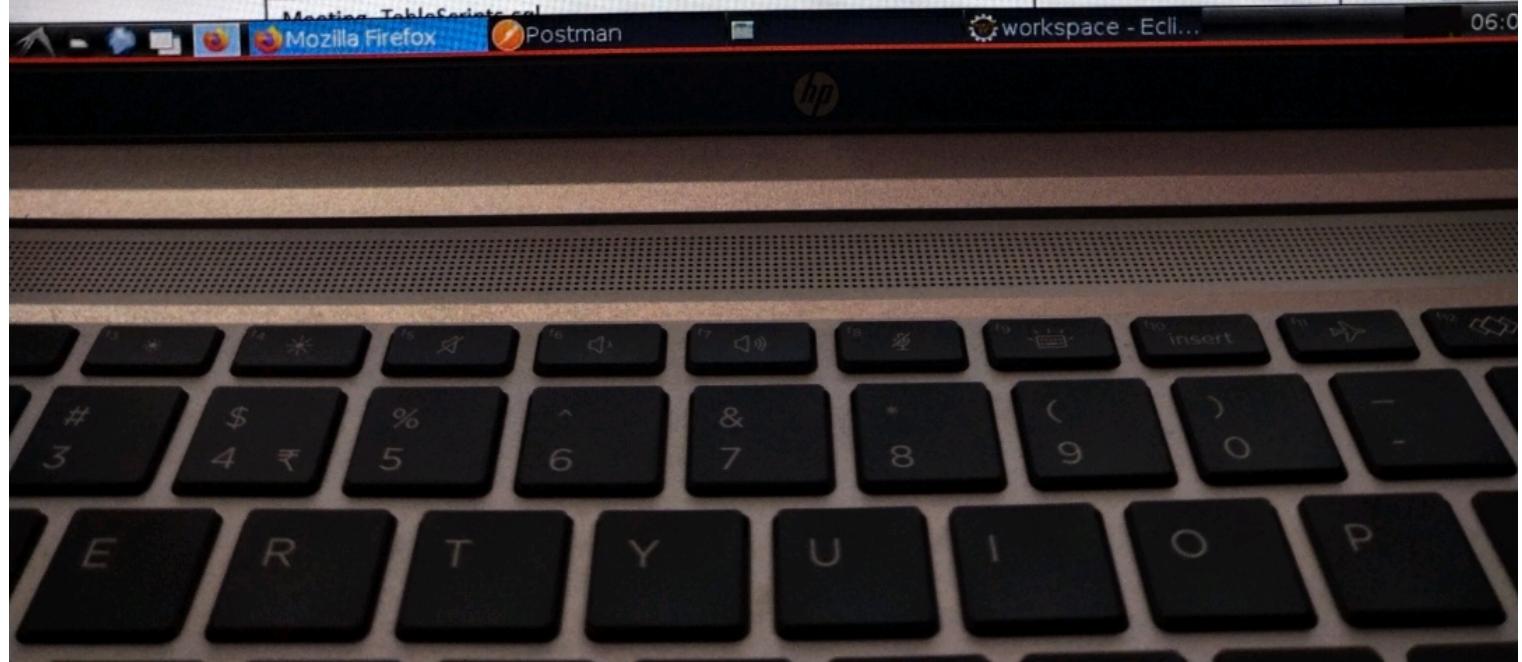
1. Schedule a meeting
2. Get all meetings of a user

Project Structure provided:

File Name	Status
application.properties	
com.infy.meetingscheduler.MeetingSchedulerApplication.java	
com.infy.meetingscheduler.exception.MeetingSchedulerException.java	
com.infy.meetingscheduler.service.MeetingService.java	
com.infy.meetingscheduler.utility.ErrorInfo.java	
log4j2.properties	
MeetingTableScript.sql	Implemented

Go to

06:00



ASSESSMENT INSTRUCTIONS	PROJECT INSTRUCTIONS	PROBLEM STATEMENT
log4j2.properties		
Meeting_TableScripts.sql		
ValidationMessages.properties		
com.infy.meetingscheduler.dto.MeetingDTO.java		
com.infy.meetingscheduler.entity.Meeting.java		To be modified
com.infy.meetingscheduler.utility.ExceptionControllerAdvice.java		
com.infy.meetingscheduler.api.MeetingAPI.java		
com.infy.meetingscheduler.repository.MeetingRepository.java		
com.infy.meetingscheduler.service.MeetingServiceImpl.java		To be implemented
com.infy.meetingscheduler.validator.MeetingValidator.java		

1. Entity Class

com.infy.meetingscheduler.entity.Meeting.java

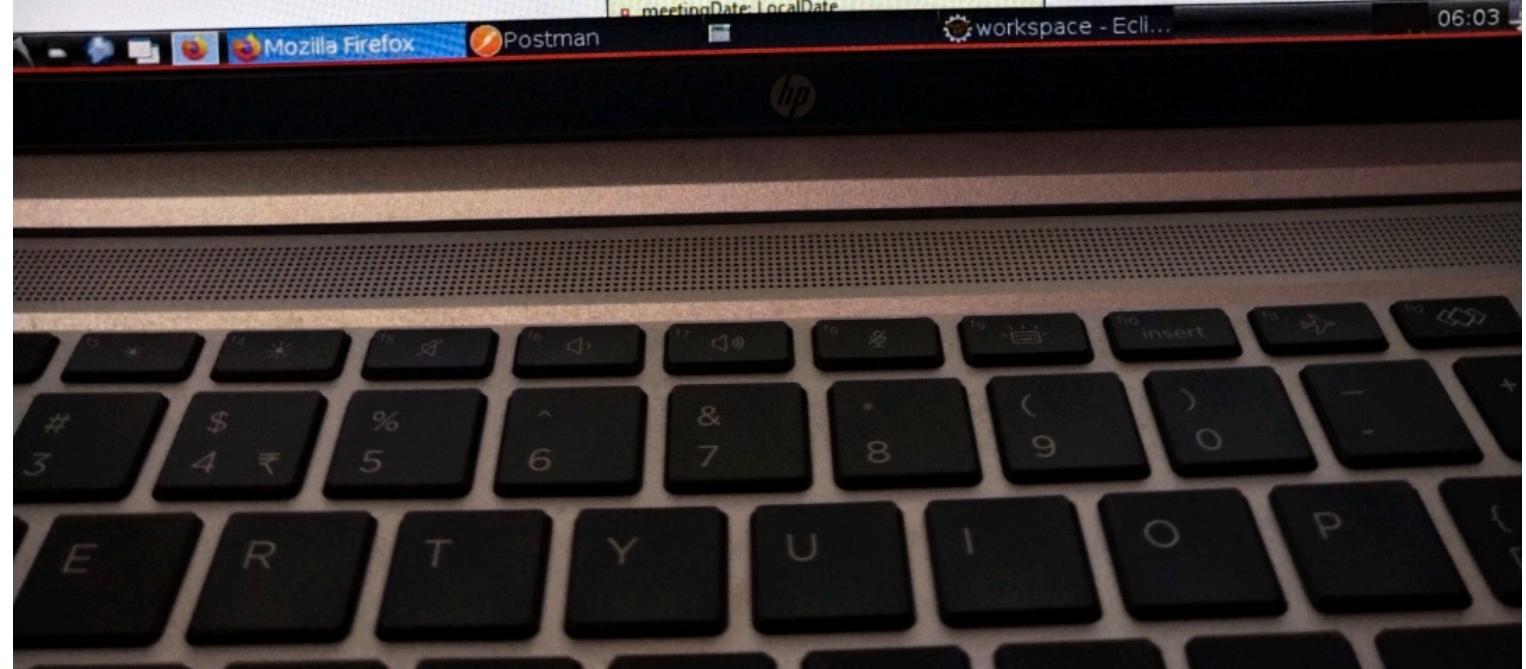
This class has been partially implemented for you.

