

タイトル	
DAO を使ってプログラムを作ってみよう！	
レベル	★★★★★

ログイン処理を DAO を使って書いてみましょう！

今回作るプログラムは・・・

ログイン画面

メアド	<input type="text"/>
パスワード	<input type="password"/>
send	<input type="button" value="send"/>

↓（ログイン成功すると）

トップ画面

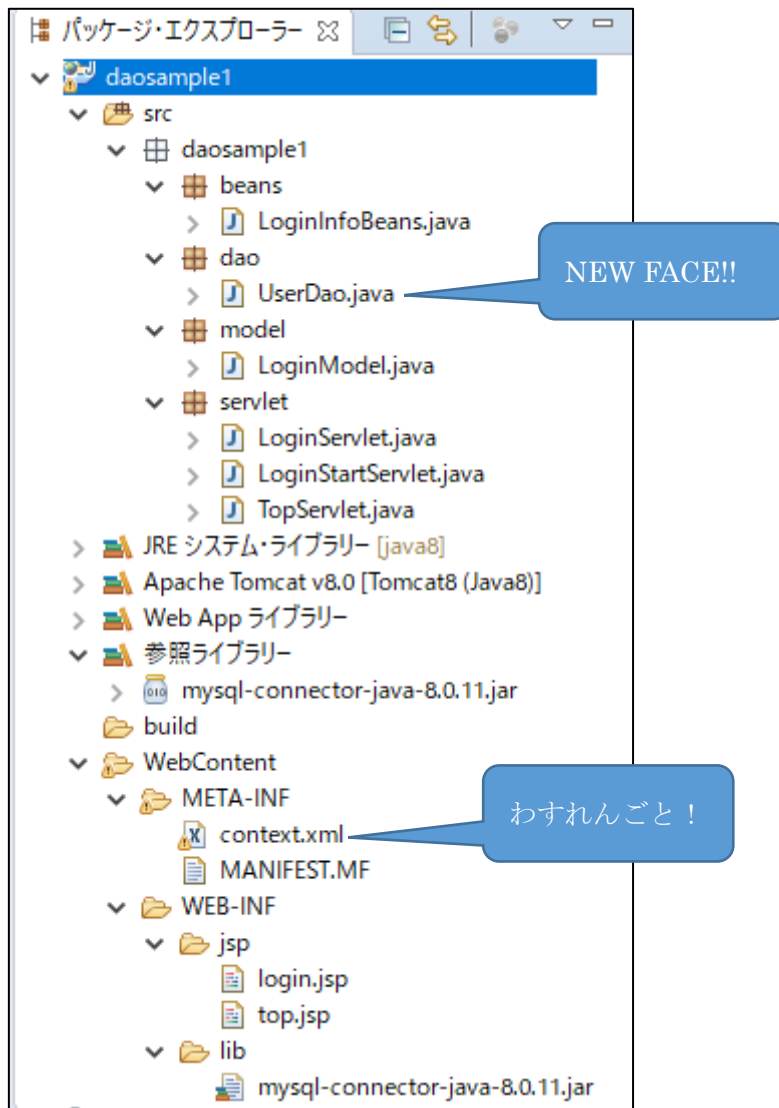
トップ画面です

今回の DAO を使うことでより実践的なソースコードになります！

まずは、見よう見まねで書いてみましょう！

※前回のガチャを使いまわすと早く出来上がります。

プロジェクトの構成は以下のような感じになります！



次ページに各クラス、ファイルの説明があります。

クラス名/ファイル名	概要	URL 割り当て
LoginInfoBeans	ログイン情報を保持するビーンズ	なし
UserDao	期待のニューフェース。 ユーザーテーブルに関する DB アクセスをここに記載する	なし
LoginModel	ログイン処理を行うモデル 内部で UserDao を使ってユーザー情報を取得する	なし
LoginServlet	ログイン処理を行うサーブレット 内部で LoginModel を使ってログイン処理をする	auth
LoginStartServlet	ログイン画面を表示するサーブレット	login
TopServlet	トップ画面を表示するサーブレット	top
content.xml	接続情報が記載されている設定ファイル	なし
login.jsp	ログイン画面の JSP ファイル	
top.jsp	トップ画面の JSP ファイル	

次ページよりソースコードを記載しています！

## LoginInfoBeans (ガチャから変更なし)

```
package daosample1.beans;
import java.io.Serializable;
public class LoginInfoBeans implements Serializable {
    private String name;
    private String mail;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getMail() {
        return mail;
    }
    public void setMail(String mail) {
        this.mail = mail;
    }
}
```

## UserDao (ニューハーフ)

```
package daosample1.dao;

import java.sql.Connection;

public class UserDao {

    private Connection con = null;

    /**
     * DB接続を行う
     * @throws Exception 接続失敗 (本来であれば自作の例外を飛ばす方が良い)
     */
    public void connect() throws Exception{

        if( con != null ){
            //すでに接続済みの場合は何もしない
            return;
        }

        InitialContext context = null;

        try{
            //////////////////////////////////////
            //DBの接続
            String resourceName = "jdbc/MySQL";
            String jndi = "java:comp/env/"+resourceName;

            context = new InitialContext();

            DataSource dataSource = (DataSource) context.lookup(jndi);

            con = dataSource.getConnection();

        }catch(NamingException e) {
            //エラー発生した場合にコンソールにログを出力する
            e.printStackTrace();
            throw e;
        }catch(SQLException e) {
            //エラー発生した場合にコンソールにログを出力する
            e.printStackTrace();
            throw e;
        }

    }

    /**
     * 接続のクローズ処理
     */
    public void close(){

        if( con != null ){
            try {
                con.close();
                con = null;
            } catch (SQLException e) {
                //クローズの失敗は、ログを出力するだけ
                System.out.println("closeに失敗しました");
            }
        }

    }

}
```

