
ECサイト初級

ユーザ登録機能の作成

4 時間目

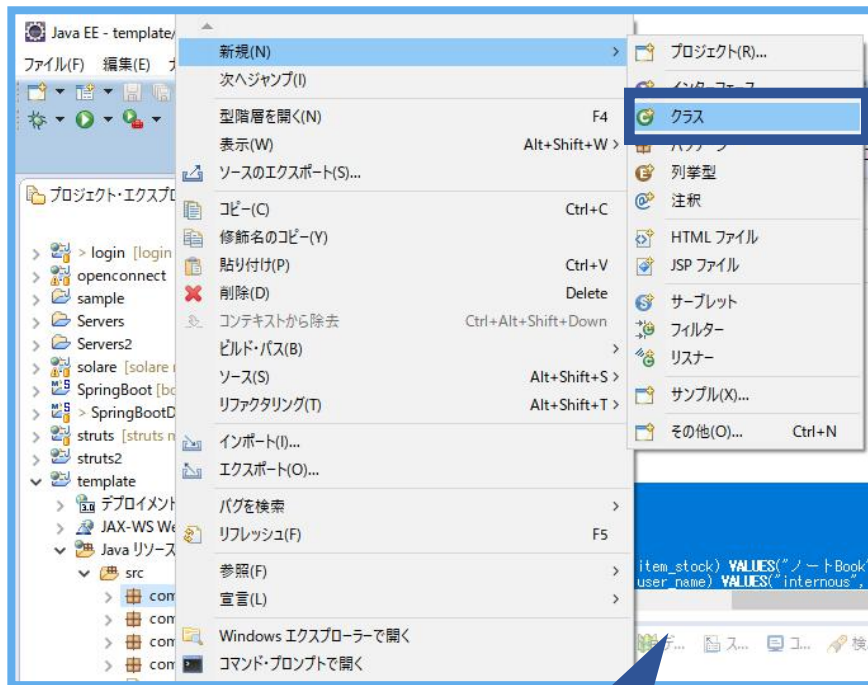
作業目次

作成機能一覧

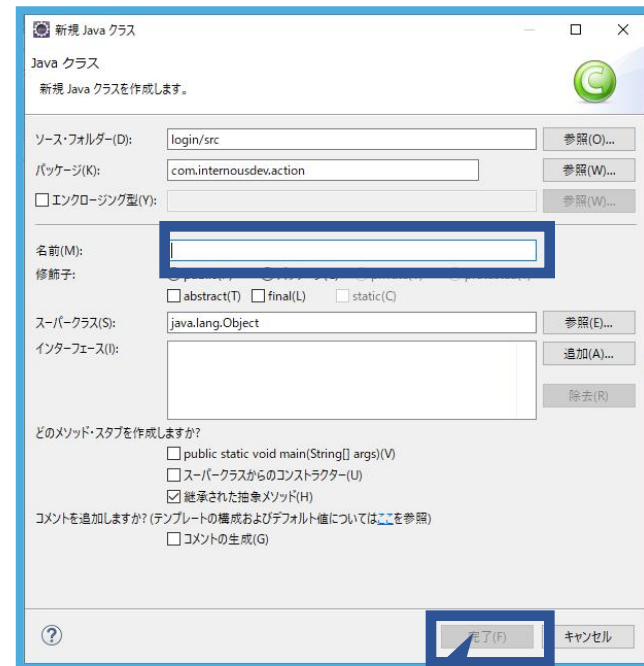
1. ログイン認証機能
- 2. ユーザー登録機能**
3. 商品購入機能
4. 購入履歴機能

UserCreateAction.javaの作成

1 UserCreateAction.java



① 「com.internousdev.template.action」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「UserCreateAction」を入力し、完了ボタンをクリックします。

UserCreateAction.javaの作成

③ 以下の内容を写経します。

UserCreateAction.java
(javaファイル)

