
ログイン認証

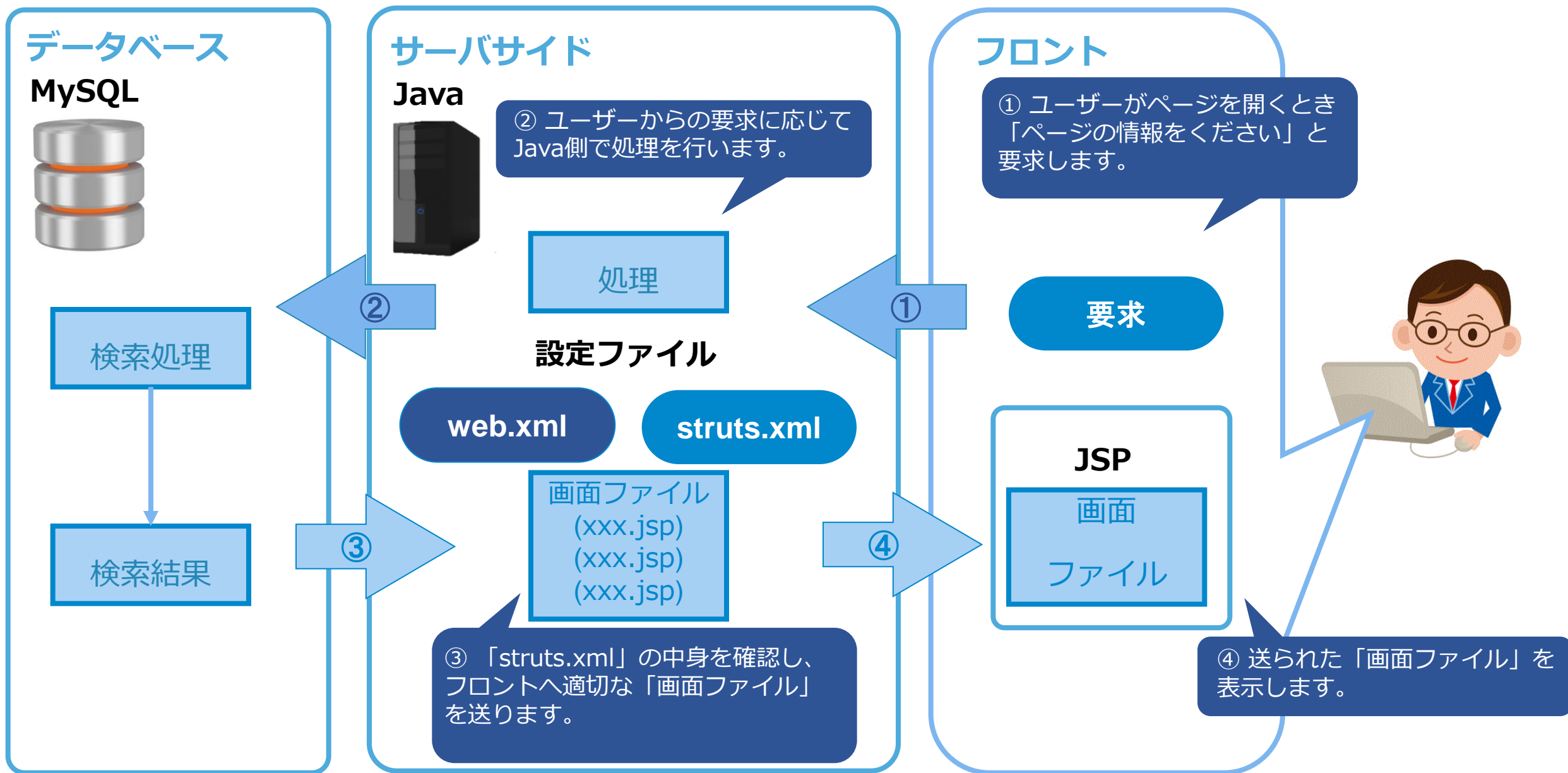
Struts設定ファイルの作成とDB作成の準備

2 時間目

設定ファイルとは

フレームワークの動作やプログラムの読み込み手順が記載されていて、内容を書き換えると「画面遷移順番の切り替え」「プログラム実行手順の切り替え」などの変更が行えます。しかし、設定ファイルを正確に記述しないと誤動作や動作不良を招くため扱いには注意が必要です。

