

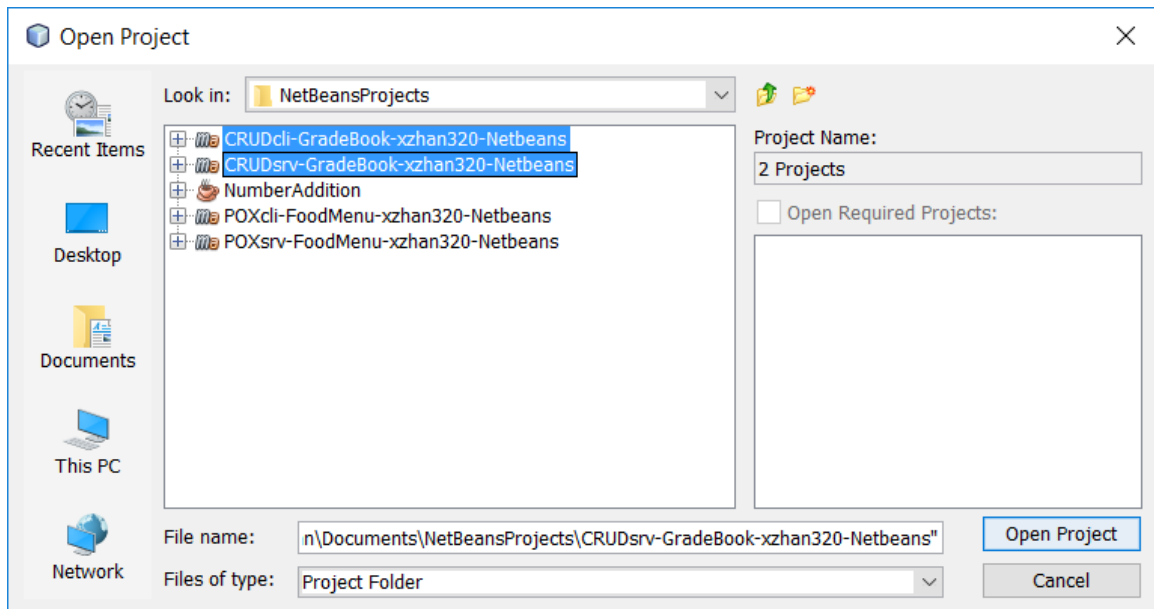
CSE 564 Project 2 CRUD Assignment

Xiaoyu Zhang

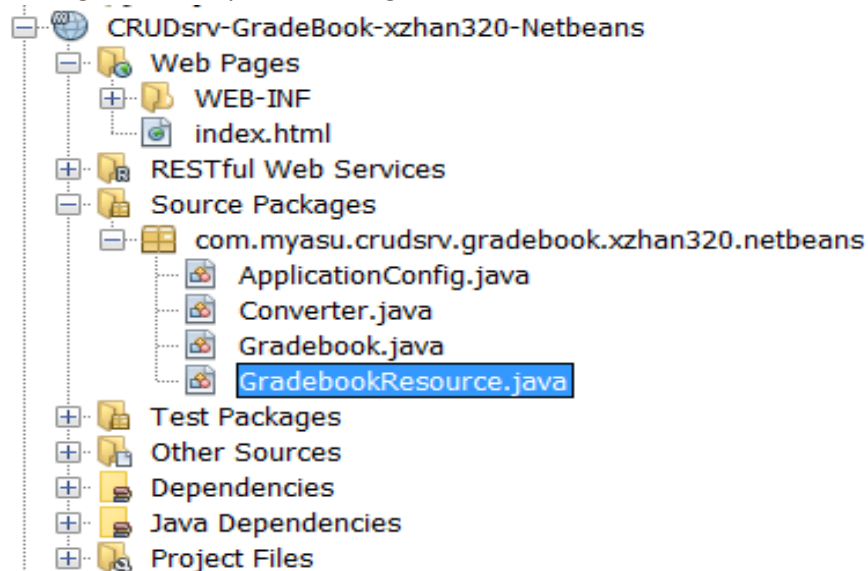
1207639026


ReadMe

1. Download and unzip the package **CRUD-GradeBook-xzhan320-Netbeans**
2. Run NetBeans IDE and click File -> Open Project. On folder selection page, open both **CRUDsrv-GradeBook-xzhan320-Netbeans** and **CRUDcli-GradeBook-xzhan320-Netbeans**.