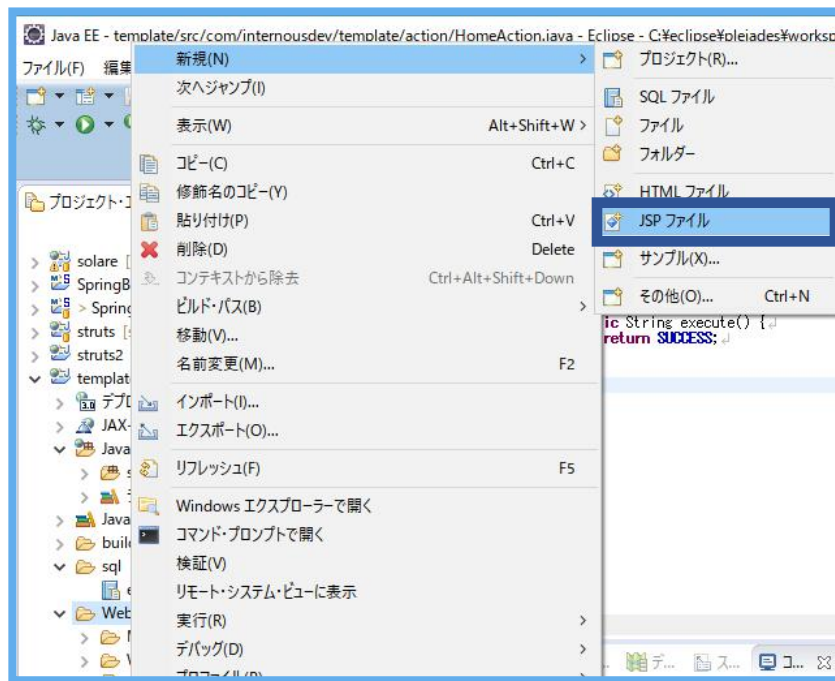
```
package com.internousdev.template.action;

import com.opensymphony.xwork2.ActionSupport;

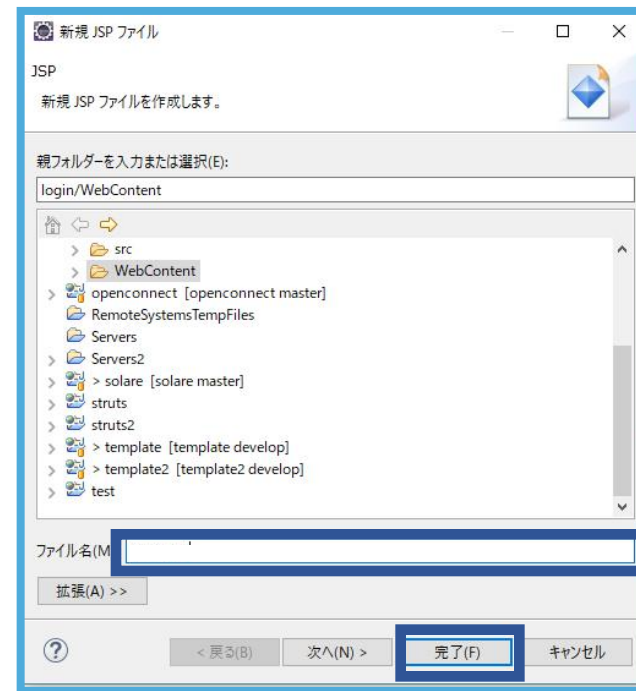
public class UserCreateAction extends ActionSupport {
    public String execute() {
        return SUCCESS;
    }
}
```

userCreate.jspの作成

2 userCreate.jsp



① 「プロジェクト」「WebContent」を選択し、「右クリック」「新規」「JSPファイル」を選択します。



② 「名前(M):」欄に「userCreate.jsp」を入力し、完了ボタンをクリックします。

userCreate.jspの作成

③ 以下の内容を写経します。

userCreate.jsp(jspファイル)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html>
<html>
<head>

<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="./css/style.css">

<title>UserCreate画面</title>

</head>
```

次へ続きます。

userCreate.jspの作成

前の続きです。

```
<body>
  <div id="header">
  </div>
  <div id="main">
    <div id="top">
      <p>UserCreate</p>
    </div>
    <div>
      <s:if test="errorMessage != " ">
        <s:property value="errorMessage" escape="false" />
      </s:if>
      <table>
        <s:form action="UserCreateConfirmAction">
```

半角のシングルクォーテーション2個

次へ続きます。

前の続きです。

```
<tr>
  <td>
    <label>ログインID:</label>
  </td>
  <td>
    <input type="text" name="loginUserId" value="" />
  </td>
</tr>
<tr>
  <td>
    <label>ログインPASS:</label>
  </td>
  <td>
    <input type="text" name="loginPassword" value="" />
  </td>
</tr>
```

次へ続きます。

userCreate.jspの作成

前の続きです。

```
<tr>
  <td>
    <label>ユーザー名:</label>
  </td>
  <td>
    <input type="text" name="userName" value="" />
  </td>
</tr>
<s:submit value="登録" />
</s:form>
</table>
```

次へ続きます。

userCreate.jspの作成

前の続きです。

