

image.png

PNG 26.3KB

i am getting this error when i am creating a servlet application i request you to correct the code

HTTP 404 Error in Servlet Application - Troubleshooting Guide

The 404 "Not Found" error you're encountering typically means Tomcat can't locate your servlet at the requested URL. Let me help you diagnose and fix this issue.

Common Causes and Solutions

1. Incorrect URL Mapping

- **Problem:** The URL `/OnlineSamyak/Insert` doesn't match your servlet's URL pattern
- **Solution:** Check your servlet's `@WebServlet` annotation or `web.xml` mapping

java

Copy

// Example of correct `WebServlet` annotation

`@WebServlet("/Insert")` // or `"/OnlineSamyak/Insert"` if that's your intended path

```
public class InsertServlet extends HttpServlet {
```

```
    // your servlet code
```

```
}
```

2. Deployment Issues

- **Problem:** The application isn't deployed correctly
- **Solution:**
 - Verify your WAR file is in Tomcat's `webapps` directory
 - Check Tomcat's logs for deployment errors
 - Ensure your application context is `OnlineSamyak` (should match your URL)

3. Missing Servlet Class

- **Problem:** The compiled servlet class isn't in the right location
- **Solution:**
 - Check your `WEB-INF/classes` directory contains the compiled servlet
 - Verify your build process includes all necessary classes

4. Web.xml Configuration (if using)

xml

Copy

<!-- Example web.xml configuration -->

<servlet>

 <servlet-name>InsertServlet</servlet-name>

 <servlet-class>com.yourpackage.InsertServlet</servlet-class>

</servlet>

<servlet-mapping>

 <servlet-name>InsertServlet</servlet-name>

 <url-pattern>/Insert</url-pattern>

</servlet-mapping>

Run HTML

Verification Steps

1. Check Tomcat's localhost.log for errors during deployment
2. Verify your application appears in Tomcat's Manager app (<http://localhost:8080/manager>)
3. Try accessing a simple HTML page first to verify basic deployment

Additional Checks

- Make sure you're using the correct Tomcat version (your error shows 10.1.39)
- Verify your servlet extends HttpServlet
- Check for any typos in your servlet class name or package

Would you like to share your servlet code and project structure? That would help me provide more specific guidance.

give a servlet project code to do registration for a student to participate in a event and send to servlet therer to database

Student Event Registration Servlet Project

Here's a complete servlet project for student event registration that stores data in a database.

1. Project Structure

Copy

StudentEventRegistration/

└─ src/

| └─ main/

| | └─ java/

| | | └─ com/

| | | └─ example/

```

| | |   └─ model/
| | |   |   └─ Student.java
| | |   └─ dao/
| | |   |   └─ StudentDAO.java
| | |   └─ servlet/
| | |   |   └─ RegistrationServlet.java
| | └─ webapp/
| | |   └─ WEB-INF/
| | |   |   └─ web.xml
| | |   └─ register.jsp
| | |   └─ success.jsp
| | |   └─ error.jsp
└─ pom.xml (for Maven)

```

2. Database Setup (MySQL Example)

sql

Copy

```
CREATE DATABASE event_registration;
```

```
USE event_registration;
```

```
CREATE TABLE students (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    name VARCHAR(100) NOT NULL,
```

```
    email VARCHAR(100) NOT NULL UNIQUE,
```

```
    phone VARCHAR(15),
```

```
    college VARCHAR(100),
```

```
    event VARCHAR(50) NOT NULL,
```

```
    registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

```
);
```

3. Model Class (Student.java)

java

Copy

```
package com.example.model;
```

```
import java.sql.Timestamp;
```

```
public class Student {
```

```
    private int id;
```

```
    private String name;
```

```
    private String email;
```

```
    private String phone;
```

```
    private String college;
```

```
    private String event;
```

```
    private Timestamp registrationDate;
```

```
    // Constructors, getters and setters
```

```
    public Student() {}
```

```
    public Student(String name, String email, String phone, String college, String event) {
```

```
        this.name = name;
```

```
        this.email = email;
```

```
        this.phone = phone;
```

```
        this.college = college;
```

```
        this.event = event;
```

```
    }
```

```
    // Getters and setters for all fields
```

```
    // ...
```

```
}
```

4. Data Access Object (StudentDAO.java)

```
java
```

Copy

```
package com.example.dao;

import com.example.model.Student;
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class StudentDAO {

    private String jdbcURL = "jdbc:mysql://localhost:3306/event_registration";
    private String jdbcUsername = "root";
    private String jdbcPassword = "password";

    private static final String INSERT_STUDENT_SQL =
        "INSERT INTO students (name, email, phone, college, event) VALUES (?, ?, ?, ?, ?)";

    protected Connection getConnection() {
        Connection connection = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(jdbcURL, jdbcUsername, jdbcPassword);
        } catch (SQLException | ClassNotFoundException e) {
            e.printStackTrace();
        }
        return connection;
    }

    public boolean registerStudent(Student student) {
        boolean result = false;
        try (Connection connection = getConnection();
            PreparedStatement statement = connection.prepareStatement(INSERT_STUDENT_SQL)) {
            statement.setString(1, student.getName());
            statement.setString(2, student.getEmail());
            statement.setString(3, student.getPhone());
            statement.setString(4, student.getCollege());
            statement.setString(5, student.getEvent());
            statement.executeUpdate();
            result = true;
        }
    }
}
```

