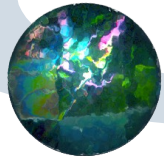
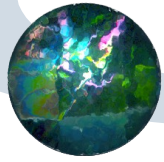


# サーブレット



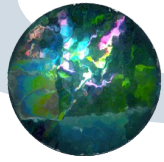
# サーブレット (Servlet) とは

- Webサーバ側で動作するJavaプログラム
  - 通常はapache等のバックグラウンドで動作する
  - Servletコンテナ上にアプリケーションを配置
- 代表的なServletコンテナ
  - Apache Tomcat
  - WebLogic
  - WebSphere
  - GlassFish



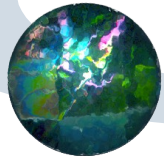
# phpやCGIとどこが違う

- 高レベルのプログラミング言語
  - ロジックの正確な実装
- MVCを分離した開発が可能
  - 開発効率
- DBやファイルシステムとの連携
- Webサーバとコンテナの分離が可能
  - セキュリティ
  - パフォーマンス



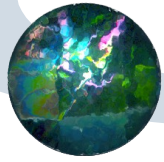
# Framework

- MVCの分離
  - View(入出力):JSP (Java Server Page)
  - Model(データとロジック):Java Beans
  - Control(ページ遷移):様々なフレームワーク



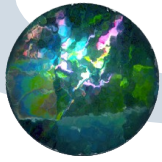
# 代表的なFramework

- Spring:
  - <http://www.springsource.com>
- Java Server Faces
  - <http://java.sun.com/j2ee/javaxserverfaces/>
- Apache Struts
  - <http://struts.apache.org/>
- WebObjects
  - <http://www.apple.com/webobjects/>



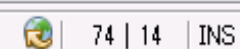
# Java Server Faces

- 入出力にJSPを使用
- ページ遷移をXMLで制御
  - faces-config.xml中のnavigation-ruleタグ
- データ保持
  - 管理対象Beans
  - スコープ : application, session, request, none
  - JSPからsetterとgetterを使って値へアクセス

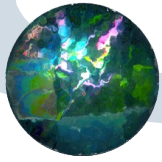


# ページ間の遷移

- ボタンを押したときの文字列で移動先を決定
  - faces-config.xml
- 対応するjavaでは
  - 戻り値が文字列のメソッド
  - 引数は無し

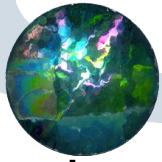




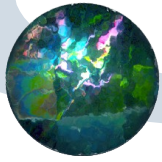


## 各ページ

- JSPページ
- 二つのタグ
  - JSF Core Tags
  - JSF HTML Tags
- バインディング
  - 管理Beansに対して、`get`や`set`を使っている



```
<body>
  <%@ include file="header.jspf" %>
  <f:view>
    <h:form>                                管理beansから sessionData.getClientIP()で取得
      You are accessing from
      <h:outputText value = "#{sessionData.clientIP}"/>.
      <br/>                                    sessionData.setUser()で設定
      <h:inputText label="ユーザ名" size="8"
                    value = "#{sessionData.user}" required="true" >
        <f:validateLength minimum="1" maximum="8"/>
      </h:inputText>
      <h:commandButton action="#{sessionData.login}"
                       value = "ログイン"/>  sessionData.login()メソッドを使用
    </h:form>
  </f:view>
</body>
```



# 管理beans

- JSPページから参照できるjavaクラス
  - faces-config.xml内に記載
  - 新規ファイルとして「管理Beans」を選ぶと、自動で追加される。