```
                <div>
                    <span>前画面に戻る場合は</span>
                    <a href='<s:url action="HomeAction" />'>こちら</a>
                </div>
            </div>
        </div>
        <div id="footer">
        </div>
    </body>
</html>
```

userCreateAction機能の確認

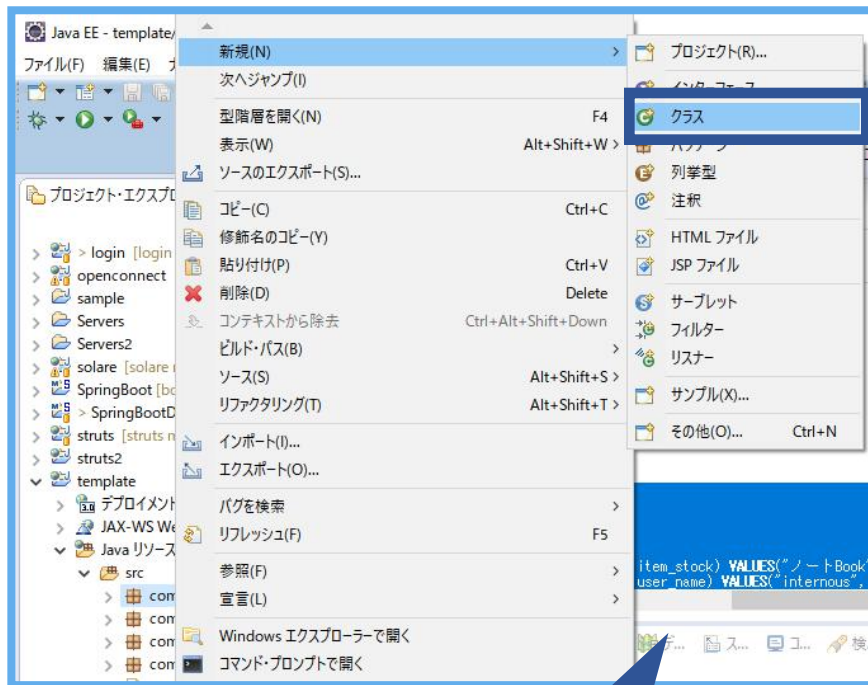
userCreate.jspの画面



templateプロジェクトを実行し、login.jsp画面のリンクから上記のuserCreate.jsp画面が表示できれば成功です。

UserCreateConfirmAction.javaの作成

3 UserCreateConfirmAction.java



① 「com.internousdev.template.action」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「UserCreateConfirmAction」を入力し、完了ボタンをクリックします。

UserCreateConfirmAction.javaの作成

③ 以下の内容を写経します。

UserCreateConfirmAction.java
(javaファイル)

```
package com.internousdev.template.action;
import java.util.Map;
import org.apache.struts2.interceptor.SessionAware;
import com.opensymphony.xwork2.ActionSupport;

public class UserCreateConfirmAction extends ActionSupport implements SessionAware {

    private String loginUserId;
    private String loginPassword;
    private String userName;
    private Map<String,Object> session;
    private String errorMessage;

    public String execute() {

        String result = SUCCESS;
```

次へ続きます。

UserCreateConfirmAction.javaの作成

前の続きです。

```
if(!loginUserId.equals(""))
    && !(loginPassword.equals(""))
    && !(userName.equals(""))) {
    session.put("loginUserId", loginUserId);
    session.put("loginPassword", loginPassword);
    session.put("userName", userName);
} else {
    setErrorMessage("未入力の項目があります。");
    result = ERROR;
}
return result;
}
```

未入力項目の有無を確認しています。

確認画面で表示したい値をセッションに格納します。

次へ続きます。

UserCreateConfirmAction.javaの作成

前の続きです。

```
public String getLoginUserId() {  
    return loginUserId;  
}  
  
public void setLoginUserId(String loginUserId) {  
    this.loginUserId = loginUserId;  
}  
  