3. Direct to the server folder (CRUDsrv-GradeBook-xzhan320-Netbeans) and open **GradebookResource.java** file under CRUDsrv-GradeBook-xzhan320-Netbeans – Source Packages – com.myasu.crudsrv.gradebook.xzhan320.netbeans – GradebookResource.java

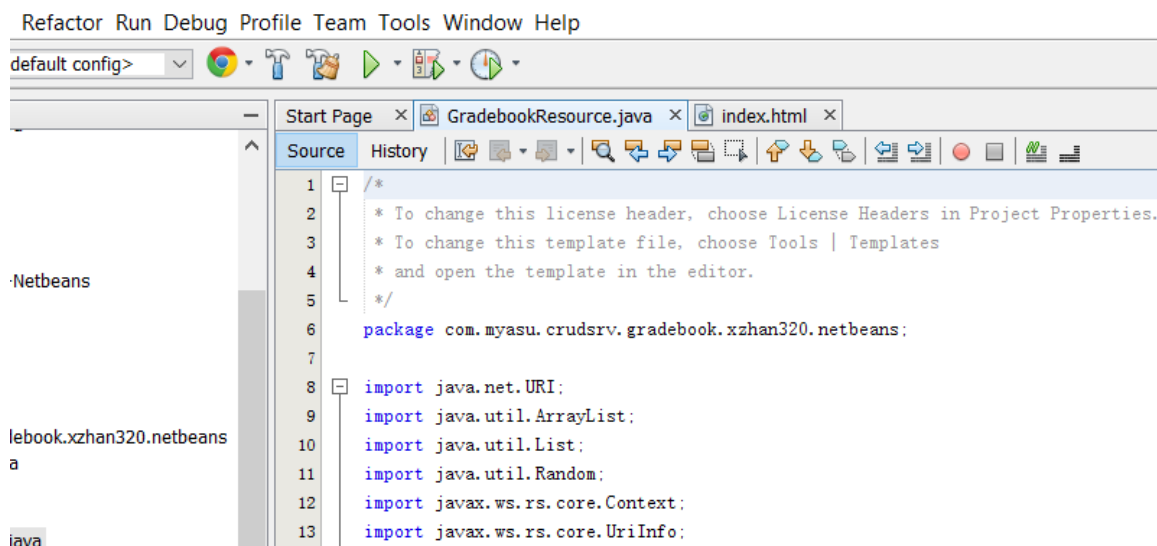


4. Clean build the server program by clicking the grey hammer button  on the control panel.

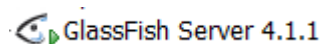
(Display below the step 5)