- Annotate this class to declare it to be an entity class mapped to the **meeting** table with **meetingId** as the primary key
- Generate the **meetingId** using the **IDENTITY** strategy

```

com.infy.meetingscheduler.entity.Meeting
    □ meetingId: Integer
    □ schedulerName: String
    □ teamName: String
    □ purpose: String
    □ meetingDate: LocalDate

```



Mozilla Firefox

file:///home/user/Desktop/user_repo/QP/index.html#

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

1. Entity Class

com.infy.meetingscheduler.entity.Meeting.java

This class has been partially implemented for you.

- Annotate this class to declare it to be an entity class mapped to the **meeting** table with **meetingId** as the primary key
- Generate the **meetingId** using the **IDENTITY** strategy

```
com.infy.meetingscheduler.entity.Meeting
{
    □ meetingId: Integer
    □ schedulerName: String
    □ teamName: String
    □ purpose: String
    □ meetingDate: LocalDate

    ● getMeetingId(): Integer
    ● setMeetingId(meetingId: Integer): void
    ● getSchedulerName(): String
    ● setSchedulerName(schedulerName: String): void
    ● getTeamName(): String
    ● setTeamName(teamName: String): void
    ● getPurpose(): String
    ● setPurpose(purpose: String): void
    ● getMeetingDate(): LocalDate
    ● setMeetingDate(meetingDate: LocalDate): void
}
```

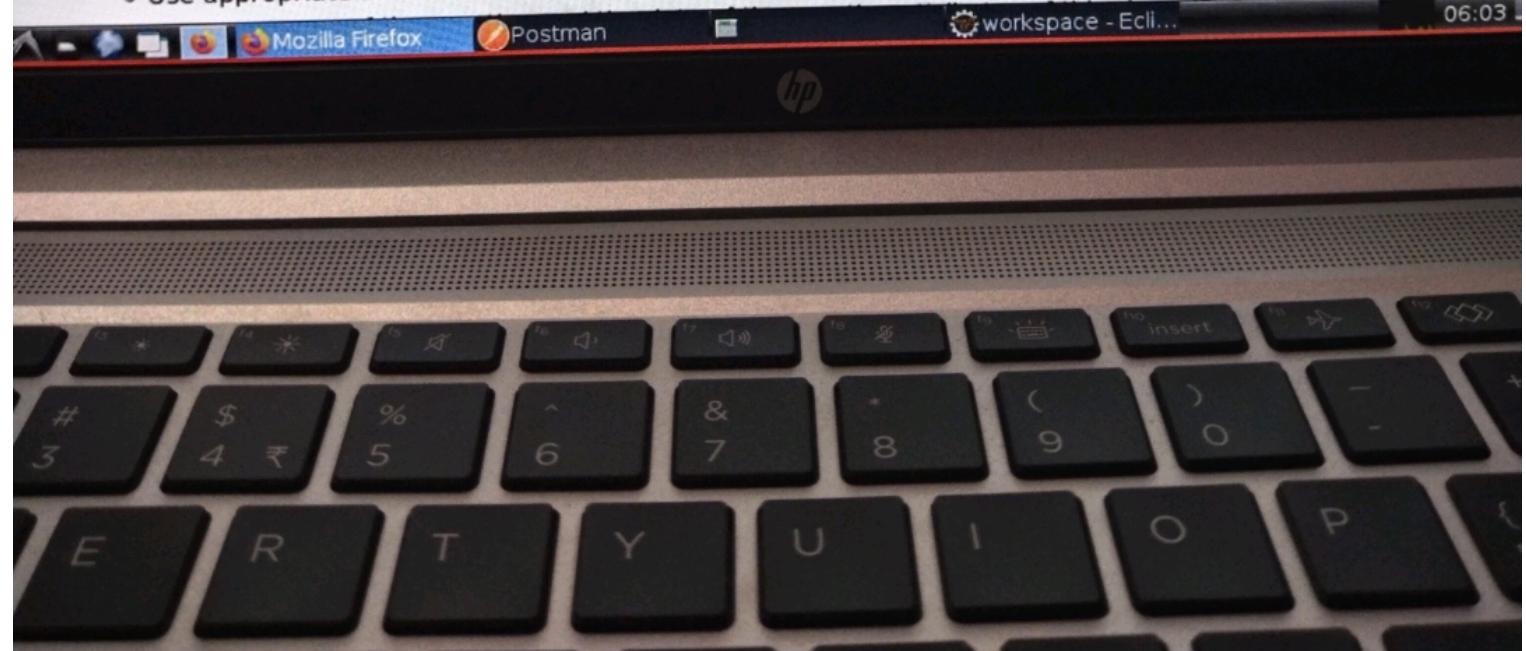
2. DTO Class

com.infy.meetingscheduler.dto.MeetingDTO.java

This is a DTO (Data Transfer Object) class that holds meeting details such as meeting scheduler name, team name, purpose of the meeting and the date of the meeting.