public String getLoginPassword() {  
    return loginPassword;  
}  
  
public void setLoginPassword(String loginPassword) {  
    this.loginPassword = loginPassword;  
}
```

次へ続きます。

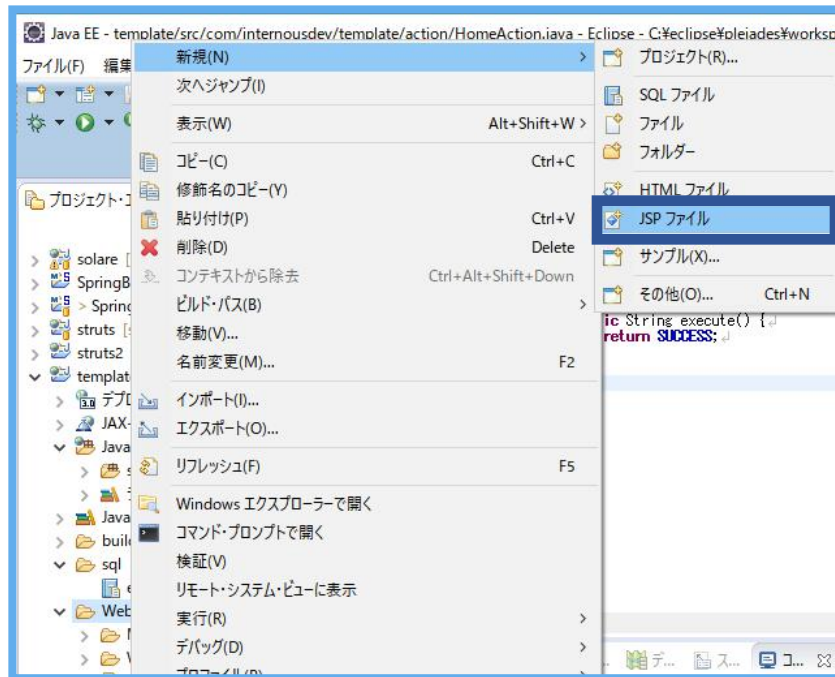
UserCreateConfirmAction.javaの作成

前の続きです。

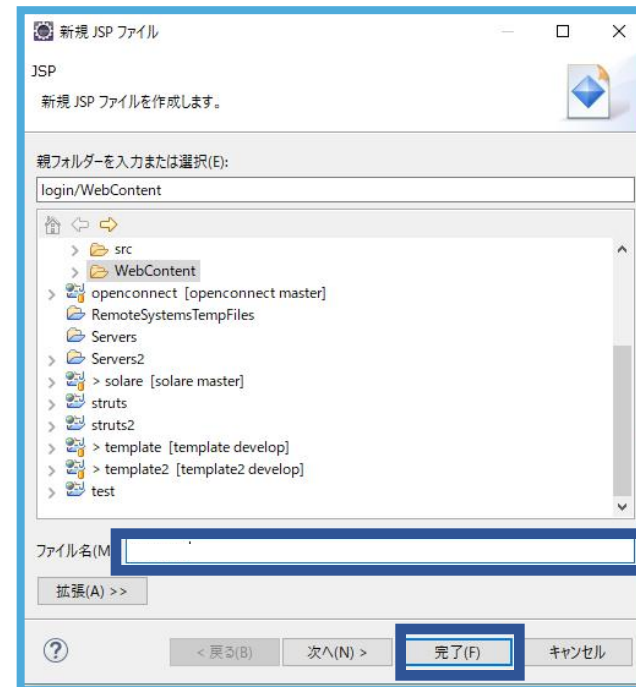
```
public String getUsername() {  
    return userName;  
}  
public void setUsername(String userName) {  
    this.userName = userName;  
}  
public Map<String, Object> getSession() {  
    return session;  
}  
@Override  
public void setSession(Map<String, Object> session) {  
    this.session = session;  
}  
public String getErrorMessage() {  
    return errorMessage;  
}  
public void setErrorMessage(String errorMessage) {  
    this.errorMessage = errorMessage;  
}  
}
```


userCreateConfirm.jspの作成

4 userCreateConfirm.jsp



① 「プロジェクト」「WebContent」を選択し、「右クリック」「新規」「JSPファイル」を選択します。



② 「名前(M):」欄に「userCreateConfirm.jsp」を入力し、完了ボタンをクリックします。

userCreateConfirm.jspの作成

③ 以下の内容を写経します。

UserCreateConfirm.jsp
(jspファイル)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html>
<html>
<head>

<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="./css/style.css">

<title>UserCreateConfirm画面</title>

</head>
```

次へ続きます。

前の続きです。

```
<body>
  <div id="header">
  </div>
  <div id="main">
    <div id="top">
      <p>UserCreateConfirm</p>
    </div>
    <div>
      <h3>登録する内容は以下でよろしいですか。</h3>
      <table>
        <s:form action="UserCreateCompleteAction">
```

次へ続きます。

前の続きです。

```
<tr id="box">
  <td>
    <label>ログインID:</label>
  </td>
  <td>
    <s:property value="loginUserId" escape="false" />
  </td>
</tr>
<tr id="box">
  <td>
    <label>ログインPASS:</label>
  </td>
  <td>
    <s:property value="loginPassword" escape="false" />
  </td>
</tr>
```

次へ続きます。

userCreateConfirm.jspの作成

前の続きです。

```
<tr id="box">
  <td>
    <label>ユーザー名:</label>
  </td>
  <td>
    <s:property value="userName" escape="false" />
  </td>
</tr>
<tr>
  <td>
    <s:submit value="完了" />
  </td>
</tr>
</s:form>
</table>
</div>
</div>
```

次へ続きます。


userCreateConfirm.jspの作成

前の続きです。

```
<div id="footer">  
</div>  
</body>  
</html>
```

userCreateConfirmAction機能の確認

userCreateConfirm.jspの画面



The screenshot displays a web page titled "UserCreateConfirm". It features a black header bar at the top. Below the header, a white box contains the title "UserCreateConfirm". The main content area has a heading "登録する内容は以下でよろしいですか。" (Is the registration information below correct?). Below this, the registration details are listed: "ログインID: taro", "ログインPASS: 123", and "ユーザー名: yamadataro". A "完了" (Complete) button is positioned to the right of the user name.