5. Run the server program by clicking the green arrow button  on the top right of the control panel.

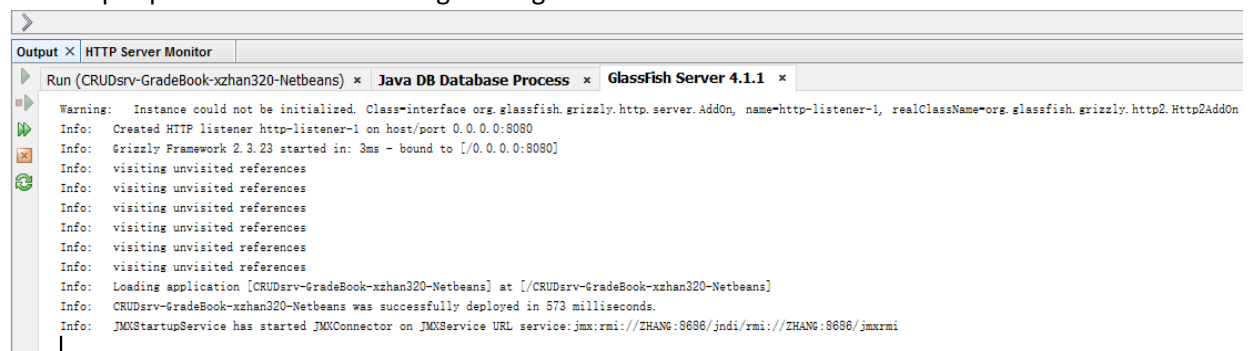
0-Netbeans - NetBeans IDE 8.1



6. When prompt, choose **GlassFish Server 4.1.1** for deployment.



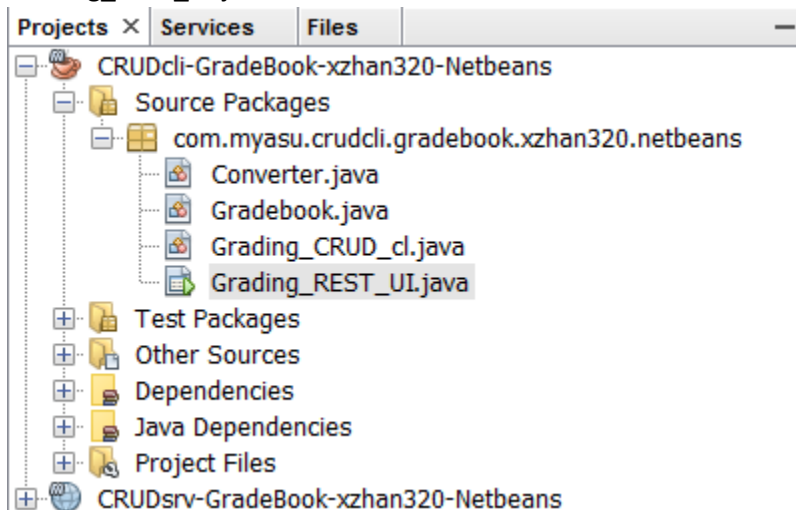
7. The output panel will show following message.





And a sample web page will pop up.



8. Now go back to Netbeans and direct to the client folder (CRUDcli-GradeBook-xzhan320-Netbeans) and open **Grading_REST_UI.java** file under CRUDcli-GradeBook-xzhan320-Netbeans – Source Packages – com.myasu.crudcli.gradebook.xzhan320.netbeans – Grading_REST_UI.java

