

Sistemas Web

CURSO 2021/2022

PASO 2.- Nuevo proyecto e introducir clases para gestionar la base de datos



Web Sistemak by [Oskar Casquero](#) & [María Luz Álvarez](#) is licensed under a [Creative Commons Reconocimiento 4.0 Internacional License](#).

NUEVO PROYECTO

- Crear nuevo proyecto Web (Dynamic Web Project).
 - Nombre del proyecto: **ShareInfo**.
 - Recordar que el proyecto tiene que tener el fichero de despliegue **web.xml**
- En la carpeta **ShareInfo/WebContent** crear las carpetas *css*, *html* y *jsp*.
- En la carpeta **ShareInfo/WebContent/WEB-INF/lib** meter las librerías (.jar) necesarias para trabajar con el proyecto:
 - **gson** - librería Java para convertir objetos java en JSON y viceversa.
 - **mysql-connector** - librería para el acceso a la base de datos.

CLASES PARA GESTIONAR LA BASE DE DATOS: **MySQLdb.java**

- En la carpeta **ShareInfo/Java Resources/src** estará la clase para gestionar la base de datos.
 - Nombre de la clase: **MySQLdb.java**
 - Paquete: **helper.db** → **crear el paquete**
 - **Copiar** la clase del MySQLdb.java de SHAREINFO.7z
- **Atributos de la clase MySQLdb:**
 - `private String url = "jdbc:mysql://localhost:3306/";`
 - `private String user = "root";`
 - `private String passwd = "XXXXX";` XXXXX password del usuario root.
 - `private String driver = "com.mysql.jdbc.Driver";`
 - `private Connection conn;`
- **Métodos de la clase MySQLdb:**
 - `public MySQLdb()` → crea y guarda la conexión con la base de datos.
 - `public void setUserInfo (String email, String password, String username)` → añade un nuevo usuario en la tabla users.
 - `public String getUsername (String email, String password)` → obtiene el nombre de un usuario de la tabla users.
 - `public void setMessageInfo (MessageInfo message)` → añade un mensaje nuevo en la tabla messages.
 - `public ArrayList <MessageInfo> getAllMessages()` → lista de todos los mensajes de la tabla message.

MySQLdb

Es una **clase normal** no es un Servlet

```
package helper.db;
```

```
import java.util.*;  
import java.sql.*;
```

```
import helper.info.*;
```

```
public class MySQLdb {  
    private String url = "jdbc:mysql://localhost:3306/";  
    private String user = "root";  
    private String passwd = "root";  
    private String driver = "com.mysql.jdbc.Driver";  
  
    private Connection conn;
```

java.sql librería para gestionar el acceso a la base de datos

helper.info importar paquete donde se define la clase `MessageInfo.java` para gestionar mas fácil los mensajes. Lo veremos en las siguientes transparencias

Atributos de la clase

El `passwd` debe ser el de tu usuario root

```
public MySQLdb() {  
    try {  
        Class.forName(this.driver).newInstance();  
        this.conn = DriverManager.getConnection(this.url, this.user, this.passwd);  
    } catch (Exception e) {  
        System.out.println("Exception: " + e.getMessage());  
    }  
}
```

Constructor de la clase: crear una conexión con la base de datos

```
public void setUserInfo(String email, String password, String username) {..  
  
public String getUsername(String email, String password) {..  
  
public void setMessageInfo(String message, String username) {..  
  
public ArrayList<MessageInfo> getAllMessages() {..  
  
}
```

Métodos de la clase.

Vemos en siguiente transparencia

El método *setUserInfo()* AÑADE un nuevo usuario en la tabla users.

```
public void setUserInfo(String email, String password, String username) {  
    String query = "INSERT INTO shareinfo.users VALUES ('" + email + "', '" + password + "', '" + username + "')";  
    System.out.println("    DB query: " + query);  
    try {  
        Statement st = this.conn.createStatement();  
        st.executeUpdate(query);  
    } catch (Exception e) {  
        System.out.println("Exception: " + e.getMessage());  
    }  
}
```

Aplica la consulta

Consulta SQL que crea en la tabla users de shareinfo un nuevo registro con los valores indicados

El método *getUsername()* BUSCA el nombre de un usuario dando el email y passwd

```
public String getUsername(String email, String password) {  
    String query = "SELECT username FROM shareinfo.users WHERE email='" + email + "' AND password='" + password + "'";  
    System.out.println("    DB query: " + query);  
    String username = null;  
    try {  
        Statement st = this.conn.createStatement();  
        ResultSet res = st.executeQuery(query);  
        while(res.next()) {  
            username = res.getString("username");  
        }  
    } catch (Exception e) {  
        System.out.println("Exception: " + e.getMessage());  
    }  
    return username;  
}
```

Consulta SQL que busca en la tabla users de shareinfo los registros con el email y el passwd indicados

Devuelve el username

El método *setMessageInfo()* AÑADE un mensaje nuevo en la tabla messages

```
public void setMessageInfo(String message, String username) {  
    String query = "INSERT INTO shareinfo.messages VALUES ('0', '" + message + "', '" + username + "')";  
    System.out.println("DB query: " + query);  
    try {  
        Statement st = this.conn.createStatement();  
        st.executeUpdate(query);  
    } catch (Exception e) {  
        System.out.println("Exception: " + e.getMessage());  
    }  
}
```

Consulta SQL que crea en la tabla messages de shareinfo un nuevo registro con los valores indicados

Aplica la consulta

El método *getAllMessages()* LISTA de todos los mensajes de la tabla message.

```
public ArrayList<MessageInfo> getAllMessages() {  
    String query = "SELECT * FROM shareinfo.messages;";  
    System.out.println("DB query: " + query);  
    ArrayList<MessageInfo> messageInfoList = new ArrayList<MessageInfo>();  
    try {  
        Statement st = this.conn.createStatement();  
        ResultSet res = st.executeQuery(query);  
        while(res.next()) {  
            messageInfoList.add(new MessageInfo(res.getString("message"), res.getString("username")));  
        }  
    } catch (Exception e) {  
        System.out.println("Exception: " + e.getMessage());  
    }  
    return messageInfoList;  
}
```

Consulta SQL que selecciona toda la tabla de mensajes

Devuelve el una Lista de MessageInfo

CLASES PARA GESTIONAR LA BASE DE DATOS: **MessageInfo.java**

- En la carpeta **ShareInfo /Java Resources/src** estará la clase para gestionar los mensajes.
 - Nombre de la clase: **MessageInfo.java**
 - Paquete: **helper.info** → **crear el paquete**
 - **Copiar** la clase MessageInfo.java de SHAREINFO.7z
- **Atributos de la clase MessageInfo:**
 - private String message;
 - private String username;
- **Métodos de la clase MessageInfo:**
 - public MessageInfo(String message, String username)
 - public String getMessage()
 - public String getUsername()

MesaggeInfo

```
package helper.info;
```

MesaggeInfo clase que facilita el acceso a la tabla de mensajes

```
public class MessageInfo {  
    private String message;  
    private String username;
```

Atributos de la clase

```
    public MessageInfo(String message, String username) {  
        this.message = message;  
        this.username = username;  
    }
```

*Constructor de la clase.
Agrupa el mensaje con el usuario*

```
    public String getMessage() {  
        return this.message;  
    }
```

```
    public String getUsername() {  
        return this.username;  
    }
```

Métodos de la clase.

```
}
```