UserCreateConfirm

登録する内容は以下でよろしいですか。

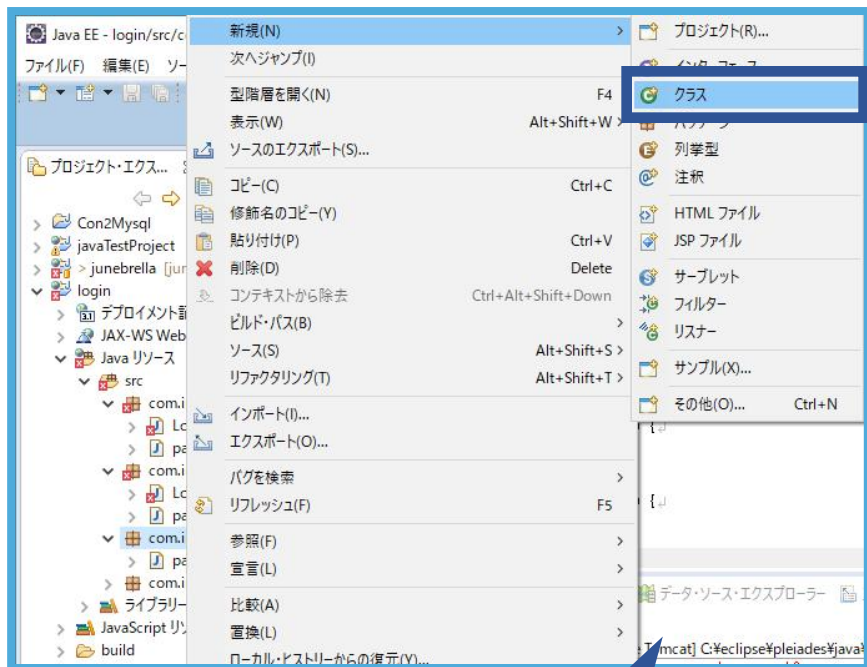
ログインID: taro
ログインPASS: 123
ユーザー名: yamadataro

完了

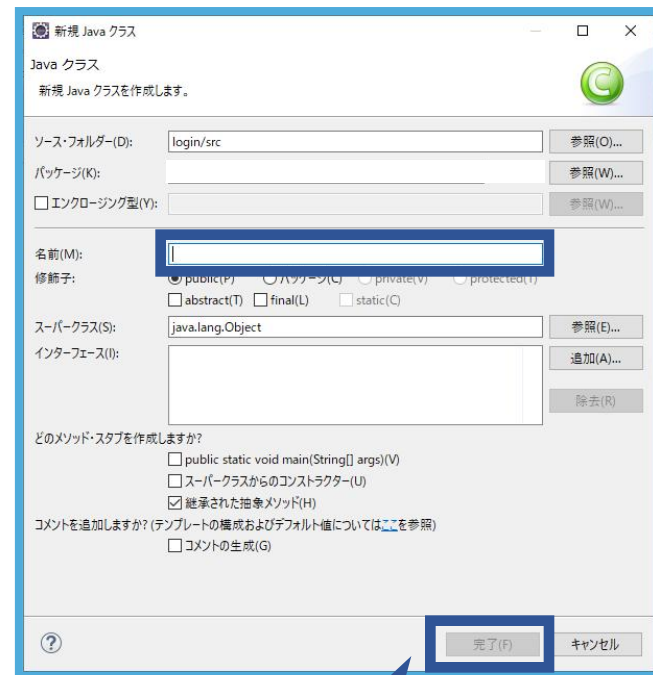
templateプロジェクトを実行し、userCreate画面の登録ボタンから上記のuserCreateConfirm.jsp画面が表示できれば成功です。

DateUtil.javaの作成

5 DateUtil.java



- ① 「com.internousdev.template.util」を選択し、「右クリック」「新規」「クラス」を選択します。



- ② 「名前(M):」欄に「DateUtil.java」を入力し、完了ボタンをクリックします。

DateUtil.javaの作成

DateUtil.java
(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.template.util;
```

