

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

----oOo----



Báo cáo bài tập số 3

SERVLET & JSP

|Giáo viên hướng dẫn|

Trương Phước Lộc

Học phần: Lập trình ứng dụng Java

Thành phố Hồ Chí Minh, tháng 5 năm 2024

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

----oOo----



Báo cáo bài tập số 3

SERVLET & JSP

Sinh viên thực hiện:

Họ tên: Lê Xuân Huy

MSSV: 20120495

Chuyên ngành: Công nghệ phần mềm

SĐT: 0348324331

Niên học: 2020-2024

Thành phố Hồ Chí Minh, tháng 5 năm 2024

Lời cam đoan

Em xin cam đoan báo cáo bài tập cá nhân số 3 do chính cá nhân em nghiên cứu và thực hiện.

Em đã kiểm tra dữ liệu theo quy định hiện hành.

Kết quả trình bày của báo cáo về bài tập là trung thực và không sao chép từ bất kì bài tập của cá nhân khác.

Các tài liệu được sử dụng trong báo cáo có nguồn gốc, xuất xứ rõ ràng.

Sinh viên thực hiện
(Kí và ghi rõ họ tên)

A handwritten signature in black ink, appearing to read 'Lê Xuân Huy', with a horizontal line underneath.

Lê Xuân Huy

Mục lục

Phần I: Thông tin sinh viên và điểm kì vọng	1
1. Thông tin sinh viên	1
2. Điểm kì vọng.....	1
Phần II: Danh sách chức năng hoàn thành.....	1
Phần III: Link Youtube	1
Phần IV: Cấu trúc mã nguồn.....	1
1. Cấu trúc thư mục	1
2. Các nhóm file trong mã nguồn.....	2
2.1. Các file kết nối, sử dụng Database.....	2
2.2. Các thư viện	3
2.3. Các lớp đối tượng.....	3
2.4. Các servlet.....	4
Phần V: Hướng dẫn sử dụng	5
1. Tạo thư mục chứa mã nguồn.....	6
2. Thiết lập cơ sở dữ liệu.....	7
3. Cài đặt server.....	7

Phần I: Thông tin sinh viên và điểm kì vọng

1. Thông tin sinh viên

Họ tên: Lê Xuân Huy

MSSV: 20120495

Chuyên ngành: Công nghệ phần mềm

SĐT: 0348324331

Niên học: 2020-2024

2. Điểm kì vọng

Điểm kì vọng cho bài tập: 10/10

Phần II: Danh sách chức năng hoàn thành

- Danh sách sinh viên: hiển thị danh sách, thêm, cập nhật, xóa, sắp xếp theo tên, tìm theo tên. Thông tin sinh viên gồm: Mã, tên, ngày sinh, địa chỉ, ghi chú.

- Danh sách lớp học: hiển thị danh sách, thêm, cập nhật, xóa, sắp xếp theo tên, tìm theo tên, tìm theo năm, hiển thị danh sách sinh viên trong lớp, thêm/xóa sinh viên vào lớp. Thông tin lớp gồm: mã, tên, giáo viên, năm và ghi chú.

- Hiển thị danh sách sinh viên trong năm, chọn sinh viên hiển thị danh sách lớp và điểm.

Chức năng thêm: Có thể chỉnh sửa điểm

Phần III: Link Youtube

<https://youtu.be/r0He0cPJGI0>

Phần IV: Cấu trúc mã nguồn

1. Cấu trúc thư mục

src

|---main

 |---java

 | |---database

 | | |---Các file kết nối, sử dụng Database

```
|    |---lib
|    |    |---Các thư viện import
|    |---model
|    |    |---Các mẫu đối tượng: Khóa học, Sinh viên
|    |---servlet
|    |    |---Các servlet sử dụng trong mã nguồn
|---webapp
|    |---META-INF
|    |---WEB-INF
|    |    |---lib
|    |    |    |---Các thư viện import
|    |    |---web.xml (File cấu hình cho mã nguồn)
|    |---Các file JSP tổ chức giao diện
```

2. Các nhóm file trong mã nguồn

2.1. Các file kết nối, sử dụng Database

- JDBCUtil.java

Đây là class sử dụng để kết nối đến database cũng như đóng database.

