DEZVOLTAREA APLICATIILOR WEB CURS 7

Lect. Univ. Dr. Mihai Stancu

DEPARTAMENTUL DE INFORMATICĂ

Suport de curs

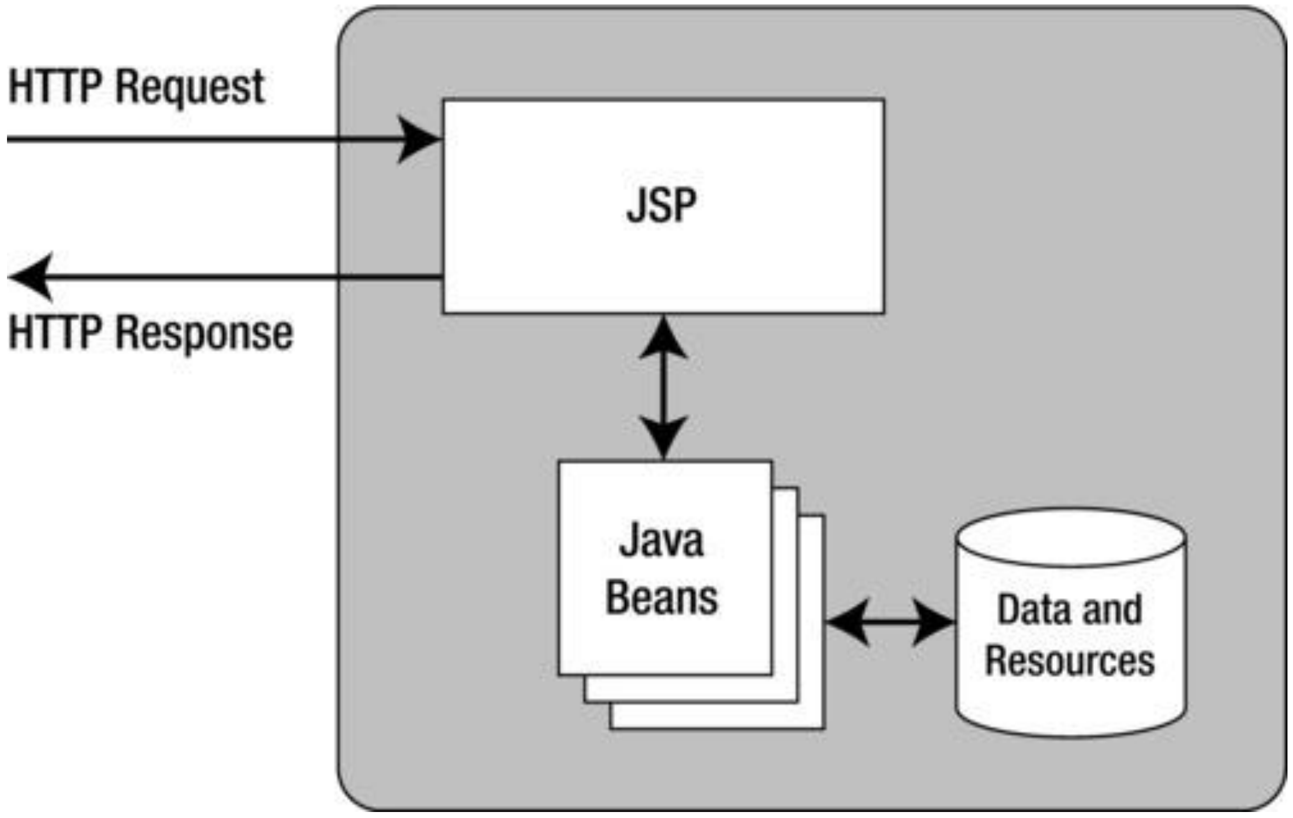
- suport (Beginning JSP, JSF and Tomcat)
  - Capitolul 3 – JSP Application Architectures

Arhitectura aplicatiei JSP

- Cresterea complexitatii
- Eficienta in dezvoltare
- Mentinerea aplicatiei
- Logica de business vs. design
- Scop: aplicatii mai stabile si mai usor de intretinut