```
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
```

```
public class DateUtil {
```

```
    public String getDate() {
```

```
        Date date = new Date();
```

```
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
```

```
        return simpleDateFormat.format(date);
```

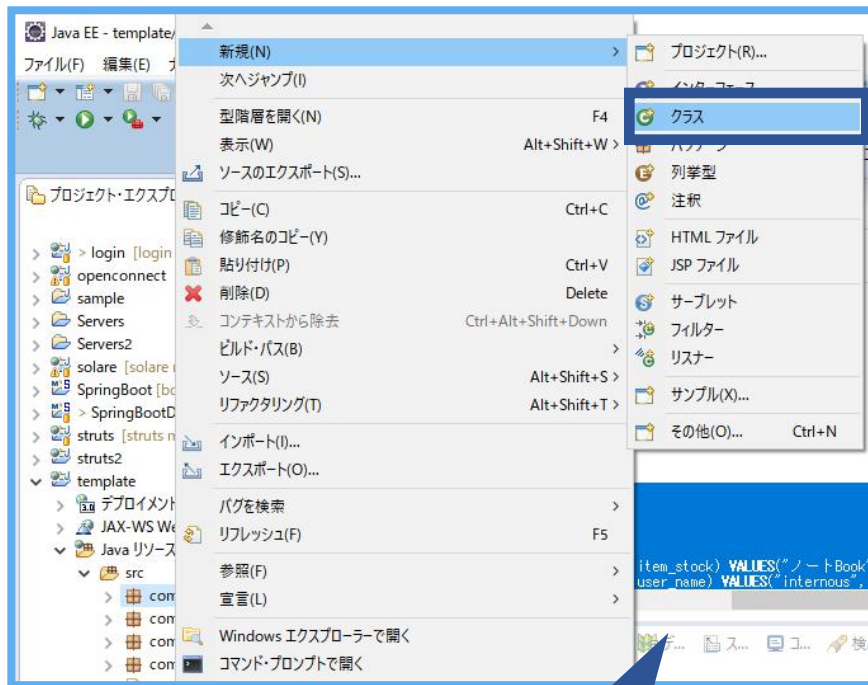
```
    }
```

```
}
```

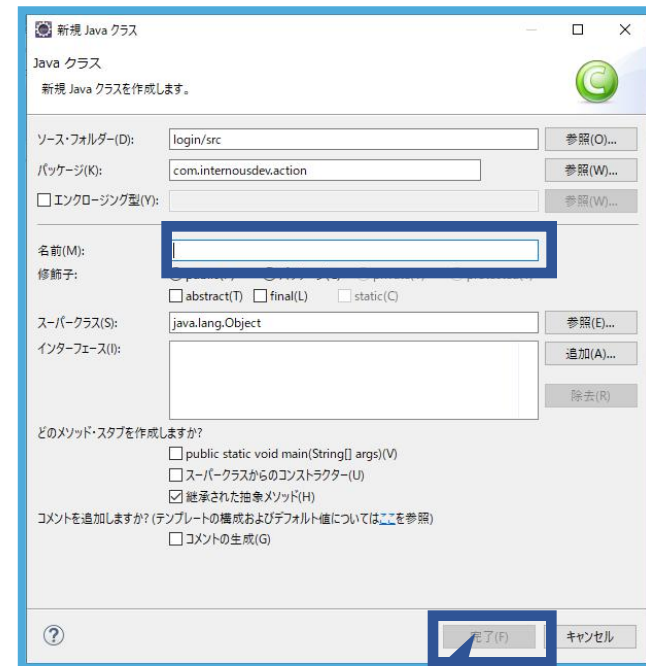
スペースを空ける。

UserCreateCompleteDAO.javaの作成

6 UserCreateCompleteDAO.java



① 「com.internousdev.template.dao」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「UserCreateCompleteDAO」を入力し、完了ボタンをクリックします。

UserCreateCompleteDAO.javaの作成

UserCreateCompleteDAO.java
(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.template.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import com.internousdev.template.util.DBConnector;
import com.internousdev.template.util.DateUtil;

public class UserCreateCompleteDAO{

    public void createUser(String loginUserId, String loginUserPassword, String userName) {

        DateUtil dateUtil = new DateUtil();
        DBConnector db = new DBConnector();
        Connection con = db.getConnection();

        String sql= "INSERT INTO login_user_transaction (login_id, login_pass, user_name,
insert_date) VALUES(?, ?, ?, ?)";
```