- Use appropriate annotations from the Java Bean Validation API for validating meeting scheduler name

[Go to t](#)



Mozilla Firefox

user/Desktop/user_re... X Firefox Privacy Notice - +

file:///home/user/Desktop/user_repo/QP/index.html# ... ☰ ☆

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

This is a DTO (Data Transfer Object) class that holds meeting details such as meeting scheduler name, team name, purpose of the meeting and the date of the meeting.

- o Use appropriate annotations from the Java Bean Validation API for validating meeting scheduler name, team name, purpose of the meeting and the date of the meeting attributes of this class. (Refer below scheduleMeeting() method of MeetingAPI for validation rules).

3. Validator Class

com.infy.meetingscheduler.MeetingValidator.java

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

```
com.infy.meetingscheduler.validator.MeetingValidator
MeetingValidator()
validateMeeting(meetingDTO: MeetingDTO): void
isValidTeamName(teamName: String): Boolean
isValidMeetingDate(meetingDate: LocalDate): Boolean
```

Note: The constructor has been implemented for you

Method Description:

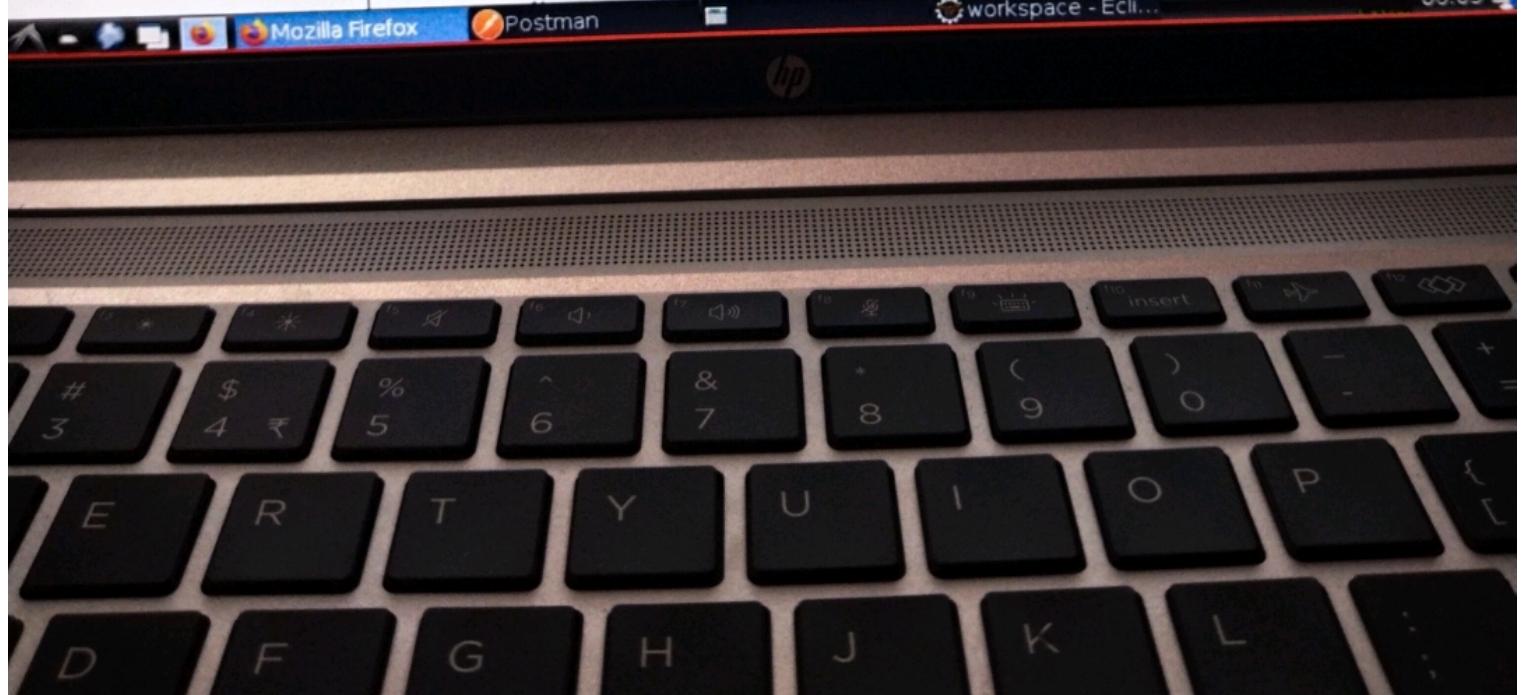
1. void validateMeeting(MeetingDTO meetingDTO)

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

Violation for	Message

Go to top

06:03



ASSESSMENT INSTRUCTIONS

PROJECT INSTRUCTIONS

PROBLEM STATEMENT

Method Description:

1. void validateMeeting(MeetingDTO meetingDTO)

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

Violation for	Message
teamName	MeetingValidator.INVALID_TEAM_NAME
meetingDate	MeetingValidator.INVALID_MEETING_DATE