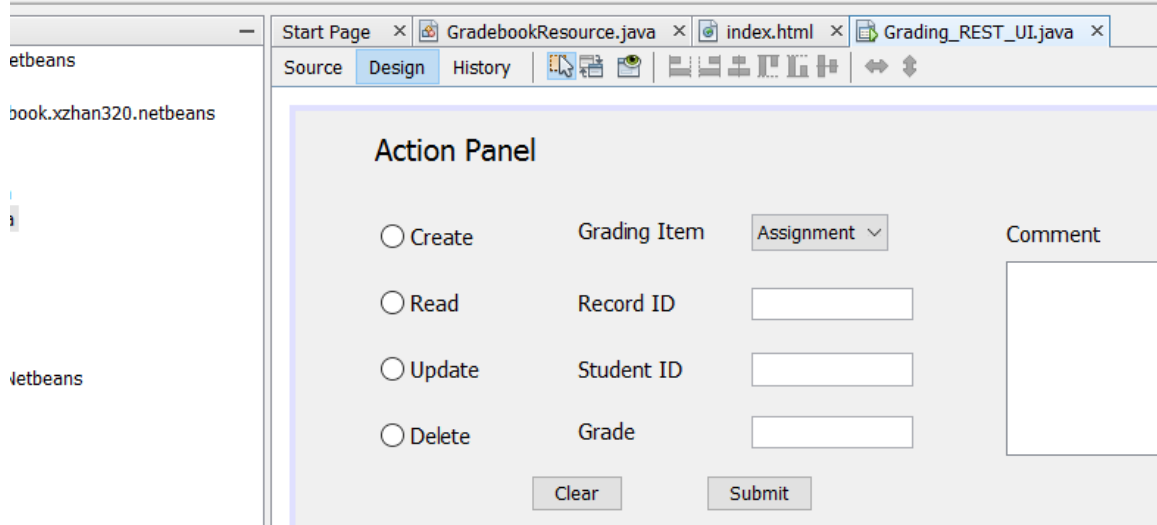


9. Once again, clean build and then run the client program by clicking the  and  button on the top right of the control panel.

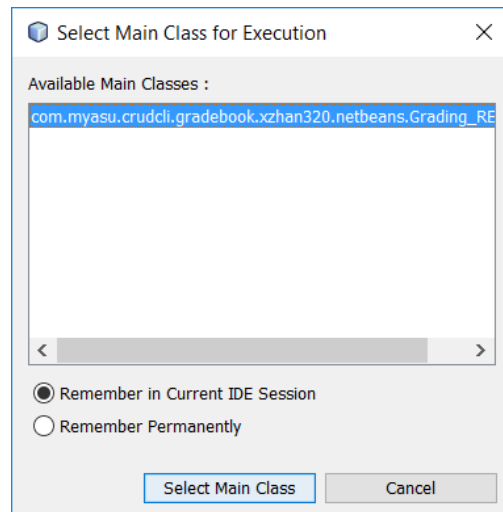
-Netbeans - NetBeans IDE 8.1

Refactor Run Debug Profile Team Tools Window Help

default config>       



Choose com.myasu.crudcli.gradebook.xzhan320.netbeans.Grading_REST_UI as Main Class.



10. Now, you will see a GUI window pop up with following components:

11. The following steps will give you a hands-on operation guide on how to interact with **CRUD** server system to store/retrieve/update/delete a valid object.

12. First, let's try to **CREATE** a student grade record as shown below:

Gradebook system

Action Panel

☒ Create Grading Item: Assignment

☐ Read Record ID:

☐ Update Student ID:

☐ Delete Grade:

Comment

No comment (-5), Case 5 failure (-5)

HTTP Header Info

HTTP Status Code: Media Type:

Location:

13. Now, click "Submit" will yield the following feedback info:

Gradebook system

Action Panel

☒ Create Grading Item: Assignment

☐ Read Record ID:

☐ Update Student ID:

☐ Delete Grade:

Comment

No comment (-5), Case 5 failure (-5)

HTTP Header Info

Last Operation: CREATE - Record ID 277 HTTP Status: Created

HTTP Status Code: Media Type:

Location:

(Note: When creating the first instance of the gradebook, the CRUD server will randomly generate a record ID attached to the gradebook entries. The reason why student ID is no

longer a primary key is that for each student ID the gradebook may have up to 4 grading records (Assignment, mid-term, quiz, class labs) associate with it. **Thus, the student ID alone is not unique identifier but the combination of (Student ID, Grading Item) is.**

14. Now, let's try to **READ** the record we just created. Click "Clear" to make panel tidy and type "277" in the Record ID text field.

The screenshot shows a web application window titled "Gradebook system". It features an "Action Panel" with four radio buttons: "Create", "Read" (which is selected), "Update", and "Delete". To the right of these buttons are input fields for "Grading Item" (a dropdown menu showing "Assignment"), "Record ID" (containing "277"), "Student ID" (empty), and "Grade" (empty). Below these fields are "Clear" and "Submit" buttons. To the right of the "Record ID" field is a large text area labeled "Comment". Below the "Action Panel" is a section titled "HTTP Header Info" with the text "Last Operation: CREATE - Record ID 277 HTTP Status: Created". Below this are input fields for "HTTP Status Code", "Media Type", and "Location".