設定ファイルとは



- 1) 設定ファイルとは
- 2) 設定ファイルの作成
 - 1 : web.xmlの作成
 - 2 : struts.xmlの作成
 - 3 : DBConnectorの作成

設定ファイルとは

struts.xml

「struts.xml」とは、
「strutsフレームワーク」を利用する際に作成する「設定ファイル」です。

javaの「処理結果」に応じて表示する画面ファイル(JSPファイル)の設定を記述します。

Strutsフレームワークのみで利用される「設定ファイル」です。

web.xml

「web.xml」とは、
動的Webプロジェクト作成時自動的に作られる「設定ファイル」です。

サイトへアクセスする際最初に表示する画面ファイル(JSPファイル)や、
プロジェクトでStruts2を使用する事を示す設定を記述します。

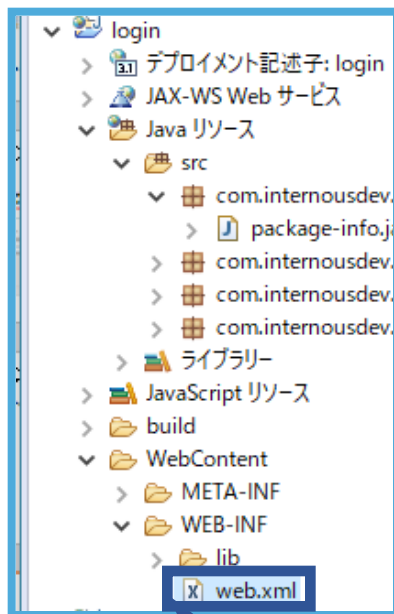
Strutsフレームワーク以外でも利用される「設定ファイル」です。

web.xmlの作成

解説

web.xmlでは、画面表示に関する設定やStrutsフレームワークの有効化設定が行えます。
web.xml<filter>タグにstrutsFilterの設定を行うとStrutsを有効にすることができます。

1 web.xml



① 「プロジェクト」「WebContent」「WEB-INF」の中にある「web.xml」ファイルを開きます。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3       xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4       xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
5       id="WebApp_ID" version="3.1">
6   <display-name>test</display-name>
7   <welcome-file-list>
8     <welcome-file>index.html</welcome-file>
9     <welcome-file>index.htm</welcome-file>
10    <welcome-file>index.jsp</welcome-file>
11    <welcome-file>default.html</welcome-file>
12    <welcome-file>default.htm</welcome-file>
13    <welcome-file>default.jsp</welcome-file>
14  </welcome-file-list>
15 </web-app> [EOF]
```

② このコードが初期状態で入っています。

③ 上記の内容を次のページの内容に変更します。

web.xmlの作成

web.xml(xmlファイル)

④ 以下の内容に書き換えます。

ここは決まり文句なので、
研修の段階では覚える必要はありません。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
```

```
<display-name>login</display-name>
```

プロジェクト名を記述します。「login」と書いてください。

```
<welcome-file-list>
  <welcome-file>login.jsp</welcome-file>
</welcome-file-list>
```

実行後、一番最初に表示されるJSPファイルを設定します。
初期画面にしたいJSPファイル名を書いてください。

前半では<welcome-file>で
「初期画面」（実行後最初に表示される画面）を
指定します。

次（後半）へ続きます。

web.xmlの作成の続き

前（前半）の続きです。

```
<filter>
  <filter-name>struts2</filter-name>
  <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>
```