Danh sách các hàm:

- Connection getConnection(): lấy kết nối
- void closeConnection(Connection c): đóng kết nối

- DAO.java

Đây là class thông qua kết nối với database trả về các đối tượng cần thiết được sử dụng trong mã nguồn, ví dụ: tên sinh viên theo ID, danh sách sinh viên, danh sách khóa học, v.v..

Danh sách các hàm:

- List<Student> getAllStudents(): lấy danh sách sinh viên
- boolean addStudent(String name, String birthday, String address, String notes): thêm sinh viên vào danh sách
- Student getStudentById(String id): lấy sinh viên theo ID
- boolean updateStudent(String id, String name, String birthday, String address, String notes): cập nhật thông tin sinh viên
- boolean deleteStudentById(String id): xóa sinh viên theo ID
- Student getStudentByName(String name): lấy sinh viên theo tên

- `List<Course> getAllCourses()`: lấy danh sách khóa học
- `Course getCourseById(String id)`: lấy khóa học theo ID
- `boolean addCourse(String name, String lecture, int year, String notes)`: thêm khóa học mới vào danh sách
- `boolean updateCourse(String id, String name, String lecture, int year, String notes)`: cập nhật khóa học
- `boolean deleteCourseById(String id)`: xóa khóa học theo ID
- `String getCourseNameById(String id)`: lấy tên khóa học theo ID
- `List<Student> getStudentsByCourseId(String courseId)`: lấy danh sách sinh viên của một khóa học theo ID của khóa học
- `List<Student> getStudentsNotInCourseId(String courseId)`: lấy danh sách sinh viên không thuộc khóa học theo ID của khóa học
- `boolean insertStudentInCourse(String studentId, String courseId)`: thêm sinh viên vào khóa học
- `boolean deleteStudentInCourse(String studentId, String courseId)`: xóa sinh viên khỏi khóa học
- `List<Student> getStudentInYear(int year)`: lấy danh sách sinh viên có môn học trong năm nào đó
- `List<Course> getCourseByStudentId(String studentId)`: lấy danh sách khóa học của một sinh viên
- `boolean updateGrade(String studentId, String courseId, float grade)`: cập nhật điểm cho sinh viên đó

2.2. Các thư viện

.../jakarta.servlet-api-6.1.0-M2.jar: hỗ trợ sử dụng servlet

.../mysql-connector-java-8.0.30.jar: driver sử dụng kết nối DB MySQL

2.3. Các lớp đối tượng

- Student.java

Đây là class sinh viên, có các thuộc tính: `private String id`, `private String name`, `private String birthday`, `private String address`, `private String notes`.

Có các hàm getter, setter cơ bản.

- Course.java

Đây là class khóa học, có các thuộc tính: `private String id`, `private String name`, `private String lecture`, `private int year`, `private String notes`, `private float grade`,

Có các hàm getter, setter cơ bản.

2.4. Các servlet

- AddCourseServlet.java:

Post: thêm Khóa học vào danh sách, chuyển hướng đến "viewCourse.jsp"

- AddStudentServlet.java:

Post: thêm Sinh viên vào danh sách, chuyển hướng đến "viewStudent.jsp"

- DeleteCourseServlet.java:

Post: xóa Khóa học khỏi danh sách, chuyển hướng đến "viewCourse.jsp"

- DeleteStudentServlet.java:

Post: xóa Sinh viên khỏi danh sách, chuyển hướng đến "viewStudent.jsp"

- InsertDeleteStudentInCourseServlet.java:

Post: thực hiện thêm hoặc xóa Sinh viên tại một khóa học chỉ định tùy theo request, chuyển hướng đến "viewDetailCourse.jsp"

- SearchNameCourseServlet.java:

Get: thực hiện tìm kiếm danh sách Khóa học theo tên, chuyển hướng đến trang "viewCourse.jsp"

- SearchYearCourseServlet.java:

Get: thực hiện tìm kiếm danh sách Khóa học theo năm học, chuyển hướng đến trang "viewCourse.jsp"

- SearchStudentServlet.java:

Get: thực hiện tìm kiếm danh sách Sinh viên theo tên, chuyển hướng đến trang "viewStudent.jsp"

- SearchYearStudentServlet.java:

Get: thực hiện tìm kiếm danh sách Sinh viên theo năm học, chuyển hướng đến trang "viewStudent.jsp"

- SortCourseServlet.java:

Get: lấy ra danh sách Khóa học, sắp xếp tăng dần hoặc giảm dần theo request, chuyển hướng đến trang "viewCourse.jsp"

- SortStudentServlet.java:

Get: lấy ra danh sách Sinh viên, sắp xếp tăng dần hoặc giảm dần theo request, chuyển hướng đến trang "viewStudent.jsp"

- UpdateCourseServlet.java:

Get: lấy Khóa học theo ID, chuyển hướng đến trang "updateCourse.jsp"

Post: thực hiện cập nhật khóa học theo thông tin đã nhập, chuyển hướng đến trang "viewCourse.jsp"

- UpdateStudentServlet.java:

Get: lấy Sinh viên theo ID, chuyển hướng đến trang "updateStudent.jsp"

Post: thực hiện cập nhật sinh viên theo thông tin đã nhập, chuyển hướng đến trang "viewStudent.jsp"

- UpdateGradeServlet.java:

Post: thực hiện cập nhật điểm của một sinh viên ở một môn nào đó.

- ViewCourseServlet.java:

Get: lấy danh sách Khóa học, chuyển hướng đến "viewCourse.jsp"

- ViewStudentServlet.java:

Get: lấy danh sách Sinh viên, chuyển hướng đến "viewStudent.jsp"

- ViewDetailCourseServlet.java:

Get: thực hiện lấy danh sách sinh viên thuộc và không thuộc một khóa học nào đó, chuyển hướng đến trang "viewDetailCourse.jsp"

- ViewCourseOfStudent.java:

Get: thực hiện lấy danh sách lớp học của một sinh viên nào đó, chuyển hướng đến trang "viewCourseOfStudent.jsp"

2.5. File JSP tổ chức giao diện

- **addCourse.jsp:** giao diện thêm Khóa học
- **addStudent.jsp:** giao diện thêm Sinh viên
- **updateCourse.jsp:** giao diện cập nhật Khóa học
- **updateStudent.jsp:** giao diện cập nhật Sinh viên
- **viewCourse.jsp:** giao diện xem danh sách Khóa học
- **viewStudent.jsp:** giao diện xem danh sách Sinh viên
- **viewDetailCourse.jsp:** giao diện xem chi tiết một Khóa học, danh sách Sinh viên thuộc và không thuộc Khóa học đó
- **viewCourseOfStudent.jsp:** giao diện xem danh sách khóa học của một sinh viên nào đó
- **viewStudentInYear.jsp:** giao diện xem danh sách Sinh viên trong năm học