```

        statement.setString(1, student.getName());
        statement.setString(2, student.getEmail());
        statement.setString(3, student.getPhone());
        statement.setString(4, student.getCollege());
        statement.setString(5, student.getEvent());

        result = statement.executeUpdate() > 0;
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result;
}
}

```

5. Servlet (RegistrationServlet.java)

java

Copy

```

package com.example.servlet;

import com.example.dao.StudentDAO;
import com.example.model.Student;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/register")
public class RegistrationServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private StudentDAO studentDAO;

```

```

public void init() {
    studentDAO = new StudentDAO();
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String phone = request.getParameter("phone");
    String college = request.getParameter("college");
    String event = request.getParameter("event");

    Student student = new Student(name, email, phone, college, event);

    try {
        boolean result = studentDAO.registerStudent(student);
        if (result) {
            response.sendRedirect("success.jsp");
        } else {
            response.sendRedirect("error.jsp");
        }
    } catch (Exception e) {
        e.printStackTrace();
        response.sendRedirect("error.jsp");
    }
}
}

```

6. JSP Pages

register.jsp

jsp

Copy

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

  <meta charset="UTF-8">

  <title>Event Registration</title>

  <style>

    body { font-family: Arial, sans-serif; margin: 20px; }

    form { max-width: 500px; margin: 0 auto; }

    .form-group { margin-bottom: 15px; }

    label { display: block; margin-bottom: 5px; }

    input, select { width: 100%; padding: 8px; }

    button { padding: 10px 15px; background-color: #4CAF50; color: white; border: none; }

  </style>

</head>

<body>

  <h1>Event Registration Form</h1>

  <form action="register" method="post">

    <div class="form-group">

      <label for="name">Full Name:</label>

      <input type="text" id="name" name="name" required>

    </div>

    <div class="form-group">

      <label for="email">Email:</label>

      <input type="email" id="email" name="email" required>

    </div>

    <div class="form-group">
```



```
<label for="phone">Phone:</label>
<input type="tel" id="phone" name="phone">
</div>
```

```
<div class="form-group">
  <label for="college">College/Institution:</label>
  <input type="text" id="college" name="college">
</div>
```

```
<div class="form-group">
  <label for="event">Select Event:</label>
  <select id="event" name="event" required>
    <option value="">-- Select an Event --</option>
    <option value="Coding Competition">Coding Competition</option>
    <option value="Robotics Workshop">Robotics Workshop</option>
    <option value="Hackathon">Hackathon</option>
    <option value="Seminar">Seminar</option>
  </select>
</div>
```

```
<button type="submit">Register</button>
</form>
</body>
</html>
```

success.jsp

.jsp

Copy

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
```

```
<meta charset="UTF-8">
<title>Registration Successful</title>
</head>
<body>
  <h1>Registration Successful!</h1>
  <p>Thank you for registering for the event. We'll contact you soon with more details.</p>
  <a href="register.jsp">Back to Registration</a>
</body>
</html>
```

error.jsp

jsp

Copy

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Registration Error</title>
</head>
<body>
  <h1>Registration Error</h1>
  <p>Sorry, there was an error processing your registration. Please try again later.</p>
  <a href="register.jsp">Back to Registration</a>
</body>
</html>
```

7. web.xml (Optional if using annotations)

xml

Copy

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
version="4.0">
```

```
<display-name>Student Event Registration</display-name>
```

```
<welcome-file-list>
```

```
    <welcome-file>register.jsp</welcome-file>
```

```
</welcome-file-list>
```

```
</web-app>
```

Run HTML

8. pom.xml (Maven Dependencies)

xml

Copy

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
```

```
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
```

```
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
<modelVersion>4.0.0</modelVersion>
```

```
<groupId>com.example</groupId>
```

```
<artifactId>StudentEventRegistration</artifactId>
```

```
<version>1.0-SNAPSHOT</version>
```

```
<packaging>war</packaging>
```

```
<dependencies>
```

```
    <!-- Servlet API -->
```

```
    <dependency>
```

```
        <groupId>javax.servlet</groupId>
```

```
        <artifactId>javax.servlet-api</artifactId>
```

```
        <version>4.0.1</version>
```

```

        <scope>provided</scope>
    </dependency>

    <!-- JSTL -->
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.28</version>
    </dependency>
</dependencies>

<build>
    <finalName>StudentEventRegistration</finalName>
</build>
</project>

```

Run HTML

Deployment Instructions

1. Create the database and table as shown above
2. Update the database credentials in StudentDAO.java
3. Build the project (mvn clean package for Maven)
4. Deploy the WAR file to Tomcat's webapps directory
5. Access the application at: <http://localhost:8080/StudentEventRegistration/register.jsp>

This complete solution handles student registration for events and stores the data in a MySQL database. The servlet processes the form submission, the DAO handles database operations, and JSP pages provide the user interface.