① Struts2の制御担当クラス（固定）

```
<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

② 「適用する範囲（URL）」（固定）
「このプロジェクト配下（/）の
すべてのページ（*）」を表す

2か所の<filter-name>タグに
同じ内容を記述

```
</web-app>
```

後半<filter>～<filter-mapping>は
「このプロジェクト全体でstruts2を使います」
という意味になることを理解できればOKです！

web.xml : プロジェクトの設定ファイル

①前半

<welcome-file>最初に表示させたいJSPファイルを書く </welcome-file>

②後半

<filter>struts2の制御担当クラスを書く（毎回固定） </filter>

<filter-mapping>「適用する範囲（URL）」を書く（/*で固定） </filter-mapping>

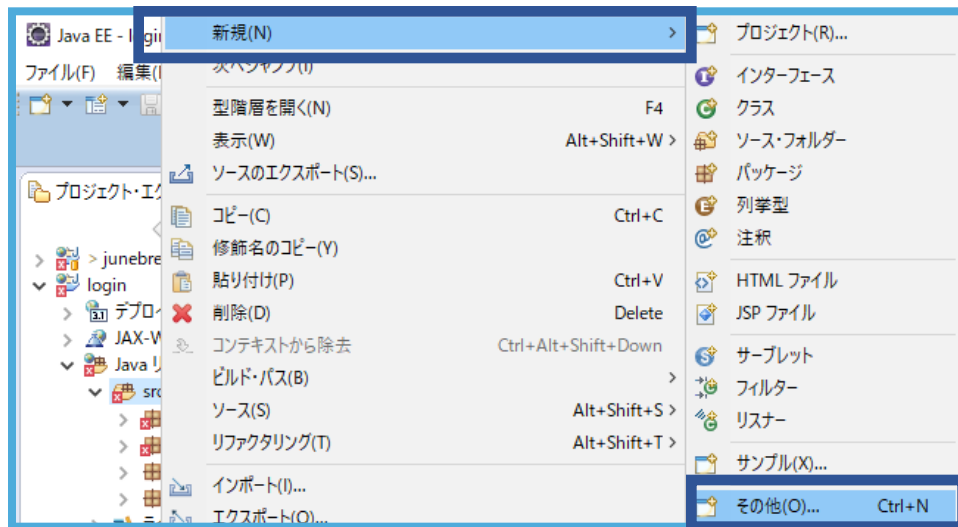
