

Prendo l'es. 2 e ne creo una web application (DYNAMIC WEB).

In WebContent / WEB-INF / lib c'è derbyclient (per usare il DB).

Costruisco una jsp dentro a WebContent chiamandola lista_studenti.jsp (scegliendo un html qualsiasi dato che poi si può modificare).

Un'applicazione può avere più client collegati quindi ci costruiamo un oggetto sessione (Sessione.java): la sessione viene costruita quando parte la sessione e l'oggetto viene ripreso durante la sessione (quindi andrà messo nella sessione web).

Creo costruttore e get della sessione. In studenti.jav vado a creare una "sessione"

Creiamo una ListaStidentiServlet.java

Per tutte le servlet bisogna aprire la connessione, svolgere SQL e poi chiudere la sessione quindi potrei fare una servlet generica in cui poi vado a cambiare solo il metodo (SQL) che uso durante la sessione.

Quindi potrei creare "DettagliStudenteServlet" con mapping "dettagli studente" (ma in questo esercizio non la creo; stessa cosa per rimuovi, inserisci, modifica, aggiorna).

Esempio: href = ...(qui ho una servlet)...

Ordine di creazione file:

- Applico modifiche (aggiungo metodi) alle classi dell'Es2: "Modello" (interfaccia) e "DatabaseManager" (classe che implementa l'interfaccia Modello). I file dell'Es2 riguardanti la "vista" qui non mi servono perché avrò il sito come "vista".
- Sessione.java: classe per separare la sessione dal "DatabaseManager.java" (se dovessi aggiungere qualcosa alla sessione, non vado a modificare il DatabaseManager).
- index.html: pagina di inizio.
- errore.html: pagina nel caso si presenti un errore.
- ServletListaStudenti.java
- lista_studenti.jsp: pagina (vista) che gestisce le operazioni sul DB.

ECLIPSE

Esercizio5

src

```
it.unipr.informatica.esercizio5
    Sessione.java
it.unipr.informatica.esercizio5.database
    DatabaseManager.java
it.unipr.informatica.esercizio5.modello
    Modello.java
    ModelloException.java
    Studente.java
it.unipr.informatica.esercizio5.web
    ListaStudentiServlet.java
configurazione.properties
```

WebContent

```
WEB-INF
    lib
        derbyclient.jar
    web.xml
lista_studenti.jsp
index.html
errore.html
```

Modello.java

... // Metodi aggiunti alla classe presa dall'Es2:

```
public Studente aggiungiStudente(String cognome, String nome) throws ModelloException;
public void rimuoviStudente(int matricola) throws ModelloException;
public Studente aggiornaStudente(Studente studente) throws ModelloException;
public Studente leggiStudente(int matricola) throws ModelloException;
```

...

DatabaseManager.java

... // Implementazioni dei metodi aggiunti alla classe

@Override

```
public Studente aggiungiStudente(String cognome, String nome) throws ModelloException {
    Connection connection = null;
    PreparedStatement statement = null;
    ResultSet resultSet = null;
    Studente risultato = null;

    try {
        connection = connetti();
        statement = connection.prepareStatement("INSERT INTO STUDENTI (COGNOME, NOME)
                                                VALUES (?, ?)", Statement.RETURN_GENERATED_KEYS);

        statement.setString(1, cognome);
        statement.setString(2, nome);
        resultSet = statement.executeQuery();

        if(resultSet.next()) {
            int matricola = resultSet.getInt(1);
            risultato = new Studente(matricola, cognome, nome);
        }
    } catch (Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(connection, statement, resultSet);
        throw new ModelloException("Impossibile aggiungere lo studente: " + cognome + " " + nome);
    }

    disconnetti(connection, statement, resultSet);
    return risultato;
}
```

@Override

```
public void rimuoviStudente(int matricola) throws ModelloException {
    Connection connection = null;
    PreparedStatement statement = null;

    try {
        connection = connetti();
        statement = connection.prepareStatement("DELETE FROM STUDENTI WHERE
                                                MATRICOLA = ?");

        statement.setInt(1, matricola);
        statement.execute();
    }
}
```

```

    } catch(Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(connection, statement, null);
        throw new ModelloException("Impossibile eliminare lo studente con matricola : " + matricola);
    }
}

```

@Override

```

public Studente aggiornaStudente(Studente studente) throws ModelloException {
    Connection connection = null;
    PreparedStatement statement = null;
    try {
        connection = connetti();
        statement = connection.prepareStatement("UPDATE STUDENTI SET COGNOME = ? AND
                                                NOME = ? WHERE MATRICOLA = ?");

        statement.setString(1, studente.getCognome());
        statement.setString(2, studente.getNome());
        statement.setInt(3, studente.getMatricola());
        statement.execute();
    } catch(Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(connection, statement, null);
        throw new ModelloException("Impossibile aggiornare lo studente: " + studente.getCognome() +
                                    " " + studente.getNome());
    }
    disconnetti(connection, statement, null);
    return studente;
}