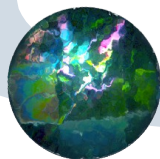
<managed-bean>

<managed-bean-name>sessionData</managed-bean-name>

<managed-bean-class>data.SessionData</managed-bean-class>

<managed-bean-scope>session</managed-bean-scope>

</managed-bean>



# アプリケーション用パラメタ

- web.xmlに「コンテキストパラメタ」として定義
- JSPページ内での参照
  - `application.getInitParameter(パラメタ名)`
- Java内での参照
  - `javax.faces.context.FacesContext.getCurrentInstance().getExternalContext().getInitParameter(パラメタ名)`

index.jsp

```
<%@page contentType="text/html"%>
<%@page pageEncoding="UTF-8"%>

<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>

<!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=UTF-8"/>
    <title><%=application.getInitParameter("title")%></title>
  </head>
  <body>
    <%@ include file="header.jspf" %>
    <f:view>
      <h:form>
        You are accessing from
        <h:outputText value = "#{sessionData.clientIP}"/>.
        <br/>
        <h:inputText label="ユーザ名" size="8"
                     value = "#{sessionData.user}"
                     required="true" >
          <f:validateLength minimum="1" maximum="8"/>
        </h:inputText>
        <h:commandButton action="#{sessionData.login}"
                         value = "ログイン"/>
      </h:form>
    </f:view>
  </body>
</html>
```

menu.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<f:view>
  <html>
    <head>
      <meta http-equiv="Content-Type"
        content="text/html; charset=UTF-8"/>
      <title><%=application.getInitParameter("title")%></title>
    </head>
    <body>
      <%@ include file="header.jspf" %>
      <h:form>
        <h:dataTable value="#{sessionData.staffList}" var="staff"
          binding="#{sessionData.staffData}"
          border="1">
          <h:column>
            <f:facet name="header">
              <h:outputText value="ID"/>
            </f:facet>
            <h:outputText value="#{staff.staff_id}"/>
          </h:column>
          <h:column>
            <f:facet name="header">
              <h:outputText value="氏名"/>
            </f:facet>
            <h:outputText value="#{staff.name}"/>
          </h:column>
          <h:column>
            <f:facet name="header">
              <h:outputText value="権限"/>
            </f:facet>
            <h:outputText value="#{staff.role.name}"/>
          </h:column>
          <h:column>
            <f:facet name="header">
              <h:outputText value="処理"/>
            </f:facet>
            <h:commandButton
```

menu.jsp

```
                action="#{sessionData.selectStaff}"
                value="選択"/>
            </h:column>
        </h:dataTable>
    </h:form>
</body>
</html>
</f:view>
```

editStaff.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@taglib prefix="f" uri="http://java.sun.com/jsf/core"%>
<%@taglib prefix="h" uri="http://java.sun.com/jsf/html"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<f:view>
    <html>
        <head>
            <meta http-equiv="Content-Type"
                content="text/html; charset=UTF-8"/>
            <title><%=application.getInitParameter("title")%></title>
        </head>
        <body>
            <%@ include file="header.jspf" %>
            <h:form>
                <h:panelGrid columns="2"border="1"
                    columnClasses="rightAlign, leftAlign">
                    <h:outputText value="有効・無効"/>
                    <h:selectBooleanCheckbox
                        value="#{sessionData.staff.valid}"/>

                    <h:outputText value="氏名"/>
                    <h:inputText value="#{sessionData.staff.name}"/>

                    <h:outputText value="権限"/>
                    <h:selectOneListbox
                        value="#{sessionData.staff.role}"
                        <f:selectItems value="#{sessionData.roleList}"/>
                    </h:selectOneListbox>

                    <h:outputText value="登録日"/>
                    <h:outputText value="#{sessionData.staff.reg_date}"/>

                    <h:outputText value="備考"/>
                    <h:inputTextarea
                        value="#{sessionData.staff.description}"/>
                </h:panelGrid>
            </h:form>
        </body>
    </html>
</f:view>
```



SessionData.java

```
/**
 *
 * @author tadaki
 */
package data;

import java.io.File;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.faces.component.UIData;
import javax.faces.context.ExternalContext;
import javax.faces.context.FacesContext;
import javax.servlet.http.HttpServletRequest;
import xmlData.Role;

public class SessionData {

    ExternalContext exCon = FacesContext.getCurrentInstance().
        getExternalContext();
    HttpServletRequest servletRequest =
        (HttpServletRequest) exCon.getRequest();
    private String clientIP = "unknown";
    private String user;
    private String validuser;
    private xmlData.AbstractData db;
    private List<xmlData.Role> roleList;
    private List<data.StaffWithRole> staffList;
    private int maxStaff = 0;
    private data.StaffWithRole staff = null;
    private UIData staffData = null;

    /** Creates a new instance of SessionData */
    public SessionData() {
        clientIP = servletRequest.getRemoteAddr();
        validuser = exCon.getInitParameter("adminUser");
    }

    /** ***** ボタンの動作など ***** */
    public String login() {
        String response = "LOGIN";
    }
}
```