※あわせて、「このプロジェクトでstruts2を使う」と指定したことになります。

struts.xmlの作成

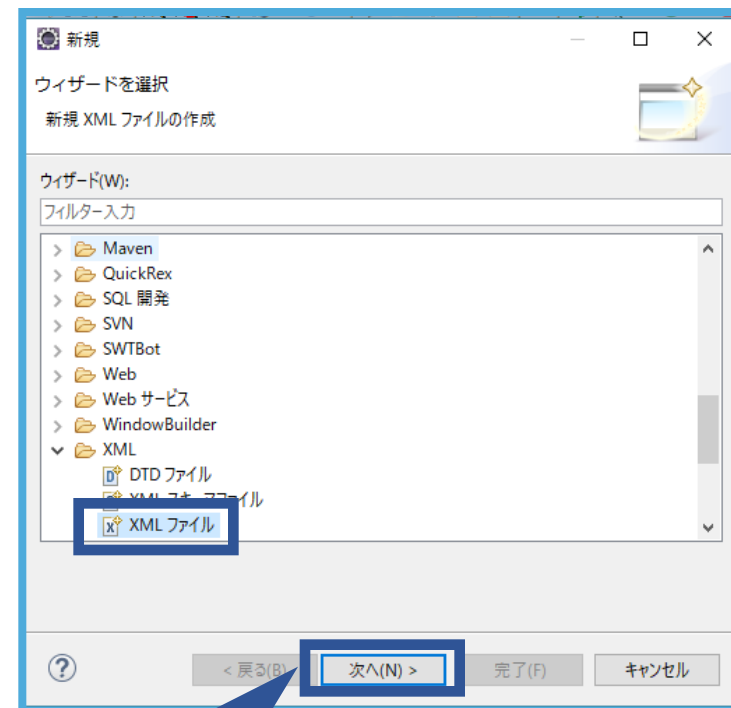
解説

struts.xmlでは、JSPとJavaファイルの連携を設定します。
「Java側の処理が終わった後次にどのJSPを呼び出すのか」等の設定ができます。

2 struts.xml

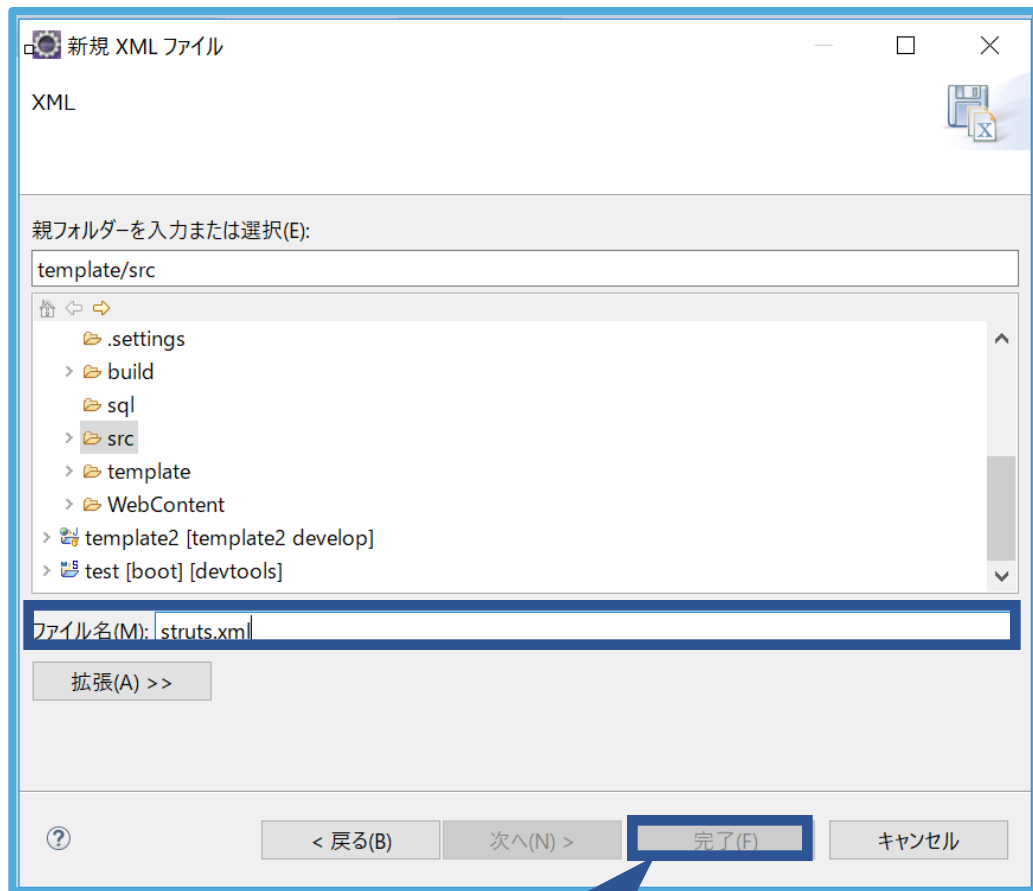


① 「Javaリソース」「src」を右クリックし
「新規」「その他」を選択します。

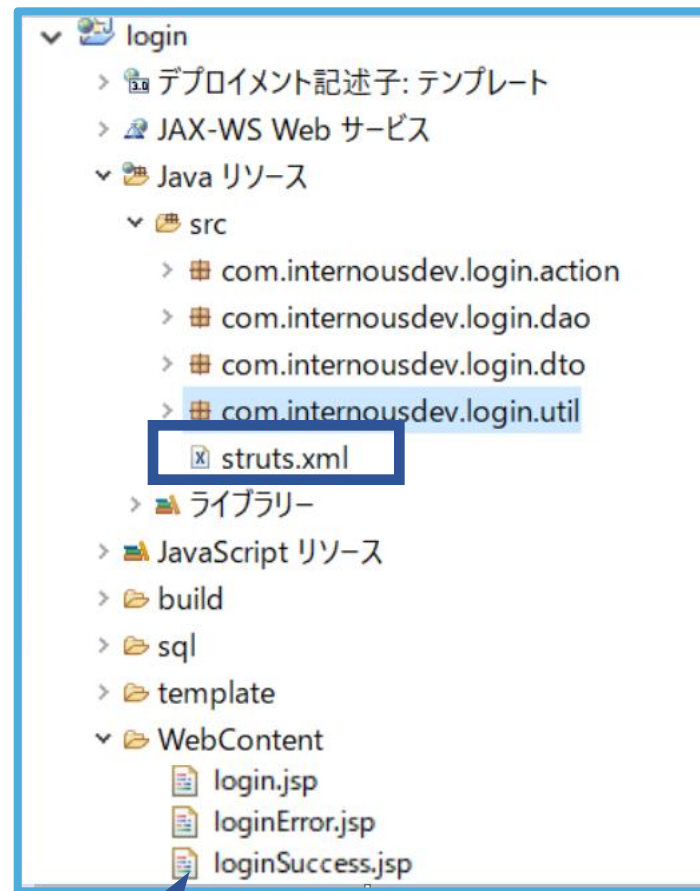


② 「ウィザード(W):」欄の「XML」「XMLファイル」
を選択し、「次へ」ボタンをクリックします。

struts.xmlの作成



③ ファイル名(M):の欄に「struts.xml」を入力して、完了をクリックします。



④ 「Javaリソース」「src」の直下に「struts.xml」が作成されていれば成功です。

struts.xmlの作成

⑤ 以下の内容を写経します。

struts.xml(xmlファイル)

決まり文句です。
スペースありなしに気を付けて写経しましょう。

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE struts PUBLIC
```

```
"-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
```

```
"http://struts.apache.org/dtds/struts-2.3.dtd">
```

```
<struts>
```

```
  <constant name="struts.devMode" value="true" />
```

```
  <package name="com.internousdev.login.action" extends="struts-default">
```

