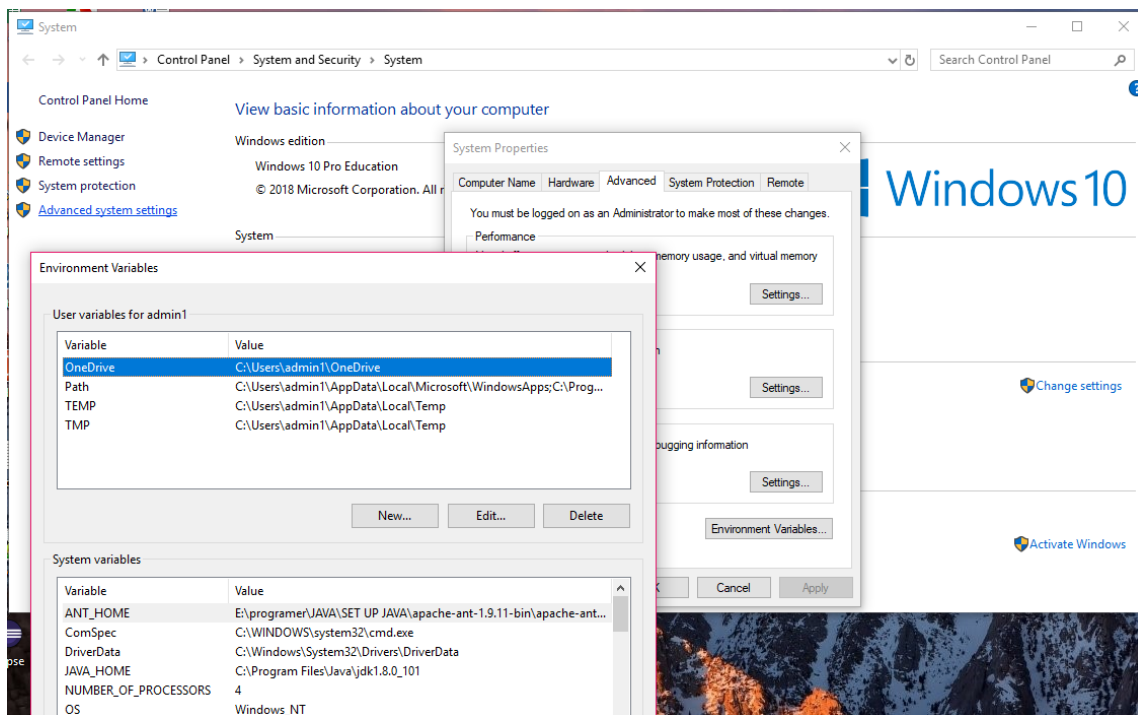


DOCUMENT JAVA WEB

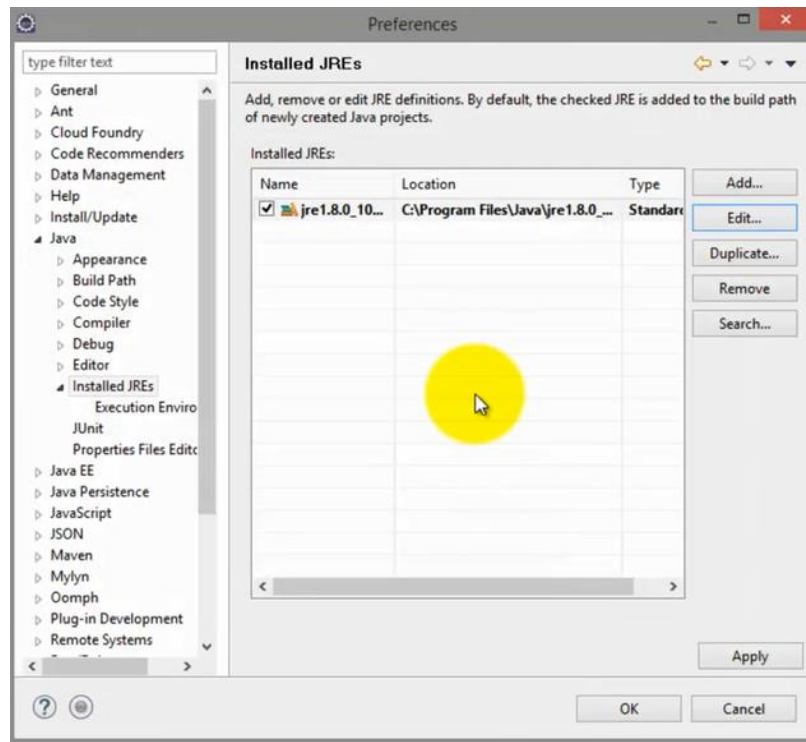
I. JSP/SERVLET

1. Cấu hình eclipse

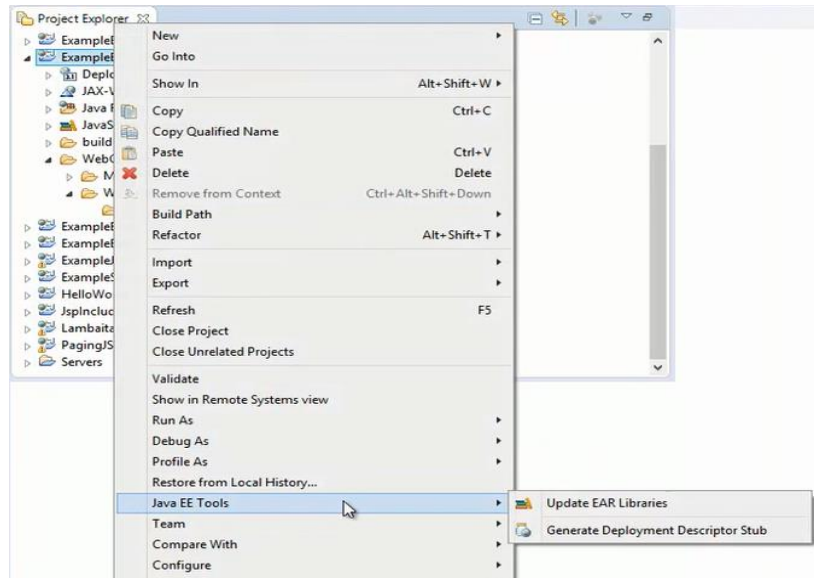
- Vào trang <http://www.eclipse.org/downloads/> chọn **Download Packages** bên dưới download 64 bit để download file zip về giải nén
- <http://www.eclipse.org/downloads/eclipse-packages/>: cách vào trực tiếp chọn 64bit ở dòng đầu tiên **Eclipse IDE for Java EE Developers** và nhấp sau đó khi mở trang mới vào **File: eclipse-jee-oxygen-3a-win32-x86_64.zip** để tải về