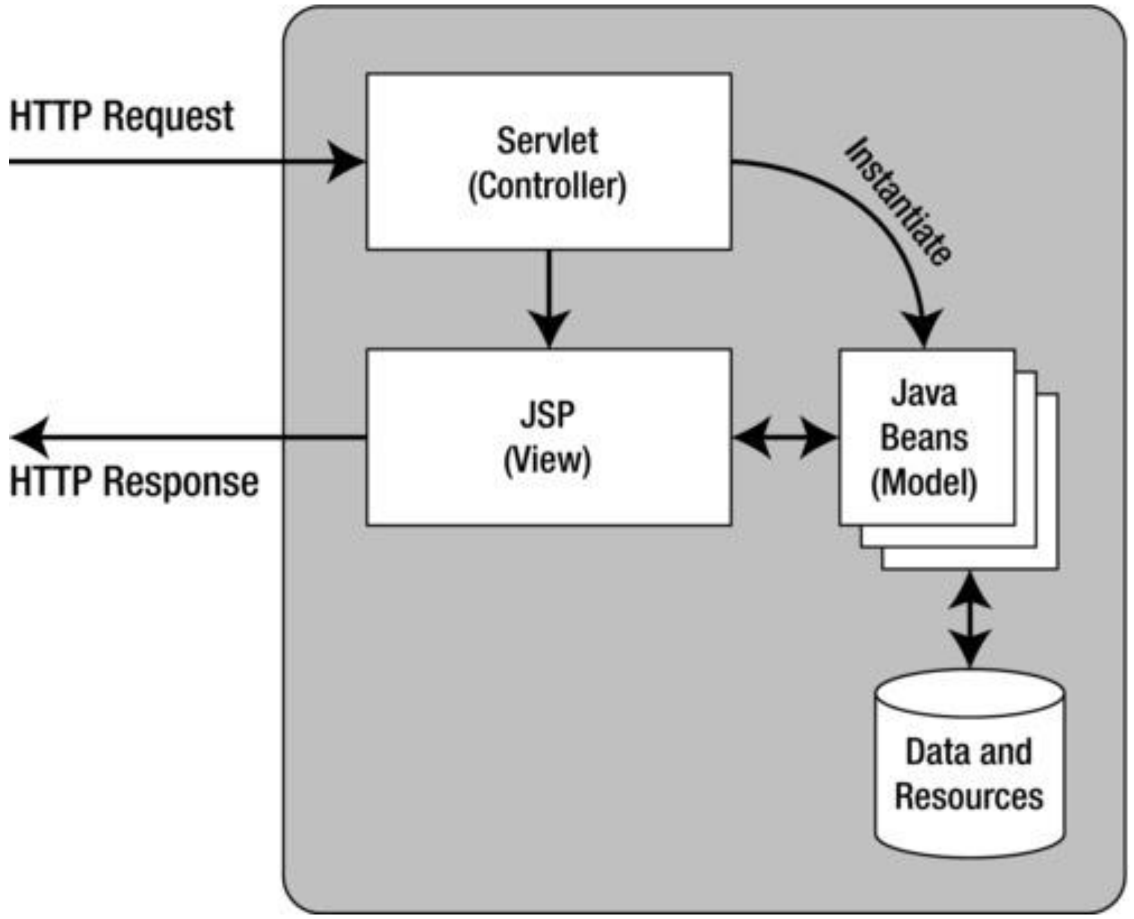
Departamentul de Informatică

ARHITECTURA STANDARD



- Neajunsuri?

Arhitectura MVC



- Model – View – Controller
- Avantaje?

Arhitectura MVC – Cerere catre Controller

- LoginPage.jsp

```
<%@ page language="java" contentType="text/html; charset=windows-1256"
pageEncoding="windows-1256"%>
<%
response.setHeader("Cache-Control", "no-store, must-revalidate");
response.setHeader("Pragma", "no-cache");
response.setDateHeader("Expires", -1);
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
     <meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
     <title>Login Page</title>
</head>
<body>
     <form action="LoginServlet" method="post">
          Please enter your user name <input type="text" name="un" id="un" /><br>
          Please enter your password <input type="text" name="pw" id="pw" />
          <input type="submit" value="submit">
     </form>
</body>
</html>
```