※次ページに続きます！

```

/**
 * ユーザーDBよりユーザー情報を取得する
 * (ログイン処理と等価)
 *
 * @param mail メールアドレス
 * @param password パスワード
 * @return
 */
public LoginInfoBeans getBy(String mail,String password){

    if( con == null){
        //接続していない場合はエラーとする
        return null;
    }
    LoginInfoBeans loginInfo = null;

    PreparedStatement stmt = null;
    ResultSet rs = null;

    try{

        ////////////////////////////////////
        //SELECT文の発行
        stmt = con.prepareStatement("SELECT * FROM user_tbl WHERE mail=? AND password=?");

        stmt.setString(1, mail);
        stmt.setString(2, password);
        rs = stmt.executeQuery();

        ////////////////////////////////////
        //DBから値を取得
        while( rs.next() ){
            loginInfo = new LoginInfoBeans();

            loginInfo.setMail(rs.getString("mail"));
            loginInfo.setName(rs.getString("name"));
        }

    }catch(SQLException e) {
        //エラー発生した場合にコンソールにログを出力する
        e.printStackTrace();
    }
    finally {
        try {
            // 接続を閉じる
            if( rs != null ){
                rs.close();
            }
            if( stmt != null ){
                stmt.close();
            }
        } catch (SQLException e) {
            System.out.println("closeに失敗しました");
            ; //closeの失敗は無視
        }
    }

    return loginInfo;
}

```

## LoginModel (ガチャから変更あり)

```
package daosample1.model;

import daosample1.beans.LoginInfoBeans;

public class LoginModel {

    /**
     * ログイン処理を行う
     * 引数でもらったmailとパスワード
     *
     * @param mail
     * @param password
     * @return null: ログイン失敗 非null: ログイン成功
     */
    public LoginInfoBeans login(String mail, String password){
        LoginInfoBeans loginInfo = null;

        UserDao userDao = new UserDao();
        try{
            //////////////////////////////////////
            //DBの接続
            userDao.connect();

            //////////////////////////////////////
            //ユーザー情報の取得
            loginInfo = userDao.getBy(mail, password);

        }catch(Exception e) {
            //エラー発生した場合にコンソールにログを出力する
            e.printStackTrace();
        }
        finally {
            //接続(コネクション)を閉じる
            if(userDao!=null) {
                userDao.close();
            }
        }

        return loginInfo;
    }
}
```

### LoginServlet (ガチャから変更なし)

```
package daosample1.servlet;

import java.io.IOException;

@WebServlet("/auth")
public class LoginServlet extends HttpServlet {

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

        ////////////////////////////////////////////////////
        //JSPからメールアドレスとパスワードを取得
        String mail = request.getParameter("mail");
        String password = request.getParameter("password");

        ////////////////////////////////////////////////////
        //Modelを呼び出しDBの値をメール、パスワードを照合する
        LoginModel loginModel = new LoginModel();

        LoginInfoBeans loginInfo = loginModel.login(mail, password);

        if( loginInfo != null ){
            ////////////////////////////////////////////////////
            //ログイン結果をセッションに保存する
            HttpSession session = request.getSession(true);

            session.setAttribute("loginInfo", loginInfo);
        }else{
            //ログイン結果がnullの場合はログイン画面に戻す
            response.sendRedirect("login");
            return;
        }

        ////////////////////////////////////////////////////
        //画面を転送する (リダイレクト)
        response.sendRedirect("top");
    }
}
```

### LoginStartServlet (ガチャから変更なし)

```
package daosample1.servlet;

import java.io.IOException;

@WebServlet("/login")
public class LoginStartServlet extends HttpServlet {

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

        RequestDispatcher rq = request.getRequestDispatcher("WEB-INF/jsp/login.jsp");
        rq.forward(request, response);
    }
}
```



TopServlet (ガチャから変更なし)

```
package daosample1.servlet;

import java.io.IOException;

@WebServlet("/top")
public class TopServlet extends HttpServlet {

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

        RequestDispatcher rq = request.getRequestDispatcher("WEB-INF/jsp/top.jsp");
        rq.forward(request, response);
    }
}
```

conext.xml (ガチャから変更なし)

```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
    <Resource
        name="jdbc/MySQL"
        auth="Container"
        type="javax.sql.DataSource"
        maxTotal="100"
        maxIdle="20"
        maxWaitMillis="-1"
        username="root"
        password="password"
        driverClassName="com.mysql.jdbc.Driver"
        url="jdbc:mysql://localhost:3306/webtestdb?serverTimezone=JST"
        validationQuery="SELECT 0"
    />
</Context>
```

※password の部分は「MySQL に接続するときのパスワード」に変更すること

### login.jsp (ガチャから変更なし)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>ログイン</title>
</head>
<body>
    <form action="auth" method="POST">
        <table border="1">
            <tr>
                <td>メアド</td>
                <td><input type="text" name="mail"></td>
            </tr>
            <tr>
                <td>パスワード</td>
                <td><input type="password" name="password"></td>
            </tr>
            <tr>
                <td colspan="2"><input type="submit" value="send"></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

### top.jsp (ガチャから変更あり)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
    pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>トップ画面</title>
</head>
<body>
    トップ画面です
</body>
</html>
```