- Tải jdk: gõ jdk vào trang chủ, sau khi click đồ biểu tượng ở đầu ta vào trang <http://www.oracle.com/technetwork/java/javase/downloads/jdk10-downloads-4416644.html> sau đó chọn accept radio button
- Kiểm tra máy có jdk cũ đang chạy k có thì xóa
- Nếu chưa có This PC thì vào Personalize/ change desktop icon/ tick vào computer
- Cấu hình biến môi trường vào link này và copy C:\Program Files\Java\jdk1.8.0_101 dán vào JAVA_HOME ở trên và dưới



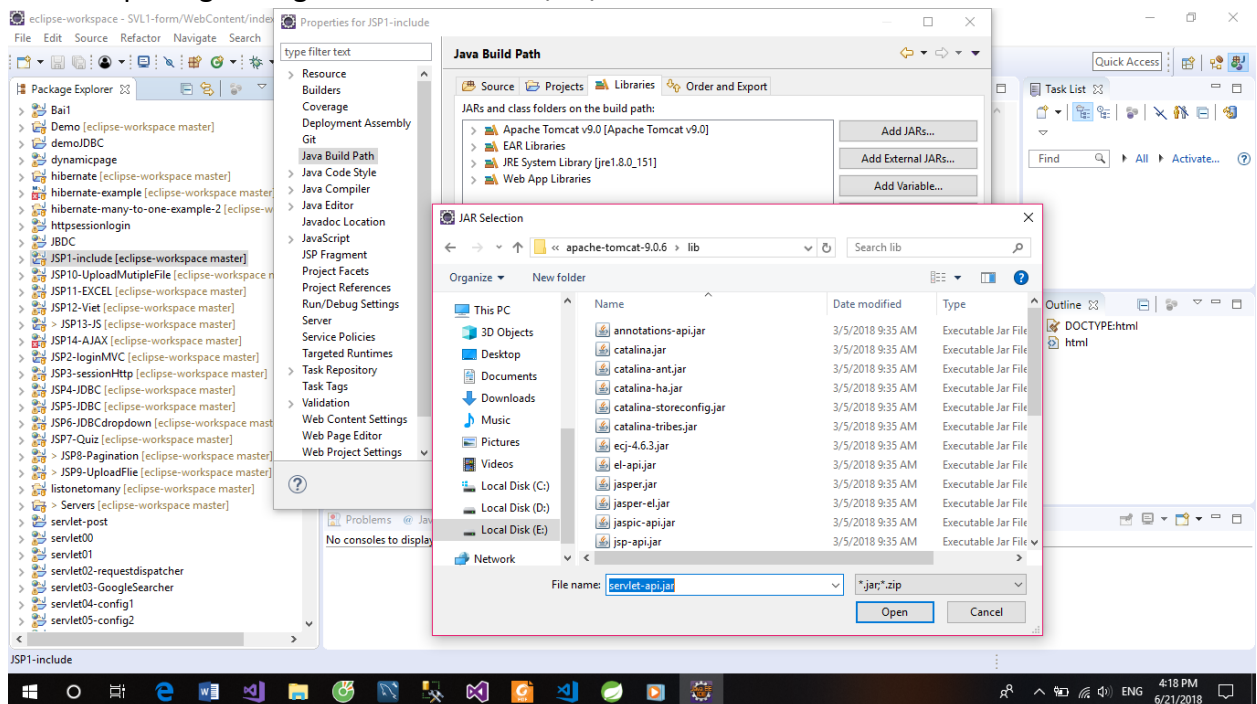
- Sau đó vào eclipse/ restart lại và chọn windows/preferences và kiểm tra vào java/installed jres:



- Cài mysql: gỡ mysql workbench: <https://www.mysql.com/products/workbench/> vào downloads/community/mysql installer
- Link vào nhanh <https://dev.mysql.com/downloads/windows/installer/> rồi chọn bản 230MB về, sau đó chọn no thanks, just start my download về, nhập pass = 1234 nhớ check connection trước
- Cấu hình apache tomcat: <https://tomcat.apache.org/download-90.cgi> chọn 64-bit Windows zip (pgp, sha1, sha512)
- Sau khi giải nén ra vào **windows/preferences/server/runtime environment** và chọn add apache tomcat 9
- Nhấp vào tab phía dưới có tên là server và chọn tomcat 9 khi nhấn vào link, sau đó có folder server ta nhấp chuột phải vào link phía dưới chọn start để check error
- Vào **window/web browser/ chọn default browser**
- File new/dynamic web project/ chọn next/ tick vào generate web with web.xml file
- Nếu quên tick web.xml thì lm ntn:



- Để kiểm tra ta tạo 1 file.jsp trong webcontent và vào properties của **project/ java build path/ add external jar/ servlet-api.jar (trong lib của apache)** sau đó chọn apply ra ấn ok để không lỗi nữa
- Cách 2: Load thư viện servlet api server tomcat
 - Copy thư viện từ folder lib/ servlet-api.jar sau đó paste vào C:\Program Files\Java\jre1.8.0_151\lib\ext
- Vào **windows/preferences/ web/jsp file** chọn ngôn ngữ là utf 8
- File new package trong src sau đó refresh/ tạo new servlet mới



2. Servlet

- So sánh doGet và doPost:

doGet	doPost
<ul style="list-style-type: none">- Tham số dk gán trên url- Có giới hạn 240 byte- K dk mã hóa- Nhanh hơn và cần được bảo mật- SD trong truy vấn lấy dl từ server	<ul style="list-style-type: none">- Update, post information to server

- Servlet request gửi yêu cầu từ client, provide some method such as getParameter, get/setAttribute, .. thông qua lớp HttpServletRequest
- So sánh jsp, servlet

Jsp	Servlet
<ul style="list-style-type: none">- K cần recompile khi thay đổi code- Tk giao diện easy- Dễ SD và bảo trì vì 2 tầng BL và presentation k bị mixed	<ul style="list-style-type: none">- Dễ viết code java => controller

- `response.setContentType("text/html; charset=utf-8");`
- Khi muốn in ra có dấu, có thể sd for để in cho nhanh
- Cách sử dụng form: file servlet trong src/package

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

```
    response.setContentType("text/html");  
    PrintWriter out=response.getWriter();
```

```
    String s = request.getParameter("username");  
    out.println("Welcome " + s);
```

```
}
```

- Cấu hình tay trong web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  
    http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">  
  <display-name>SVL2-form</display-name>  
  <welcome-file-list>  
    <welcome-file>index.html</welcome-file>  
  </welcome-file-list>  
  
  <servlet>
```

```

<servlet-name>Hello</servlet-name>
<servlet-class>com.duyphuong.Hello</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>Hello</servlet-name>
  <url-pattern>/next</url-pattern>
</servlet-mapping>
</web-app>

```

- Còn file jsp trong web content:

```

<form action="next" method="get">
  <input type="text" name="username" />
  <input type="submit" value="Display"/>
</form>

```

- Lấy tham số từ form:

```
String s = request.getParameter("username");
```

- Cấu hình bằng annotation: @WebServlet("/Hello")

```
<form action="Hello" method="get">
```

3. Cài đặt jstl

- Tải thư viện jstl 1.2 jar <http://www.java2s.com/Code/Jar/j/Downloadjstl12jar.htm>
- sau đó giải nén và copy vào trong thư mục lib của webcontent

```
<%@ taglib prefix = "c" uri = "http://java.sun.com/jsp/jstl/core" %>
```

- Expression language(EL) được dùng để truy cập dl được lưu trữ bởi cái đối tượng như implicit obj: session, request,... Ngoài ra còn dk SD trong jstl core tag
- Cú pháp: \${obj}

```

1  <!--@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2
3  <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
4
5  <html>
6
7  <head>
8    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
9    <title>Display Page</title>
10 </head>
11 <body>
12
13   <table border="1">
14
15     <c:forEach items = "${listusers}" var = "list">
16
17       <tr>
18         <td>
19           ${list.name1}
20         </td>
21         <td>
22           ${list.name2}
23         </td>
24       </tr>
25     </c:forEach>
26   </table>
27 </body>
28 </html>

```

- khi muốn nhấp vài nút nhảy qua 1 trang jsp khác để hiển thị danh sách <form action="a.jsp" method="get">
- trong trang a.jsp: chỉ có RequestDispatcher và setAttributes
- <%@ page import = "java.util.*" %>
- <% viết code java trong này %>
- <h1><%= request.getAttribute("message") %></h1> //in ra
- Request.setAttributes("name", DAO.ExportList());
- Chuyển trang
RequestDispatcher rd = request.getRequestDispatcher("/WEB-INF/View/Login.jsp");
- rd.forward(request, response);
- Tạo class DAO và Users DAO trả về 1 list Request.getAttributes("name");
- Nếu lỗi thì thêm _rt sau core, chỉ có var="list" còn lại để \${ }
- Jsp action: jsp:include
- <jsp:include page="Header.jsp"/>
- Cách tạo theo mô hình MVC: tạo view trong webcontent, tạo 1 file index.jsp để dẫn vào Click here
- <h1><%= request.getAttribute("message2") != null ? request.getAttribute("message2"): " " %></h1> // nếu lỗi thì in ra

4. Session tracking

- Là quản lý phiên làm việc của user
- 4 kỹ thuật:
 - Cookies
 - Hidden form field
 - URL rewriting
 - HttpSession
- Các bước quản lý session:
 - B1: khởi tạo thông qua phương thức getSession HttpSession session = request.getSession(true); // true là tạo mới session; false là trả về session có gtri null
 - B2: Lưu trữ: session.setAttribute("sessionname", name);
 - B3: Hiển thị: <%= session.getAttribute("sessionname") %>
 - B4: xóa: HttpSession session = request.getSession(false); session.removeAttribute("sessionname")

5. JDBC

- Là 1 chuẩn API có cn connect java programming language to database
- JSP/ servlet sd jdbc để kết nối csdl
- Các bước tạo kết nối:
 - GG: mysql connector:
<https://www.mysql.com/products/connector/?PHPSESSID=17d1801c98afa5268fd9e336f865adde>
JDBC Driver for MySQL (Connector/J) : tải về

- Import vào trong iib
- Cấu hình để SD:

```
package DB;
import java.sql.*;

public class DBConnection {
    private static String DB_URL = "jdbc:mysql://localhost:3306/jspbai4";
    private static String USER_NAME = "root";
    private static String PASSWORD = "1234";
    public static Connection createConnection() {
        Connection conn = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            conn = DriverManager.getConnection(DB_URL, USER_NAME,
PASSWORD);
            System.out.println("connect successfully!");
        } catch (Exception ex) {
            System.out.println("connect failure!");
            ex.printStackTrace();
        }

        return conn;
    }
}
```

Sau khi sd xong nhớ đóng conn.close();

- PreparedStatement

```
public class RegisterDAO {
    public static boolean InsertAccount(Connection conn, Account acc) {
        PreparedStatement pstmt = null;
        String sql = "insert into jsppai4.account(uname,pass) values (?,?)";
        try {
            pstmt = conn.prepareStatement(sql);
            String uname = acc.getUname();
            String pass = acc.getPass();
            pstmt.setString(1, uname);
            pstmt.setString(2, pass);
            int kt = pstmt.executeUpdate();
            // pstmt.executeQuery(); để select
            if(kt != 0) {
                pstmt.close();
                return true;
            }
        }
    }
}
```