```

@Override

```

public Studente leggiStudente(int matricola) throws ModelloException {
    Connection connection = null;
    PreparedStatement statement = null;
    ResultSet resultSet = null;
    Studente risultato = null;
    try {
        connection = DriverManager.getConnection(url);
        statement = connection.prepareStatement("SELECT * FROM STUDENTI WHERE
                                                MATRICOLA = ?");

        statement.setInt(1, matricola);
        resultSet = statement.executeQuery();
        if(resultSet.next())
            risultato = new Studente(matricola, resultSet.getString("COGNOME"),
                                      resultSet.getString("NOME"));
    } catch(Throwable throwable) {
        throwable.printStackTrace();
        disconnetti(connection, statement, resultSet);
        throw new ModelloException("Impossibile leggere lo studente di matricola: " + matricola);
    }
    disconnetti(connection, statement, resultSet);
    return risultato;
}

```

Sessione.java

```
package it.unipr.informatica.esercizio5;
import it.unipr.informatica.esercizio5.database.DatabaseManager;

public class Sessione {
    protected DatabaseManager databaseManager;
    public Sessione() {
        databaseManager = new DatabaseManager();
    }

    public DatabaseManager getDatabaseManager() {
        return databaseManager;
    }
}
```

lista_studenti.jsp

```
// Non posso mettere un import qui perché sono già dentro la classe quindi lo metto all'inizio:
<%@page import="it.unipr.informatica.esercizio5.Sessione"%>
<%@page import="it.unipr.informatica.esercizio5.modelo.Studente"%>
<%@page import="java.util.List"%>
<%@page import="it.unipr.informatica.esercizio5.database.DatabaseManager"%>
<%@page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Lista degli Studenti</title>
</head>
<body>
<%
    List<Studente> studenti = (List<Studente>)session.getAttribute("lista_studenti");
    session.removeAttribute("lista_studenti");

    String messaggio = (String)session.getAttribute("messaggio");
    if(messaggio != null) {
        session.removeAttribute("messaggio");
        // Ho la lista degli studenti; ora devo creare una tabella in html e quindi uscire da java
    %>
    <p><%=messaggio %></p>
    <%
        }
    %>
    <p><a href="aggiungi_studente.jsp">Aggiungi studente</a></p>
    <table>
    <tr><td></td><td></td><td></td><td>Matricola</td><td>Cognome</td><td>Nome</td></tr>
    <%
        for(Studente studente : studenti) { // Uso il for che genera la tabella con i dati
    %>
    <tr>
    // Aggiungo una colonna con un link che utilizza la matricola per poi dare informazioni:
```

```

<td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>">Dettagli</a></td>
<td><a href="modifica_studente?matricola=<%=studente.getMatricola() %>">Modifica</a></td>
<td><a href="rimuovi_studente?matricola=<%=studente.getMatricola() %>">Rimuovi</a></td>
<td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>"><%=studente.getMatricola()
%></a></td>
<td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>"><%=studente.getCognome()
%></a></td>
<td><a href="dettagli_studente?matricola=<%=studente.getMatricola() %>"><%=studente.getNome()
%></a></td>
</tr>
// Al posto di aprire i tag java ( <%...%> ) posso scrivere "out.print(...);"
<%
    }
%>
</table>
</body>
</html>

```

ListaStudentiServlet.java

package it.unipr.informatica.esercizio5.web;

import it.unipr.informatica.esercizio5.Sessione;

...

@SuppressWarnings("serial")

public class ListaStudentiServlet **extends** HttpServlet {

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

try {

HttpSession session = request.getSession();

Sessione sessione = (Sessione)session.getAttribute("sessione"); // Prendo l'attributo "sessione"

if(sessione == **null**) { // Se l'attributo "sessione" è vuoto:

sessione = **new** Sessione();

session.setAttribute("sessione", sessione); // "sessione" ora contiene al suo interno, un oggetto di classe "Sessione"

}

DatabaseManager databaseManager = sessione.getDatabaseManager(); // Estraggo da sessione il DatabaseManager

List<Studente> studenti = databaseManager.ricaricaStudenti(); // Uso un metodo del DatabaseManager per creare la lista degli studenti

session.setAttribute("lista_studenti", studenti); // Setto un nuovo attributo con nome "lista_studenti"

request.getRequestDispatcher("lista_studenti.jsp").forward(request, response); // Vado alla "lista_studenti.jsp" la quale potrà usufruire degli attributi ("sessione" e "lista_studenti") creati qui

} **catch**(Throwable throwable) {

request.getRequestDispatcher("errore.html").forward(request, response);

}

}

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    doGet(request, response);
}
}
```

index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Esercizio 5</title>
</head>
<body>
<a href="lista_studenti">Lista degli studenti</a>
</body>
</html>
```

errore.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Esercizio 5</title>
</head>
<body>
<h1>Errore Fatale</h1>
Contattare l'amministratore dell'applicazione.
</body>
</html>
```