2. Boolean isValidTeamName(String teamName)

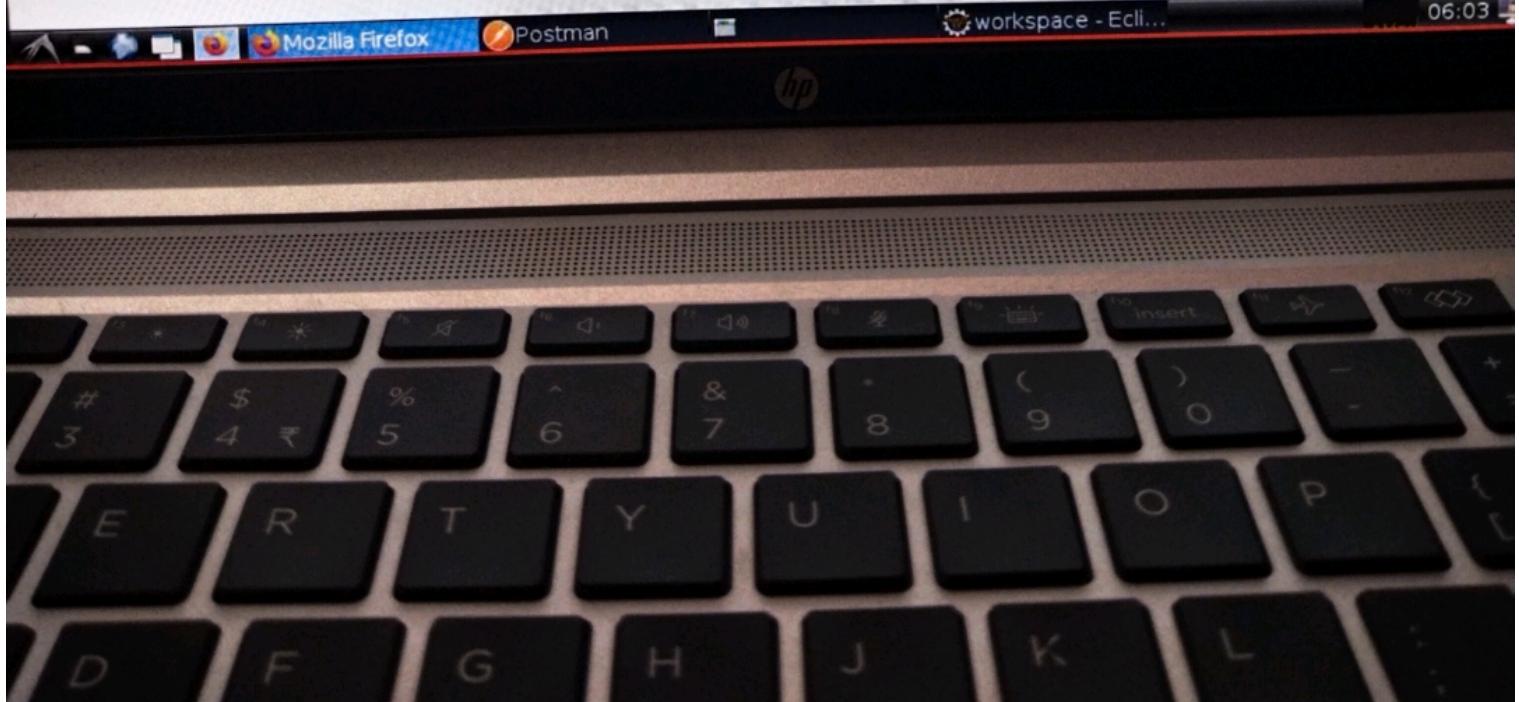
- This method is to validate the **teamName** received
- **teamName** should be one among the following, (perform case-sensitive check)

teamName
ETAMYSJAVA
ETAMYSUI
ETAMYSBI
ETAMYSMS
ETAMYSAI

- If the above condition fails return false, else return true

Go to top

06:03



Mozilla Firefox

/user/Desktop/user_re X /home/user/Desktop/user_re X + file:///home/user/Desktop/user_repo/QP/index.html# ... ☰

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

Method Description:

1. void validateMeeting(MeetingDTO meetingDTO)

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

Violation for	Message
teamName	MeetingValidator.INVALID_TEAM_NAME
meetingDate	MeetingValidator.INVALID_MEETING_DATE

2. Boolean isValidTeamName(String teamName)

- This method is to validate the **teamName** received
- **teamName** should be one among the following, (perform case-sensitive check)

teamName
ETAMYSJAVA
ETAMYSUI
ETAMYSBI
ETAMYSMS
ETAMYSAI

▪ If the above condition fails return false, else return true

Go to top

07:32

Firefox Postman workspace - Ecl... hp

E R T Y U I O P { [

Mozilla Firefox

/home/user/Desktop/user_re X /home/user/Desktop/user_re X +

file:///home/user/Desktop/user_repo/QP/index.html# ... ☰ ☆

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

▪ If the above condition fails return false, else return true

3. Boolean isValidMeetingDate(LocalDate meetingDate)

- This method validates the **meetingDate** received
- **meetingDate** should not be a **Saturday** or a **Sunday**
- If the above condition fails return false, else return true

4. Repository

com.infy.meetingscheduler.repository.MeetingRepository.java

- This interface should extend the appropriate repository interface
- Add the following three methods to the interface
 - The first method should accept the name of the scheduler and return all the meetings scheduled by the scheduler as **List<Meeting>**
 - Use **Query Creation based on method name approach** for this method
 - The second method should accept the name of the scheduler and the meeting date and return all the meetings scheduled by the scheduler for the given date as **List<Meeting>**
 - Use **Query Creation based on method name approach** for this method
 - The third method should accept the name of the team and the meeting date and return all the meetings scheduled for the team on the given date as **List<Meeting>**
 - Use **Query Creation based on method name approach** for this method

5. Service Class

com.infy.meetingscheduler.service.MeetingServiceImpl.java

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

com.infy.meetingscheduler.service.MeetingServiceImpl

MeetingRepository MeetingRepository

Postman workspace - Ecli... Go to top 07:32



ASSESSMENT INSTRUCTIONS

PROJECT INSTRUCTIONS

PROBLEM STATEMENT

com.infy.meetingscheduler.service.MeetingServiceImpl.java

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

```
com.infy.meetingscheduler.service.MeetingServiceImpl
```

- meetingRepository: MeetingRepository
- getAllMeetingsOfScheduler(schedulerName: String): List<MeetingDTO>
- scheduleMeeting(meetingDTO: MeetingDTO): MeetingDTO

- Inject the MeetingRepository object using the proper annotation
- Mark the class with the appropriate stereotype annotation with the value **meetingService**
- Mark this class with necessary annotation to manage transactions

Method Description:

1. List<MeetingDTO> getAllMeetingOfScheduler(String schedulerName)

- This method is used to get all the meetings of a scheduler based on the schedulerName
- Invoke the appropriate method of the MeetingRepository interface which will return all the meetings scheduled by the scheduler
- If no meeting details are available, throw an object of the **MeetingSchedulerException** with the message **MeetingService.NO_MEETINGS_FOUND**
- Else, populate a list, **List<MeetingDTO>** with all the meeting details in the ascending order of the meeting date and return the sorted list

Hint: You can use the prepareDTO() method of the MeetingDTO class to convert the object of Meeting to MeetingDTO

2. MeetingDTO scheduleMeeting(MeetingDTO meetingDTO)

- This method is used to schedule a meeting and add the meeting details to the database

Go to top

07:32

Ex



Mozilla Firefox

/user/Desktop/user_re... X /home/user/Desktop/user_re... X +
file:///home/user/Desktop/user_repo/QP/index.html# ... ☰ ☆