次へ続きます。

UserCreateCompleteDAO.javaの作成

前の続きです。

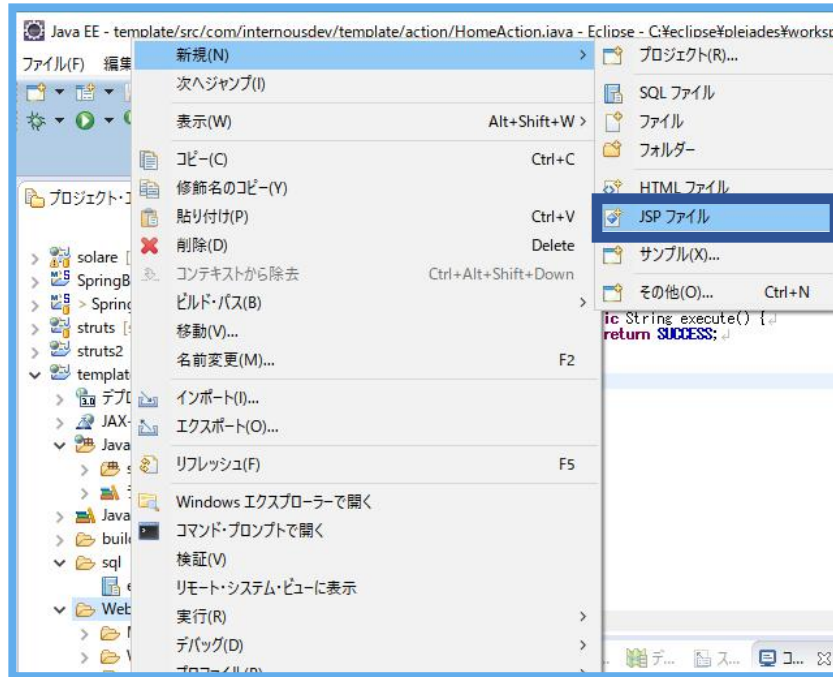
```
try {
    PreparedStatement ps = con.prepareStatement(sql);
    ps.setString(1, loginUserId);
    ps.setString(2, loginUserPassword);
    ps.setString(3, userName);
    ps.setString(4, dateUtil.getDate());

    ps.execute();

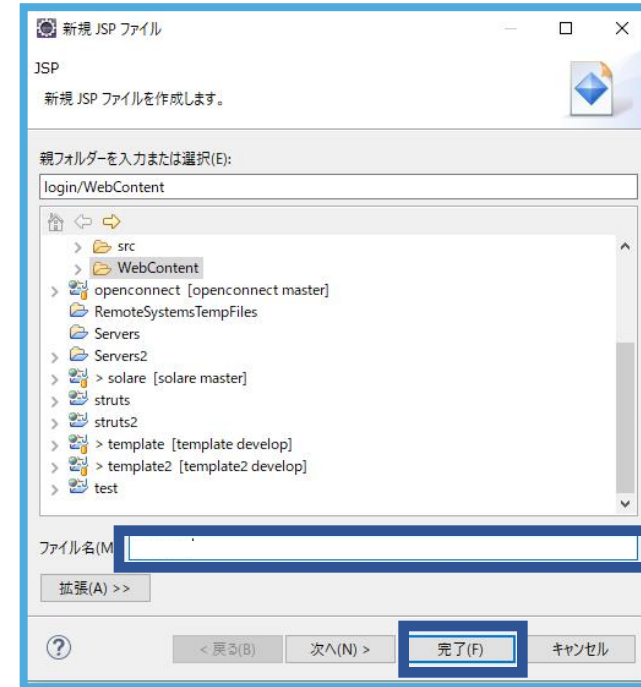
} catch (Exception e) {
    e.printStackTrace();
} finally {
    try {
        con.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

userCreateComplete.jspの作成

7 userCreateComplete.jsp



① 「プロジェクト」「WebContent」を選択し、「右クリック」「新規」「JSPファイル」を選択します。



② 「名前(M):」欄に「userCreateComplete.jsp」を入力し、完了ボタンをクリックします。

userCreateComplete.jspの作成

③ 以下の内容を写経します。

UserCreateComplete.jsp
(jspファイル)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html>
<html>
<head>

<meta charset="UTF-8">
<link rel="stylesheet" type="text/css" href="./css/style.css">

<title>UserCreateComplete画面</title>

</head>
```

次へ続きます。

userCreateComplete.jspの作成

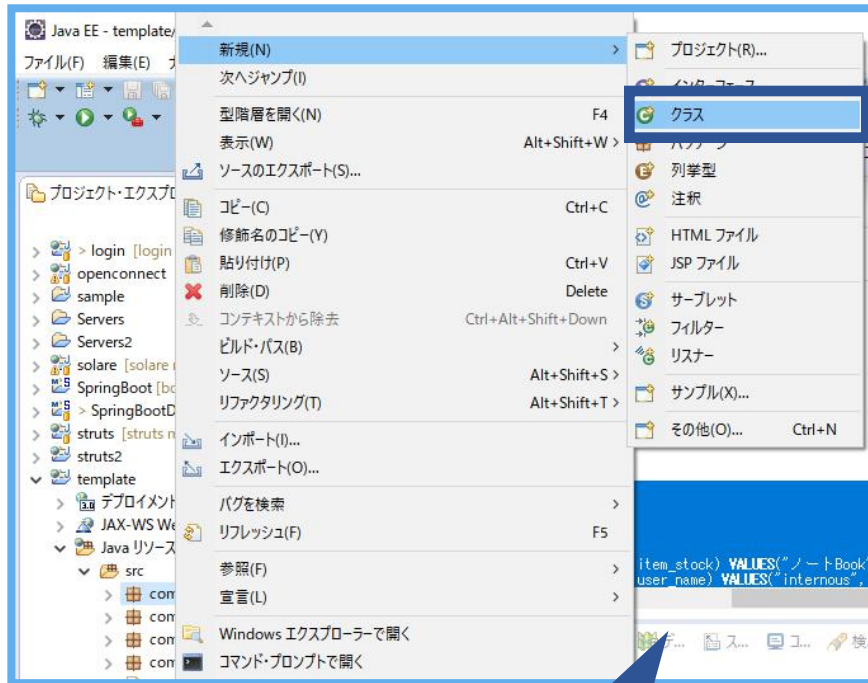
前の続きです。

```
<body>
  <div id="header">
  </div>
  <div id="main">
    <div id="top">
      <p>UserCreateComplete</p>
    </div>
    <div>
      <h3>ユーザーの登録が完了致しました。</h3>
      <div>
        <a href='<s:url action="HomeAction" />'>ログインへ</a>
      </div>
    </div>
  </div>
  <div id="footer">
  </div>
</body>
</html>
```