SessionData.java

```
        System.err.println(validuser);
        if (!user.equals(validuser)) {
            response = "ERROR";
        } else {
            try {
                readData();
            } catch (Exception ex) {
                Logger.getLogger(SessionData.class.getName()).
                    log(Level.SEVERE, null, ex);
                response = "ERROR";
            }
        }
        return response;
    }

    public String selectStaff() {
        staff = (data.StaffWithRole) staffData.getRowData();
        return "EDITSTAFF";
    }

    public String updateStaff() {
        String s = "UPDATE";
        if (db == null) {
            s = "ERROR";
            return s;
        }
        try {
            db.updateStaff(staff.getStaff());
        } catch (Exception ex) {
            s = "ERROR";
            Logger.getLogger(SessionData.class.getName()).
                log(Level.SEVERE, null, ex);
        }
        return s;
    }

    /*****
private void readData() throws Exception {
    String filename = exCon.getInitParameter("dataFolder");
    System.err.println(filename);
    File file = new File(filename);
    db = new xmlData.XMLData(file.getPath());
    db.connect();
    db.getRoles();
    *****/
```

SessionData.java

```
db.getStaffs();
HashMap<Integer, xmlData.Role> roleMap = db.getRoleMap();
roleList = Collections.synchronizedList(
    new ArrayList<xmlData.Role>());
for (xmlData.Role r : roleMap.values()) {
    roleList.add(r);
}
staffList = Collections.synchronizedList(
    new ArrayList<data.StaffWithRole>());
for (xmlData.Staff s : db.getStaffList()) {
    staffList.add(new data.StaffWithRole(s,
        roleMap.get(s.getRole())));
}
getMaxStaff(staffList);
}

private void getMaxStaff(List<data.StaffWithRole> staffList) {
    maxStaff = 0;
    for (data.StaffWithRole s : staffList) {
        maxStaff = Math.max(maxStaff, s.getStaff_id());
    }
}

/***** 設定及び取得メソッド *****/
public String getClientIP() {
    return clientIP;
}

public String getUser() {
    return user;
}

public void setUser(String user) {
    this.user = user;
}

public StaffWithRole getStaff() {
    return staff;
}

public void setStaff(StaffWithRole staff) {
    this.staff = staff;
}
```

SessionData.java

```
public UIData getStaffData() {  
    return staffData;  
}  
  
public void setStaffData(UIData staffData) {  
    this.staffData = staffData;  
}  
  
public List<data.StaffWithRole> getStaffList() {  
    return staffList;  
}  
  
public List<Role> getRoleList() {  
    return roleList;  
}  
}
```

## StaffWithRole.java

```
/**
 *
 * @author tadaki
 */
package data;

import xmlData.*;
import java.util.Date;

public class StaffWithRole {

    private xmlData.Staff staff;
    private xmlData.Role role;

    public StaffWithRole(final xmlData.Staff staff, xmlData.Role role) {
        this.staff = staff;
        this.role = role;
    }

    public xmlData.Staff getStaff() {
        return staff;
    }

    public int getStaff_id() {
        return staff.getStaff_id();
    }

    public String getName() {
        return staff.getName();
    }

    public Role getRole() {
        return role;
    }

    public void setRole(Role role) {
        this.role = role;
        staff.setRole(role.getRole_id());
    }

    public boolean isValid() {
        return staff.isValid();
    }
}
```

StaffWithRole.java

```
public void setValid(boolean valid) {
    staff.setValid(valid);
}

public Date getReg_date() {
    return staff.getReg_date();
}

public void setReg_date(Date reg_date) {
    staff.setReg_date(reg_date);
}

public String getDescription() {
    return staff.getDescription();
}

public void setDescription(String description) {
    staff.setDescription(description);
}
}
```