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

2. MeetingDTO scheduleMeeting(MeetingDTO meetingDTO)

- This method is used to schedule a meeting and add the meeting details to the database
- Invoke the validateMeeting() method of the MeetingValidator class to validate the object of the MeetingDTO class
- Invoke the appropriate method of the MeetingRepository interface which will return all the meetings scheduled by the scheduler for the given date
- If meeting details are available, throw an object of the **MeetingSchedulerException** with the message **MeetingService.MEETING_DATE_UNAVAILABLE**
- Else, invoke the appropriate method of the MeetingRepository interface which will return all the meetings scheduled for the team on the given date
- If meeting details for the team are available on the given date, throw an object of the **MeetingSchedulerException** with the message **MeetingService.TEAM_UNAVAILABLE**
- Else, create and populate a new object of **Meeting** with the appropriate values from the **MeetingDTO** object (Hint: You can use the prepareEntity() method of the MeetingDTO class for this)
- Then invoke the appropriate method of the MeetingRepository to add the **Meeting** object to the database which will return an object of **Meeting** with the generated value of **meetingId**
- Set the **meetingId** of the existing **MeetingDTO** object using the **Meeting** object obtained in the previous step.
- Return the modified **MeetingDTO** object

6. API Class

com.infy.meetingscheduler.api.MeetingAPI.java

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

Go to top

07:33

com.infy.meetingscheduler.api.MeetingAPI

workspace - Ecli...

hp



/user/Desktop/user_re X

/home/user/Desktop/user_re X

+

file:///home/user/Desktop/user_repo/QP/index.html#

... ☰ ☆



ASSESSMENT INSTRUCTIONS

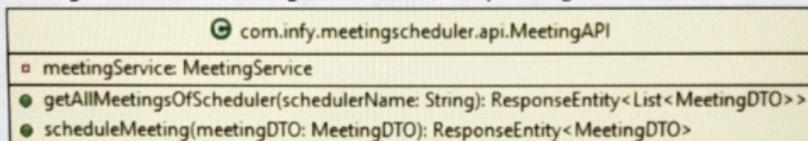
PROJECT INSTRUCTIONS

PROBLEM STATEMENT

6. API Class

com.infy.meetingscheduler.api.MeetingAPI.java

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

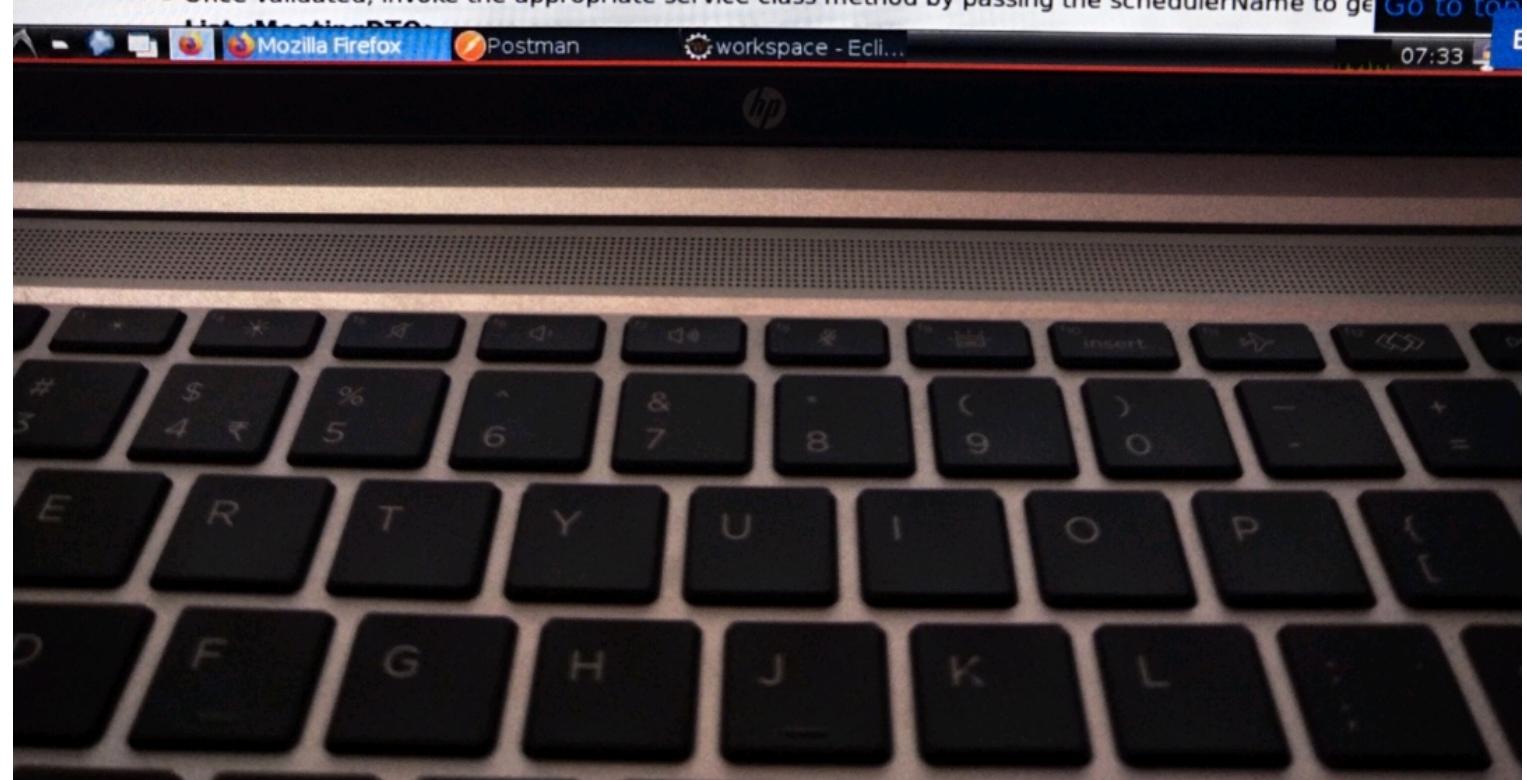


- Annotate this class with appropriate annotation to declare it as REST controller class
- Annotate this class with appropriate annotation so that all its methods are mapped with **api** as base URL
- Annotate this class with appropriate annotation to enable the bean validation support
- Inject MeetingService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

Method Description:

1. **ResponseEntity<List<MeetingDTO>> getAllMeetingOfScheduler (String schedulerName)**

- This method is to retrieve all the meetings of the given scheduler
- It accepts **GET** request with URI **meetings/{schedulerName}**
- The path variable schedulerName should be validated as below (use **Bean Validation API** for validation),
 - **schedulerName** should consist of letters of the English alphabet with a single whitespace character between two words (it can have multiple words). If invalid, set **Scheduler name should have only letters and spaces** 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 schedulerName to get



Mozilla Firefox

/home/user/Desktop/user_re X /home/user/Desktop/user_re X +

file:///home/user/Desktop/user_repo/QP/index.html# ... ☰ ☆

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

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

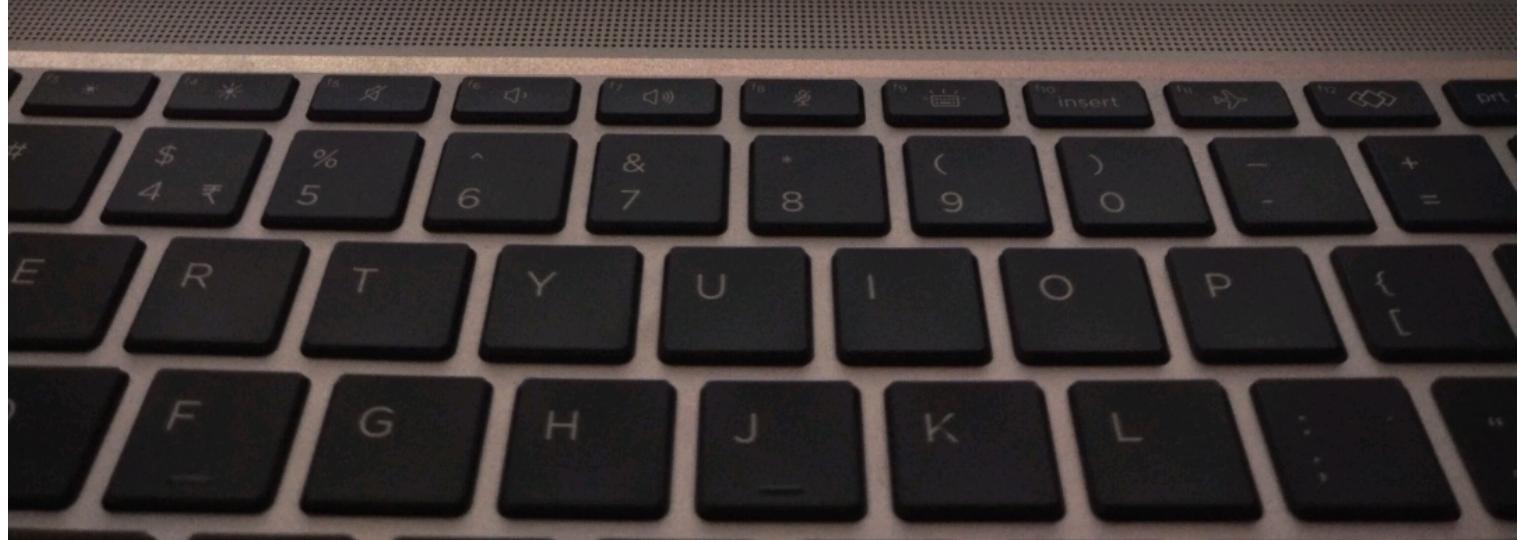
2. public ResponseEntity<MeetingDTO> scheduleMeeting (MeetingDTO meetingDTO)

- This method schedules a meeting in the MeetingScheduler application
- It accepts **POST** request with URI **meetings**
- It receives meeting details in the form of JSON. The JSON data should be populated in **MeetingDTO** parameter of this method after successful validation.
- The meeting details will consist of the scheduler name, team name, purpose and meeting date. 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),
 - schedulerName:**
If schedulerName is not provided then set **Scheduler name is mandatory** as the validation message
If it is provided, it should consist of letters of the English alphabet with a single whitespace character between two words (it can have multiple words). If invalid, set **Scheduler name should have only letters and spaces** as the validation message
 - teamName:**
If teamName is not provided then set **Team name is mandatory** as the validation message
 - purpose:**
If purpose is not provided then set **Purpose is mandatory** as the validation message
 - meetingDate:**
If meetingDate is not provided then set **Meeting date is mandatory** as the validation message

Go to top

07:33 Exec

Mozilla Firefox Postman workspace - Ecli... hp



Mozilla Firefox

file:///home/user/Desktop/user_repo/QP/index.html#

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

- **purpose:**
If purpose is not provided then set **Purpose is mandatory** as the validation message
- **meetingDate:**
If meetingDate is not provided then set **Meeting date is mandatory** as the validation message
If it is provided, it should not be a past date. If invalid, set **Meeting date should not be a past date** as the validation message
- Then invoke the appropriate service class method to schedule the meeting by passing the **MeetingDTO** object which will return a **MeetingDTO** object with the **meetingId** populated.
- Return an object of **ResponseEntity** with the MeetingDTO object obtained in the previous step as the response and the HTTP response status as **CREATED**.