15. Now, click "Submit" will yield the following feedback info:

The screenshot shows the same "Gradebook system" window. In the "Action Panel", the "Read" radio button is still selected. The "Record ID" field still contains "277". The "Student ID" field now contains "001" and the "Grade" field contains "90.0". The "Submit" button is now highlighted with a blue border. The "Comment" text area now contains the text "No comment (-5), Case 5 failure (-5)". The "HTTP Header Info" section now shows "Last Operation: READ - Record ID 277 HTTP Status: OK". The "HTTP Status Code" field now contains "200" and the "Media Type" field contains "application/xml". The "Location" field remains empty.

(Note: The Location field is only populated when a Resource is created)

16. Now, let's try to **UPDATE** the record we have created.

The screenshot shows a web application window titled "Gradebook system". It features an "Action Panel" with four radio buttons: "Create", "Read", "Update" (which is selected), and "Delete". To the right of these buttons are input fields for "Grading Item" (a dropdown menu showing "Assignment"), "Record ID" (containing "277"), "Student ID" (containing "001"), and "Grade" (containing "95"). A "Comment" text area contains the text "Case 5 failure (-5)". Below the input fields are "Clear" and "Submit" buttons. At the bottom, there is a section for "HTTP Header Info" with the text "Last Operation: READ - Record ID 277 HTTP Status: OK". Below this are input fields for "HTTP Status Code" (containing "200"), "Media Type" (containing "application/xml"), and "Location" (an empty field).

17. Now, click "Submit" will yield the following feedback info:

This screenshot is identical to the previous one, but the "Submit" button is now highlighted with a blue border, indicating it has been clicked. The "HTTP Header Info" section now displays "Last Operation: UPDATE - Record ID 277 HTTP Status: OK". Additionally, the "Grade" input field now contains the value "95.0" instead of "95".

18. Now, let's try to **DELETE** the record we have created.

Gradebook system

Action Panel

☐ Create Grading Item Assignment Comment

☐ Read Record ID 277 Case 5 failure (-5)

☐ Update Student ID 001

☒ Delete Grade 95.0

Clear Submit

HTTP Header Info Last Operation: UPDATE - Record ID 277 HTTP Status: OK

HTTP Status Code 200 Media Type application/xml

Location

19. Now, click "Submit" will yield the following feedback info:

Gradebook system

Action Panel

☐ Create Grading Item Assignment Comment

☐ Read Record ID 277

☐ Update Student ID

☒ Delete Grade

Clear Submit

HTTP Header Info Last Operation: DELETE - Record ID 277 HTTP Status: No Content

HTTP Status Code 204 Media Type null

Location

20. The following steps will give you in-depth guide on how CRUD server system detect and handle invalid inputs.

21. Error request handling 1: Delete a record that does not exist.

Let's try to **DELETE** the same above record once again. Click "Submit" will yield the following feedback info:

The screenshot shows a web browser window titled "Gradebook system". Inside, there is an "Action Panel" with four radio buttons: "Create", "Read", "Update", and "Delete". The "Delete" button is selected. To the right of the radio buttons are input fields for "Grading Item" (a dropdown menu showing "Assignment"), "Record ID" (containing "277"), "Student ID" (empty), and "Grade" (empty). There is also a "Comment" text area. Below these fields are "Clear" and "Submit" buttons. At the bottom, there is a section titled "HTTP Header Info" with the text "Last Operation: DELETE - Record ID 277 HTTP Status: Gone". Below this are input fields for "HTTP Status Code" (containing "410"), "Media Type" (containing "text/html"), and "Location" (empty).

22. Error request handling 2: Read a record that does not exist.

Let's try to **READ** a record that has been deleted. Click "Submit" will yield the following feedback info:

The screenshot shows the same "Gradebook system" web browser window. In the "Action Panel", the "Read" radio button is now selected. The "Record ID" field still contains "277". The "HTTP Header Info" section now displays "Last Operation: READ - Record ID 277 HTTP Status: Not Found". The "HTTP Status Code" field contains "404" and the "Media Type" field contains "application/xml".