```

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return false;
}
}
}
- Lấy ra danh sách acc:

    public static List<Account> DisplayAccount(Connection conn) {
        List<Account> list = null;
        list = new ArrayList<Account>();
        PreparedStatement pstmt = null;
        String sql = "select * from jspbai4.account";
        try {
            pstmt = conn.prepareStatement(sql);

            ResultSet rs = pstmt.executeQuery();
            while(rs.next()) {
                list.add(new Account(rs.getString(1),
rs.getString(2)));
            }
            if(list != null) {
                pstmt.close();
            }
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return list;
    }
}

```

.....

JDBC

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

```



```

public class ConnectSQLServerExample {
    private static String DB_URL = "jdbc:sqlserver://localhost:1433;"
        + "databaseName=testdb;"
        + "integratedSecurity=true";
    private static String USER_NAME = "sa";
    private static String PASSWORD = "1234567890";

    /**
     */
    public static void main(String args[]) {
        try {
            // connect to database 'testdb'
            Connection conn = getConnection(DB_URL, USER_NAME, PASSWORD);
            // create statement
            Statement stmt = conn.createStatement();
            // insert 'student'
            stmt.executeUpdate("INSERT INTO student(id, name, address) "
                + "VALUES (5, 'Vinh', 'Hanoi')");
            // get data from table 'student'
            ResultSet rs = stmt.executeQuery("select * from student");
            // show data
            while (rs.next()) { //cot 1 2 3
                System.out.println(rs.getInt(1) + " " + rs.getString(2)
                    + " " + rs.getString(3));
            }
            // close connection
            conn.close();
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }
}

```

```

/**
    Class.forName("com.mysql.jdbc.Driver");// MYSQL

*/
public static Connection getConnection(String dbURL, String userName,
    String password) {
    Connection conn = null;
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        conn = DriverManager.getConnection(dbURL, userName, password);
        System.out.println("connect successfully!");
    } catch (Exception ex) {
        System.out.println("connect failure!");
        ex.printStackTrace();
    }
    return conn;
}
}

```

PREPARE STM

```

String sqlInsert = "INSERT INTO student VALUES(?, ?, ?)";
String selectAll = "SELECT * FROM student";
try {
    // connect to database
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection(DB_URL,
        USER_NAME, PASSWORD);

    // crate statement to insert student
    PreparedStatement stmt = conn.prepareStatement(sqlInsert);
    stmt.setInt(1, 1);
    stmt.setString(2, "Vinh");
    stmt.setString(3, "Hanoi");
    stmt.execute();
}

```

```

// select all student
stmt = conn.prepareStatement(selectAll);
// get data from table 'student'
ResultSet rs = stmt.executeQuery();
// show data
while (rs.next()) {
    System.out.println(rs.getInt(1) + " " + rs.getString(2)
        + " " + rs.getString(3));
}
ResultSetMetaData rsmd = rs.getMetaData();

// show metadata
System.out.println("Tong so column cua bang 'student': "
    + rsmd.getColumnCount());
System.out.println("Ten column thu 2: "
    + rsmd.getColumnName(2));
System.out.println("Column type cua column thu 2: "
    + rsmd.getColumnTypeName(2));
stmt.close();
conn.close();
} catch (SQLException ex) {
    ex.printStackTrace();
} catch (ClassNotFoundException e) {
    e.printStackTrace();
}
}

```

6. Export/ Import data to mysql

- Export: click vào navigation


```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Insert title here</title>
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
<script src="bootstrap/js/jquery-3.3.1.min.js"></script>
<script src="bootstrap/js/bootstrap.min.js"></script>
</head>

```

- Click here

//khi lan dau ta an vao link //k co khoang trang

- String pageidsrt = request.getParameter("pageid"); //lấy tham số

10. Upload file

- Single: apache common upload file
- <https://commons.apache.org/proper/commons-fileupload/>
- Bỏ view ở ngoài webcontent r tải về thư viện mới nhất .zip trong java 2s copy vào lib
- <https://commons.apache.org/proper/commons-fileupload/apidocs/org/apache/commons/fileupload/disk/DiskFileItemFactory.html>
- Xem cách sd
- Tải thêm thư viện apache common io
 - Upload multiple file:
- Gỡ jquery upload multiple file mvc jsp servlet:
 <https://crunchify.com/spring-mvc-tutorial-how-to-upload-multiple-files-to-specific-location/>

11. Thêm dữ liệu từ file excel

- Apache POI
- Bỏ dòng 1
- Đuôi xls
- Download apache poi jar gỡ
- Load thêm thư viện mysql connector
- <https://poi.apache.org/spreadsheet/quick-guide.html>
- Đọc vào r chèn xuống mysql