7. Utility Class

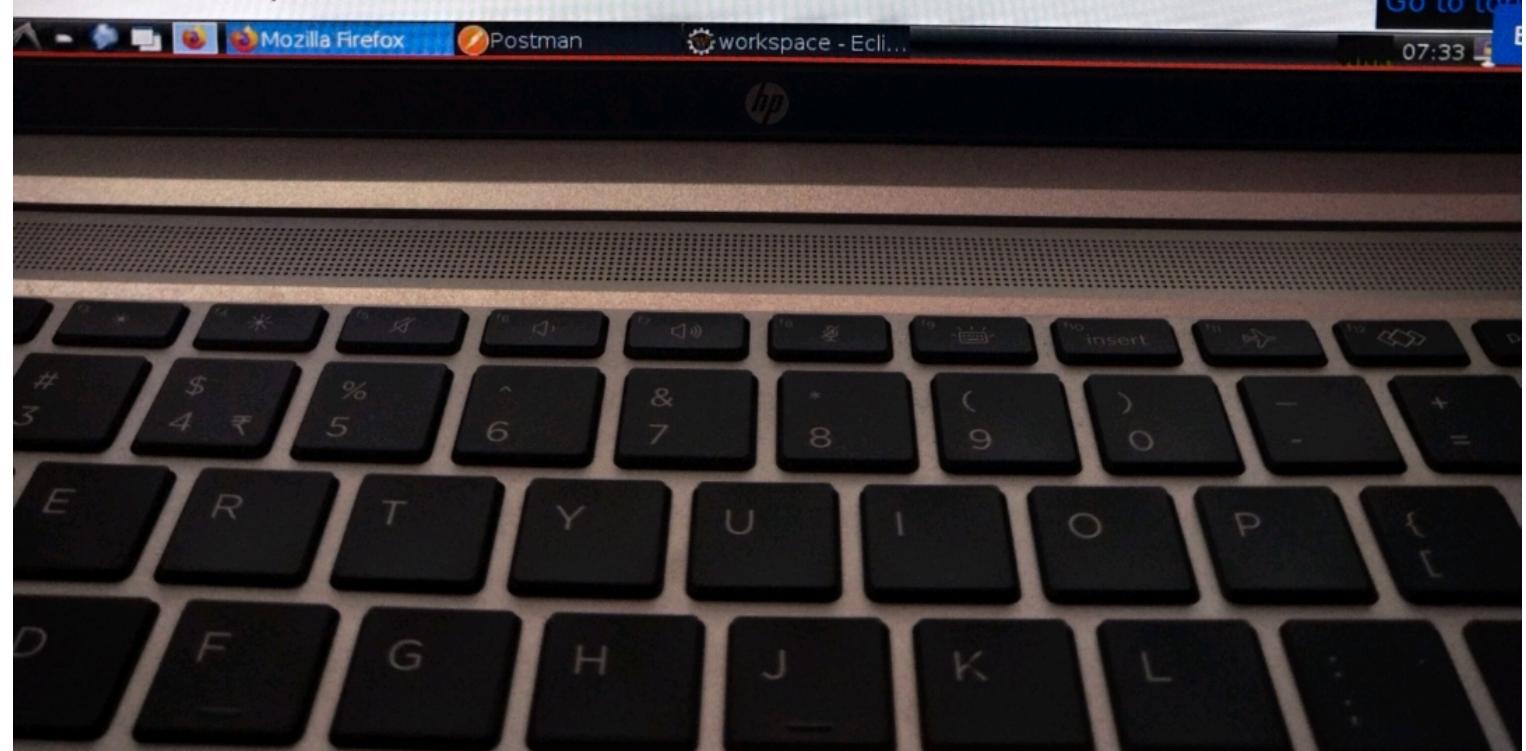
com.infy.meetingscheduler.utility.ExceptionControllerAdvice.java

Implement the ExceptionControllerAdvice to log and handle all the exceptions according to description given below,

com.infy.meetingscheduler.utility.ExceptionControllerAdvice
▪ LOGGER: Log
▪ environment: Environment
● meetingSchedulerExceptionHandler(exception: MeetingSchedulerException): ResponseEntity<ErrorInfo>
● generalExceptionHandler(exception: Exception): ResponseEntity<ErrorInfo>
● validatorExceptionHandler(exception: Exception): ResponseEntity<ErrorInfo>

- Annotate the class with appropriate annotation to declare it as central exception handler class
- Inject Environment object using appropriate annotation
- The body of the methods is already implemented for you

Method Description:





ASSESSMENT INSTRUCTIONS

PROJECT INSTRUCTIONS

PROBLEM STATEMENT

Method Description:

1. **ResponseType<ErrorInfo> meetingSchedulerExceptionHandler(MeetingSchedulerException exception)**
 - Annotate this method for handling **MeetingSchedulerException** exception

2. **ResponseType<ErrorInfo> generalExceptionHandler(Exception exception)**
 - Annotate this method for handling **Exception** exception

3. **ResponseType<ErrorInfo> validatorExceptionHandler(Exception exception)**
 - Annotate this method for handling **MethodArgumentNotValidException** and **ConstraintViolationException** exceptions



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

Note: The date values have been used considering the present date (today) to be 9th July, 2020. Please change the dates appropriately to test your implementation.

REST end point	Request Type(valid/invalid)	End point URI with sample input	Sample Response
Get all the	Valid test data	api/meetings/Scott Adams	[{

Go to top

07:33



Assessment Instructions				Project Instructions	Problem Statement
REST end point	Request Type(valid/invalid)	End point URI with sample input	Sample Response		
Get all the meetings of a scheduler	Valid test data	api/meetings/Scott Adams	<pre>[[{ "meetingId": 1, "schedulerName": "Scott Adams", "teamName": "ETAMYSJAVA", "purpose": "Discuss the new tool", "meetingDate": "2020-07-11" }]]</pre>		
	Invalid test data 1	api/meetings/123	<pre>{ "errorMessage": "Scheduler name should have only letters and spaces", "errorCode": 400 }</pre>		
	Invalid test data 2	api/meetings/xyz	<pre>{ "errorMessage": "No meetings were found for the given scheduler", "errorCode": 400 }</pre>		

Project Instructions					
ASSESSMENT INSTRUCTIONS		PROJECT INSTRUCTIONS		PROBLEM STATEMENT	
Invalid test data 2		api/meetings/xyz		<pre>{ "errorMessage": "No meetings were found for the given scheduler", "errorCode": 400 }</pre>	
Schedule a meeting		api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie Spencer", "teamName": "ETAMYSJAVA", "purpose": "Welcome new members to the team", "meetingDate": "2020-07-15" }</pre>		<pre>{ "meetingId": 6, "schedulerName": "Jamie Spencer", "teamName": "ETAMYSJAVA", "purpose": "Welcome new members to the team", "meetingDate": "2020-07-15" }</pre>	
Invalid test data 1		api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie123", "teamName": "ETAMYSJAVA", "meetingDate": "2020-07-10" }</pre>		<pre>{ "errorMessage": "Scheduler name should have only letters and spaces, Purpose is mandatory", "errorCode": 400 }</pre>	