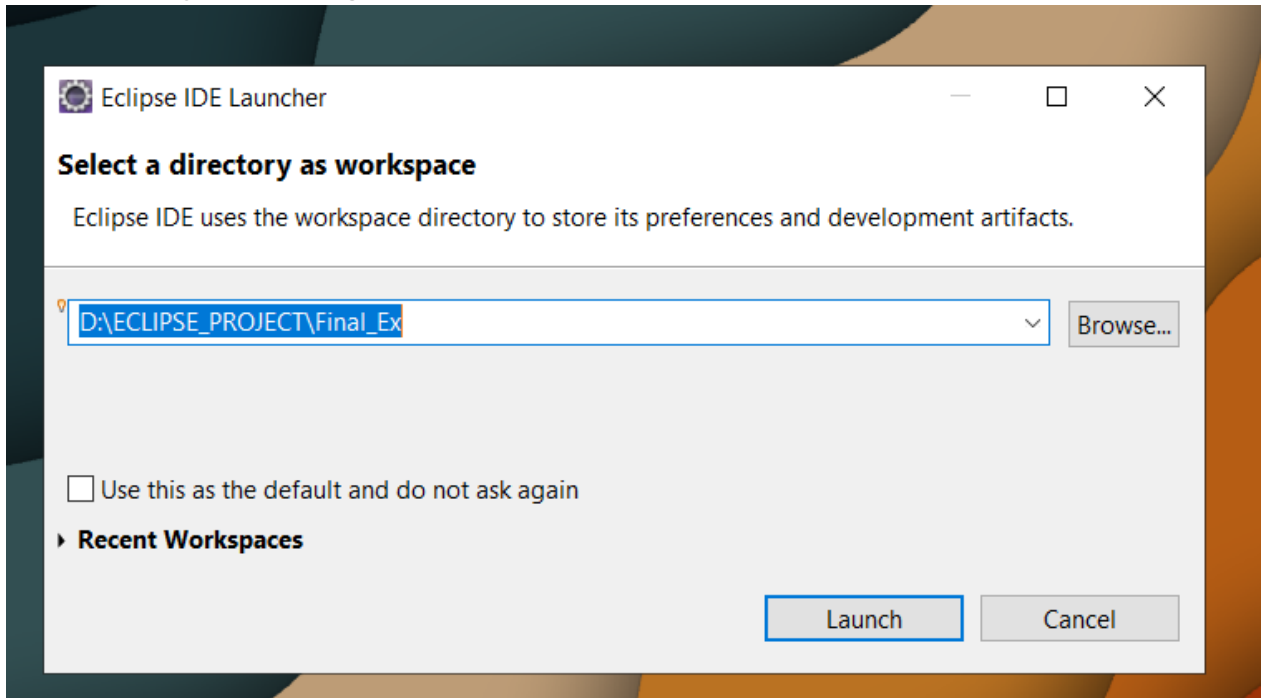
Phần V: Hướng dẫn sử dụng

- IDE: Eclipse
- Server: apache-tomcat-10.1.23

Tải IDE Eclipse để mở mã nguồn. (Tự tham khảo)

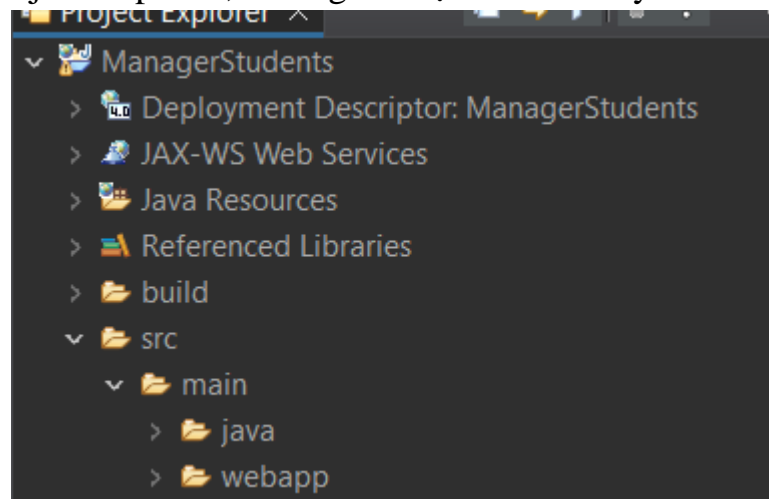
1. Tạo thư mục chứa mã nguồn

Đầu tiên bạn cần tạo thư mục chứa toàn bộ mã nguồn, toàn bộ mã nguồn tính từ thư mục gốc “ManagerStudents”



Trong ví dụ, thư mục chứa toàn bộ mã nguồn chính là đường dẫn: “D:\ECLIPSE_PROJECT\Final_Ex”.

Tại mục Project Explorer, sẽ có giao diện như thế này:



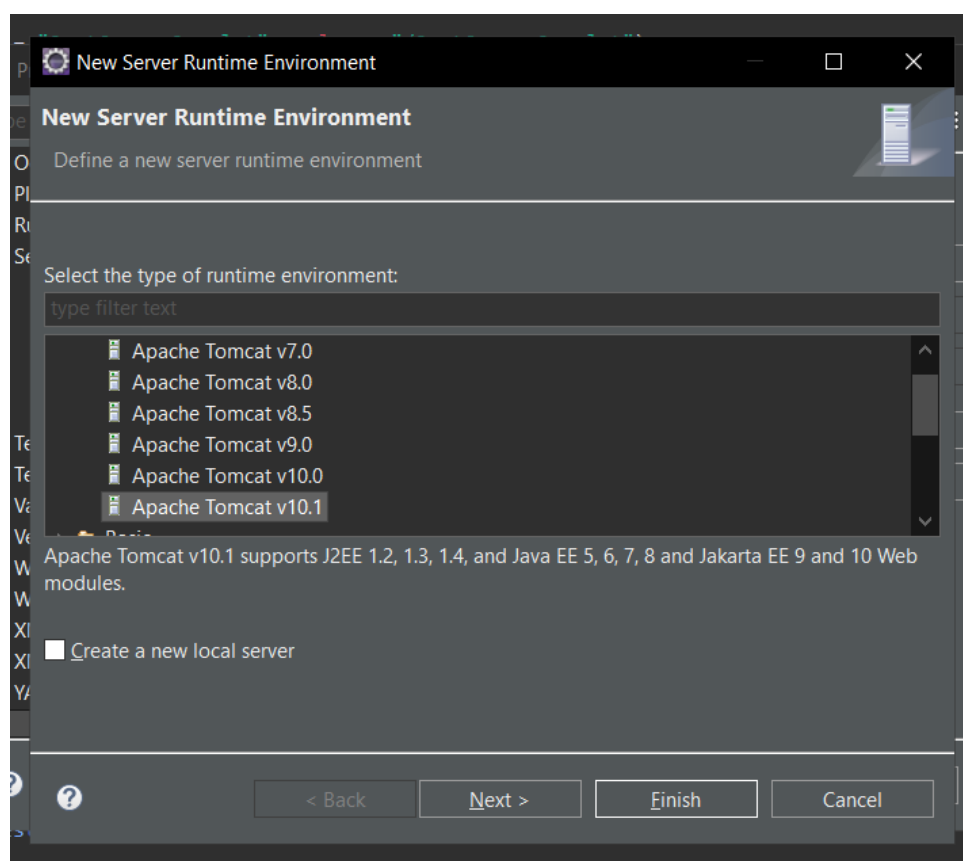
2. Thiết lập cơ sở dữ liệu

Cơ sở dữ liệu được sử dụng ở đây là MySQL, đã có Script SQL để chạy Cơ sở dữ liệu trên các Hệ quản trị CSDL khác nhau. Tùy vào Hệ quản trị CSDL mà bạn có các cài đặt Driver cũng như kết nối cho phù hợp.