- Cererea adresata catre Controller (Java Servlet)

Departamentul de Informatică

Arhitectura MVC – Servlet as Controller

- LoginServlet.java

```
...
public class LoginServlet extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, java.io.IOException {
          try {
              UserBean user = new UserBean();
              user.setUserName(request.getParameter("un"));
              user.setPassword(request.getParameter("pw"));
              user = UserDAO.login(user);

              if (user.isValid()) {
                    HttpSession session = request.getSession(true);
                    session.setAttribute("currentSessionUser", user);
                    response.sendRedirect("userLogged.jsp"); // logged-in page
              }
              else
                    response.sendRedirect("invalidLogin.jsp"); // error page
          }
          catch (Throwable theException) {
              System.out.println(theException);
          }
    }
}
```

- Interogare Model pentru obtinerea informatiilor
- Prelucrarea infromatiilor
- Redirectionarea catre View

Departamentul de Informatică

Arhitectura MVC – Model – DAO Class

- UserDao.java

```
...
public class UserDAO {
    static Connection currentCon = null;
    static ResultSet rs = null;

    public static UserBean login(UserBean bean) {
          ...
          String searchQuery = "select * from users where username='" +
bean.getUsername() + "' AND password='" + bean.getPassword() + "'";
          try {
              currentCon = ConnectionManager.getConnection();
              stmt = currentCon.createStatement();
              rs = stmt.executeQuery(searchQuery);
              // GET THE DATA
             ...
              // CLOSE THE DB CONNECTION
             ...
          return bean;
    }
}
```

- Data Access Object (maparea entitatilor din DB)
    - Conectarea la DB
    - Obtinerea informatiilor
    - Return Java Bean

Departamentul de Informatică

Arhitectura MVC – Model – Config Manager

- ConnectionManager.java

```
...
public class ConnectionManager {
    static Connection con;
    static String url;

    public static Connection getConnection() {
          try {
              String url = "jdbc:mysql://localhost/TEST";
              Class.forName("com.mysql.jdbc.Driver");
              try {
                    con = DriverManager.getConnection(url, "root", "*******");
              } catch (SQLException ex) {
                    ex.printStackTrace();
              }
          } catch (ClassNotFoundException e) {
              System.out.println(e);
          }
          return con;
    }
}
```

- JDBC (conform versiune DB)

Departamentul de Informatică

Arhitectura MVC – Model – Java Bean

- UserBean.java

```
...
public class UserBean {
    private String username;
    private String password;
    private String firstName;
    private String lastName;
    public boolean valid;

    public String getFirstName() {
       return firstName;
    }
    public void setFirstName(String newFirstName) {
       firstName = newFirstName;
    }

    public String getLastName() {
       return lastName;
    }
    public void setLastName(String newLastName) {
       lastName = newLastName;
    }
    ...
```

- Java Bean – incapsulare informatii “user”

Arhitectura MVC – View – JSP Page

- userLogged.jsp

```
<%@ page language="java" contentType="text/html; charset=windows-1256"
     pageEncoding="windows-1256" import="ExamplePackage.UserBean"%>
<%
response.setHeader("Cache-Control","no-store,must-revalidate");
response.setHeader("Pragma","no-cache");
response.setDateHeader ("Expires", -1);

if(session.getAttribute("currentSessionUser") != null) {
%>
<html>
<head>
     <meta http-equiv="Content-Type" content="text/html; charset=windows-1256">
     <title>   User Logged Successfully   </title>
     </head>
     <body onload="noBack();">
        <a href="LogoutServlet">Logout</a>
<% UserBean currentUser = (UserBean)(session.getAttribute("currentSessionUser"));%>
        Welcome <%= currentUser.getFirstName() + " " + currentUser.getLastName() %>
      </body>
</html>
<%
} else {
     response.sendRedirect("LoginPage.jsp");
}
%>
```

- Preluare informatii din context pregatite de Controller (Servlet)

Departamentul de Informatică

Cuvinte cheie

- MVC
- servlet
- JSP
- Java Bean
- context

Resurse utile

- https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller
- http://www.thejavageek.com/2013/08/11/mvc-architecture-with-servlets-and-jsp/