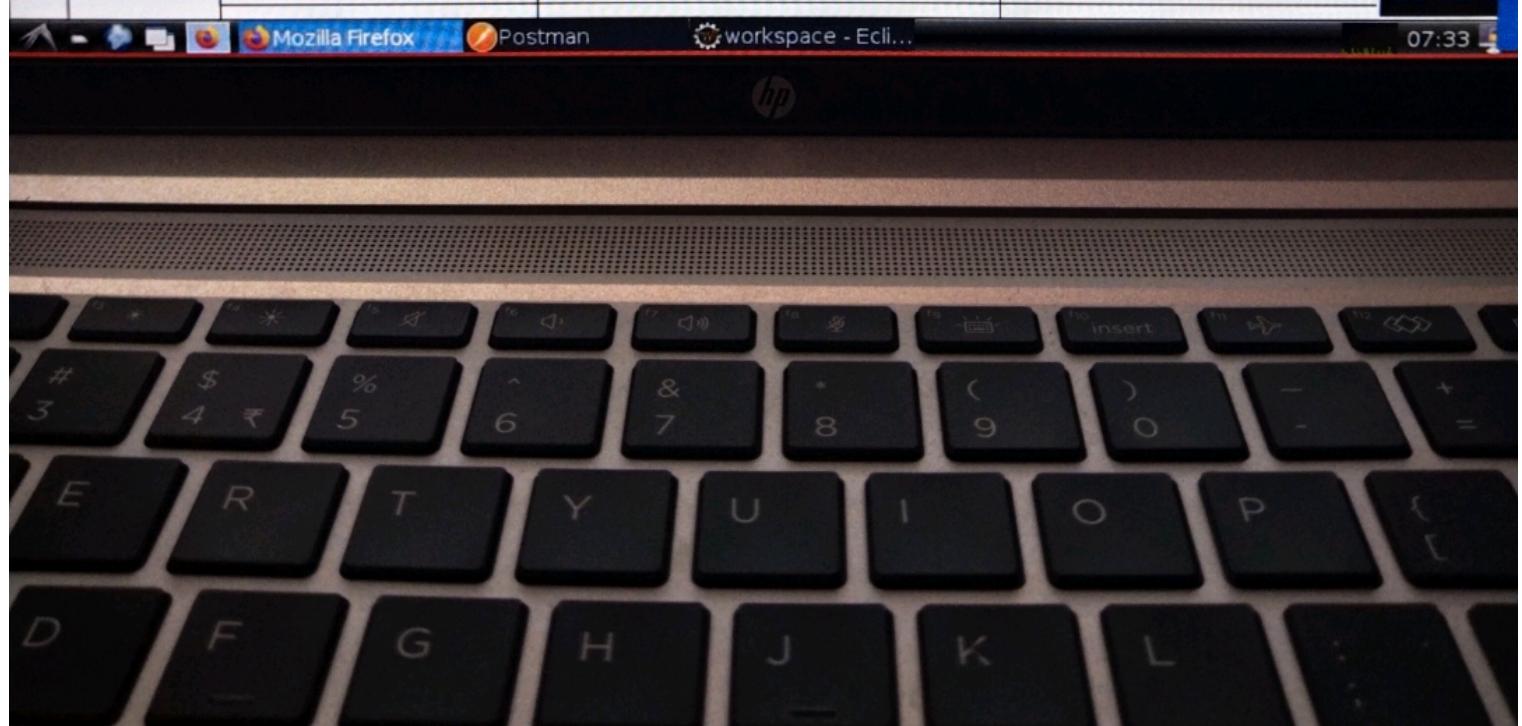
Mozilla Firefox

/home/user/Desktop/user_repo X /home/user/Desktop/user_repo X +
file:///home/user/Desktop/user_repo/QP/index.html# ... ☰ ☆ ⌂ ⌃ ⌄ ⌅

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

	Invalid test data 2	api/meetings/xyz	<pre>{ "errorMessage": "No meetings were found for the given scheduler", "errorCode": 400 }</pre>
Schedule a meeting	Valid test data	api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie Spencer", "teamName": "ETAMYSJAVA", "purpose": "Welcome new members to the team", "meetingDate": "2020-07-15" }</pre>	<pre>{ "meetingId": 6, "schedulerName": "Jamie Spencer", "teamName": "ETAMYSJAVA", "purpose": "Welcome new members to the team", "meetingDate": "2020-07-15" }</pre>
	Invalid test data 1	api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie123", "teamName": "ETAMYSJAVA", "meetingDate": "2020-07-10" }</pre>	<pre>{ "errorMessage": "Scheduler name should have only letters and spaces, Purpose is mandatory", "errorCode": 400 }</pre>

Go to top



Mozilla Firefox

/home/user/Desktop/user_repo/QP/index.html#

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

Invalid test data 2	api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie Spencer", "teamName": "ETAMYS", "purpose": "Welcome new members to the team", "meetingDate": "2020-07-15" }</pre>	<pre>{ "errorMessage": "Team name should be one among ETAMYSJAVA, ETAMYSMS, ETAMYSBI, ETAMYSUI, ETAMYSAI", "errorCode": 400 }</pre>
Invalid test data 3	api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie Spencer", "teamName": "ETAMYSJAVA", "purpose": "Welcome new members to the team", "meetingDate": "2020-07-11" }</pre>	<pre>{ "errorMessage": "Meeting date cannot be a Saturday or a Sunday", "errorCode": 400 }</pre>
Invalid test data 4	api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie Spencer", "teamName": "ETAMYSJAVA", "purpose": "Welcome new members to the team", "meetingDate": "2020-07-11" }</pre>	<pre>{ "errorMessage": "The scheduler has a meeting on the requested date, please choose another date" }</pre>

Mozilla Firefox Postman workspace - Ecli... Go to top 07:33



Mozilla Firefox

/home/user/Desktop/user_re... X /home/user/Desktop/user_re... X +
file:///home/user/Desktop/user_repo/QP/index.html#

ASSESSMENT INSTRUCTIONS PROJECT INSTRUCTIONS PROBLEM STATEMENT

	Invalid test data 4	api/meetings Sample JSON: <pre>{ "schedulerName": "Scott Adams", "teamName": "ETAMYSUI", "purpose": "Welcome new memebers to the team", "meetingDate": "2020-07-11" }</pre>	<pre>{ "errorMessage": "The scheduler has a meeting on the requested date, please choose any other date", "errorCode": 400 }</pre>
	Invalid test data 5	api/meetings Sample JSON: <pre>{ "schedulerName": "Jamie Spencer", "teamName": "ETAMYSJAVA", "purpose": "Welcome new memebers to the team", "meetingDate": "2020-07-11" }</pre>	<pre>{ "errorMessage": "The team has a meeting on the requested date, please choose any other date", "errorCode": 400 }</pre>

PREV

Go to top

07:34