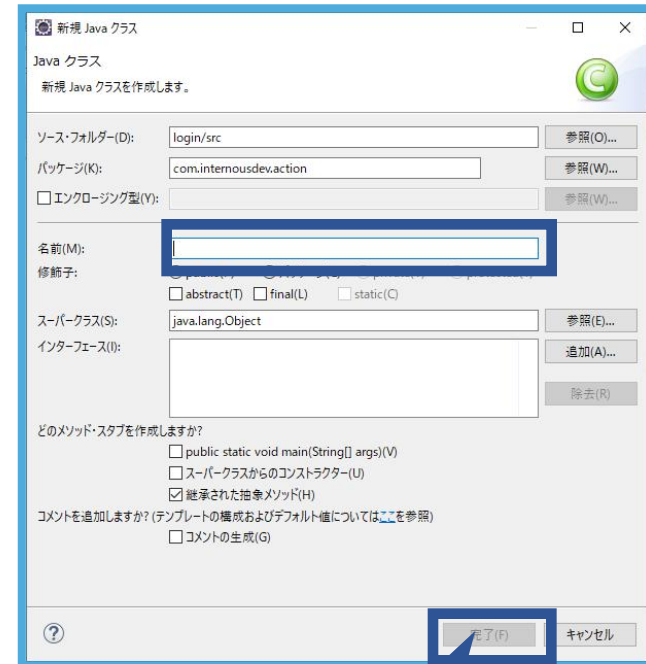
次へ続きます。

UserCreateCompleteAction.javaの作成

7 UserCreateCompleteAction.java



① 「com.internousdev.template.action」を選択し、「右クリック」「新規」「クラス」を選択します。



② 「名前(M):」欄に「UserCreateCompleteAction」を入力し、完了ボタンをクリックします。

UserCreateCompleteAction.javaの作成

UserCreateCompleteAction.java
(javaファイル)

③ 以下の内容を写経します。

```
package com.internousdev.template.action;
import java.util.Map;
import org.apache.struts2.interceptor.SessionAware;
import com.opensymphony.xwork2.ActionSupport;
import com.internousdev.template.dao.UserCreateCompleteDAO;

public class UserCreateCompleteAction extends ActionSupport implements SessionAware {

    private String loginUserId;
    private String loginPassword;
    private String userName;
    private Map<String,Object> session;

    public String execute(){
        UserCreateCompleteDAO userCreateCompleteDAO = new UserCreateCompleteDAO();
        userCreateCompleteDAO.createUser(session.get("loginUserId").toString(),
            session.get("loginPassword").toString(),
            session.get("userName").toString());
    }
}
```

DAOを経由して、
入力された内容をDBに登録します。

次へ続きます。

UserCreateCompleteAction.javaの作成

前の続きです。

```
String result = SUCCESS;

return result ;
}

public String getLoginUserId() {
    return loginUserId;
}

public void setLoginUserId(String loginUserId) {
    this.loginUserId = loginUserId;
}

public String getLoginPassword() {
    return loginPassword;
}

public void setLoginPassword(String loginPassword) {
    this.loginPassword = loginPassword;
}
```

次へ続きます。

UserCreateCompleteAction.javaの作成

前の続きです。

```
public String getUsername() {  
    return userName;  
}  
  
public void setUsername(String userName) {  
    this.userName = userName;  
}  
  
public Map<String, Object> getSession() {  
    return session;  
}  
  
@Override  
public void setSession(Map<String, Object> session) {  
    this.session = session;  
}  
}
```

userCreateComplete機能の確認

userCreateComplete.jspの画面



templateプロジェクトを実行し、userCreateConfirm画面の完了ボタンから上記のuserCreateComplete.jsp画面が表示できれば成功です。

userCreateComplete機能の確認

login_user_transactionテーブル

```
MariaDB [(none)]> use ecsite
Database changed
MariaDB [ecsite]> select * from login_user_transaction;
```

id	login_id	login_pass	user_name	insert_date	updated_date
1	internous	internous01	test	NULL	NULL
2	taro	123	taro123	2018-04-18 21:09:52	NULL

2 rows in set (0.00 sec)

コマンドプロンプト等を使用し、登録したユーザが実際にlogin_user_transactionテーブルに格納されているか確認します。