23. Error request handling 3: Update a record that does not exist.

Let's try to **UPDATE** a record that does not exist. Click "Submit" will yield the following feedback info:

The screenshot shows the 'Gradebook system' window. In the 'Action Panel', the 'Update' radio button is selected. The 'Grading Item' is set to 'Assignment'. The 'Record ID' is 278. The 'Student ID' and 'Grade' fields are empty. The 'Comment' field is empty. The 'Submit' button is highlighted. Below the form, the 'HTTP Header Info' section shows 'Last Operation: UPDATE - Record ID 278' and 'HTTP Status: Conflict'. The 'HTTP Status Code' is 409 and the 'Media Type' is application/xml. The 'Location' field is empty.

Note: The Conflict status is determined and returned by the code module below:

```
else {  
    LOG.info("Creating a {} {} Status Response", Response.Status.CONFLICT.getStatusCode(), Response.Status.CONFLICT.getReasonPhrase());  
    LOG.debug("Cannot update Gradebook Resource {} as it has not yet been created", content);  
  
    response = Response.status(Response.Status.CONFLICT).entity(content).build();  
}
```

24. Error request handling 4: Create duplicate records.

First, let's try to create a new record as shown:

The screenshot shows the 'Gradebook system' window. In the 'Action Panel', the 'Create' radio button is selected. The 'Grading Item' is set to 'Assignment'. The 'Record ID' is 278. The 'Student ID' is 002. The 'Grade' is 100.0. The 'Comment' field contains 'Well done!'. The 'Submit' button is highlighted. Below the form, the 'HTTP Header Info' section shows 'Last Operation: CREATE - Record ID 278' and 'HTTP Status: Created'. The 'HTTP Status Code' is 201 and the 'Media Type' is application/xml. The 'Location' field contains the URL 'http://localhost:8080/CRUDsrv-GradeBook-xzhan320-Netbeans/webresources/Gradebook/278'.

Then, if we click “Submit” once again, we will receive a Conflict Status:

The screenshot shows the 'Gradebook system' window. The 'Action Panel' has the 'Create' radio button selected. The 'Grading Item' dropdown is set to 'Assignment'. The 'Record ID' field contains '278'. The 'Student ID' and 'Grade' fields are empty. The 'Comment' field is empty. Below the form are 'Clear' and 'Submit' buttons. The 'HTTP Header Info' section displays: 'Last Operation: CREATE - Record ID 278 HTTP Status: Conflict'. The 'HTTP Status Code' is '409' and the 'Media Type' is 'application/xml'. The 'Location' field is empty.

However, if we change the “Grading Item” with everything else remaining unchanged, the CRUD server will return no bad status. Instead, a new record ID will be assigned for the same student ID with different grading items:

The screenshot shows the 'Gradebook system' window. The 'Action Panel' has the 'Create' radio button selected. The 'Grading Item' dropdown is set to 'Quiz'. The 'Record ID' field contains '279'. The 'Student ID' field contains '002'. The 'Grade' field contains '100.0'. The 'Comment' field contains 'Well done!'. Below the form are 'Clear' and 'Submit' buttons. The 'HTTP Header Info' section displays: 'Last Operation: CREATE - Record ID 279 HTTP Status: Created'. The 'HTTP Status Code' is '201' and the 'Media Type' is 'application/xml'. The 'Location' field contains the URL: 'http://localhost:8080/CRUDsrv-GradeBook-xzhan320-Netbeans/webresources/Gradebook/279'.

The standard of CRUD server on collision inspection is carried out by the if-statement below:

```
// Collision Inspection
Gradebook curr_grdbook;
for(int index = 0; index < gradebookList.size(); index++)
{
    curr_grdbook = gradebookList.get(index);
    if(curr_grdbook.getStudentId().equals(gradebook.getStudentId()) && curr_grdbook.getGradingItem().equals(gradebook.getGradingItem()))
    {
        // Collision Found
    }
}
```