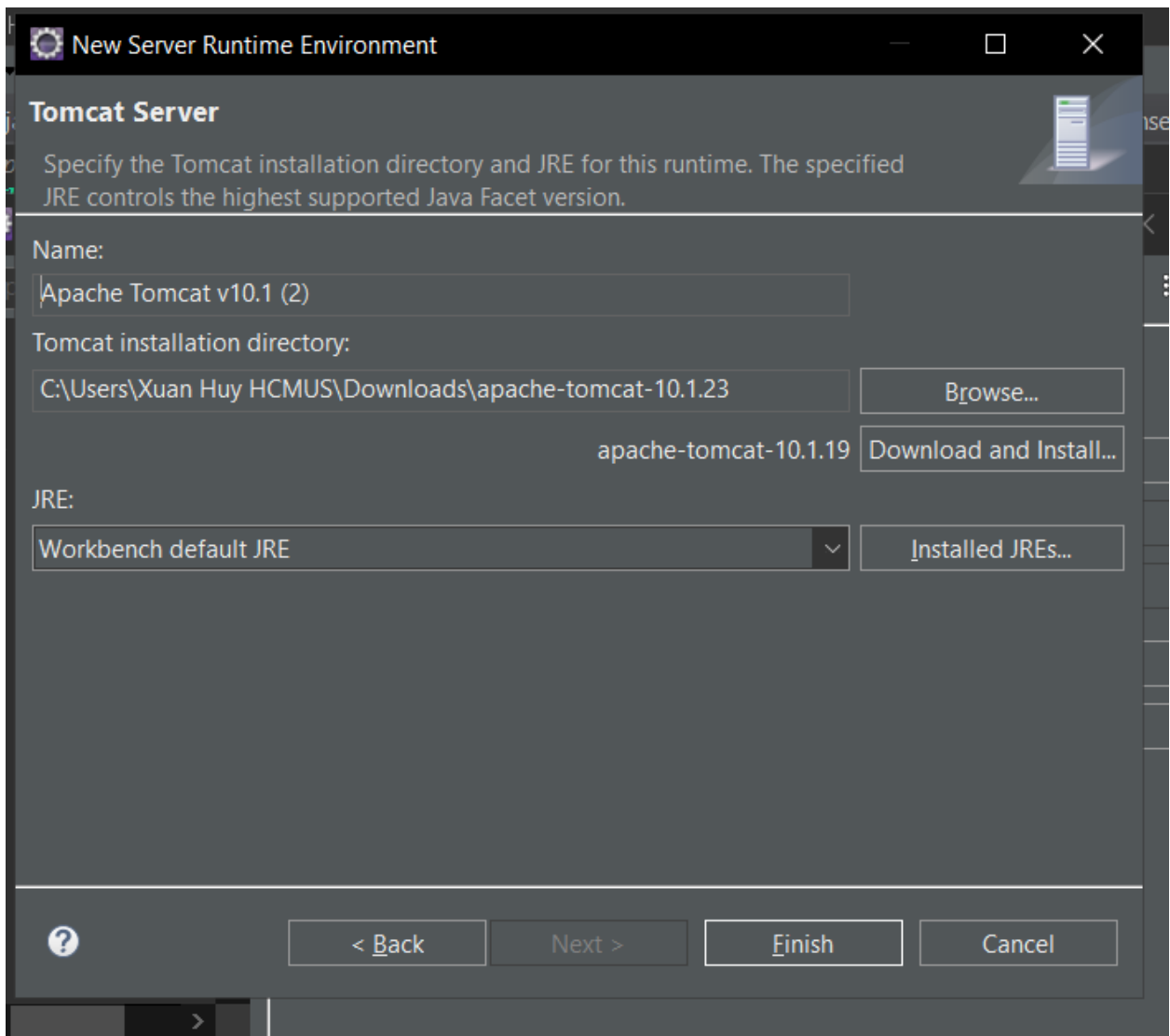
3. Cài đặt server

Server sử dụng ở đây là: apache-tomcat-10.1.23. (Ít nhất phải là phiên bản 10.1.x)

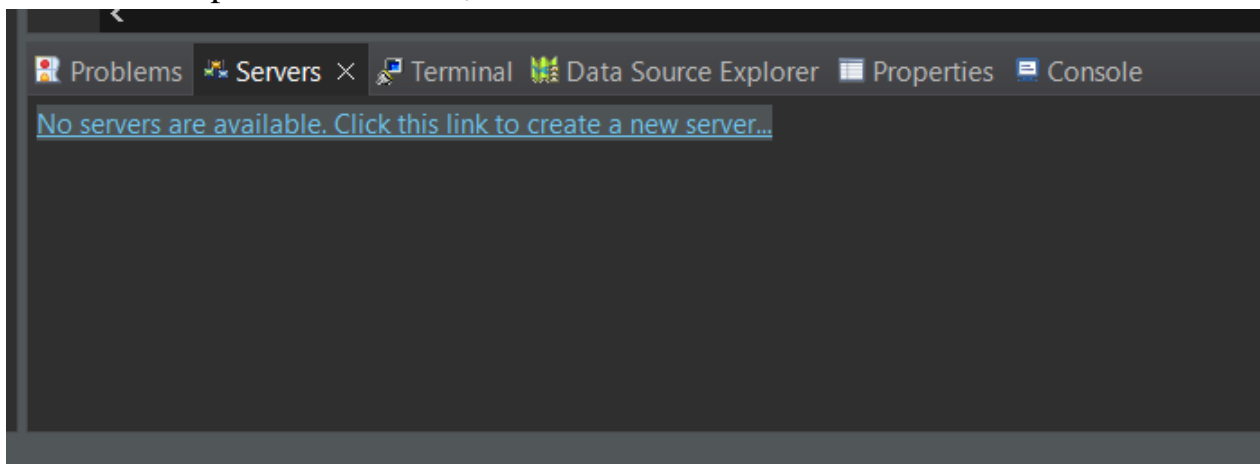
Bạn download gói apache-tomcat về, chọn Window → Preferences → Server → Runtime Environments → Add



Chọn phiên bản phù hợp. Sau đó chọn đường dẫn bạn đã tải. Ví dụ:



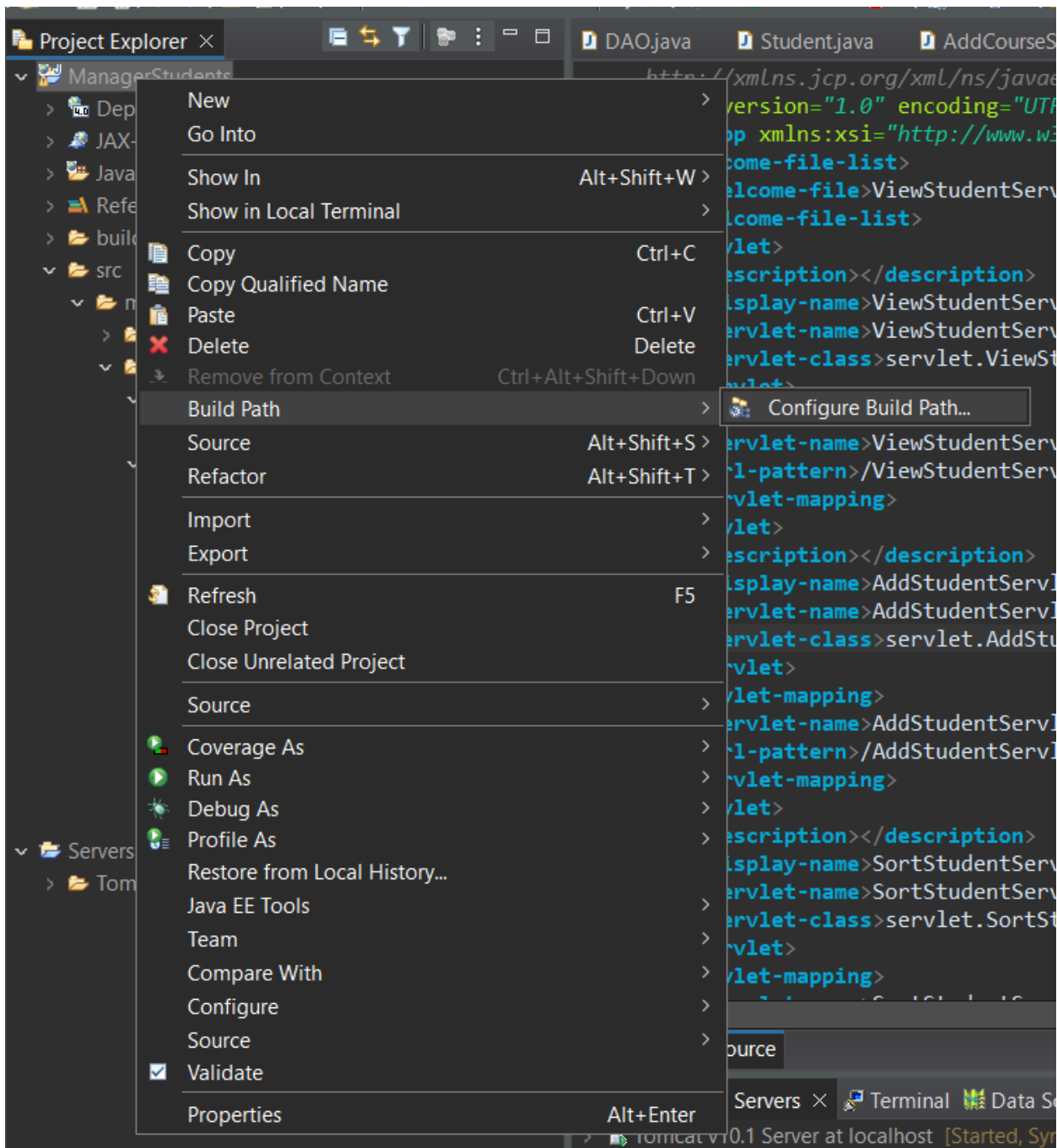
Bước tiếp theo là kích hoạt server:



Tại mục server, chọn server đã thêm vào.

Trước khi chạy chương trình, hãy thêm đường dẫn của ser về vào dự án:

Nháy chuột phải vào thư mục gốc, chọn Build Path → Configure Build Path...
→ Add Library → Server Runtime. Sau đó chọn Server phù hợp và Apply.



Nháy chuột phải vào thư mục gốc, chọn Run As → Run on Server.