このプロジェクトで使うActionファイルが
保存されているパッケージを宣言します。

```
    <action name="LoginAction" class="com.internousdev.login.action.LoginAction" method="execute">
```

```
      <result name="success">loginSuccess.jsp</result>
```

```
      <result name="error">loginError.jsp</result>
```

```
    </action>
```

```
  </package>
```

execute()メソッドから返ってきた値によって、
次に遷移するJSPを振り分けます。

```
</struts>
```

呼び出されるJavaクラスファイルと実行
するメソッドを宣言します。

struts.xml : struts2の処理の流れを設定するファイル (プロジェクトの設計図)

<struts> </struts>の中身について

①<constant name=〇〇〇 value=〇〇〇>

Struts2の挙動を制御する設定です。name:設定名、value:設定値 をそれぞれ記述します。

※この研修では毎回同じ値を使います。

②<package name= 〇〇〇>

Actionファイルが保存されているパッケージ名を記述します。

③<action name= 〇〇〇>

プロジェクト内で使うActionファイルを記述します。

struts.xmlのまとめ（続き）

④<class= ○○○>

パッケージ名、Actionクラス名の順で記述します。

※③で呼び出されたActionクラスの場所を記述します。

⑤<method= ○○○>

普通はexecuteメソッドを記述します。

※③で呼び出されたActionクラスのどのメソッドが呼ばれるかを記述します。

⑥<result name="success">○○○.jsp

⑤のメソッドからSUCCESSが返された場合、ここで記述したJSPファイルに遷移します。

⑦<result name="error">○○○.jsp

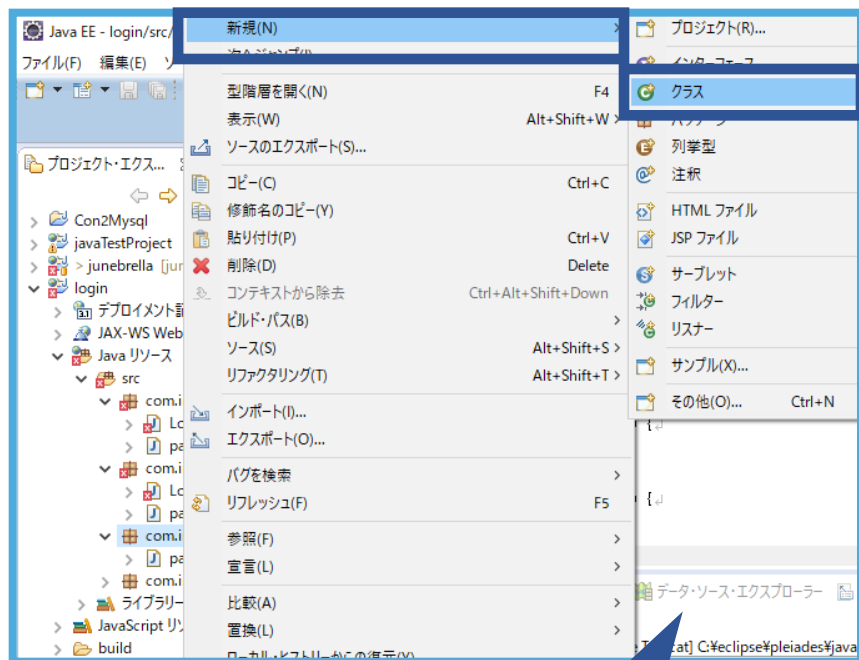
⑤のメソッドからERRORが返された場合、ここで記述したJSPファイルに遷移します。

DBConnectorの作成

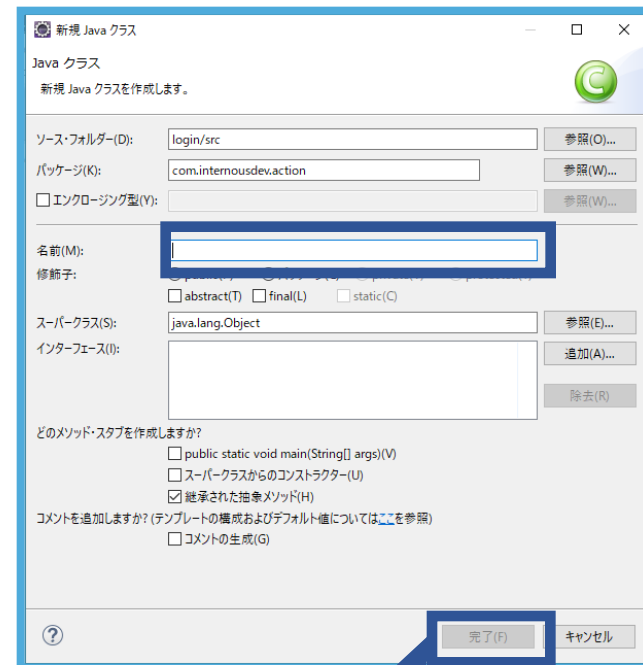
解説

データベースを使う必要がある場合、DB接続 (コネクション)の設定を行います。
DBConnectorでは、接続するDBの「場所」「名前」や接続する「ユーザ名」「パスワード」の設定を行います。

3 DBConnector



① 「src」「com.internousdev.login.util」を右クリックし、「新規」「クラス」を選択します。



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

DBConnectorの作成

③ 以下の内容を写経します。
DB接続の単元の解説を参考にしてください。

DBConnector.java(javaファイル)

```
package com.internousdev.login.util;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;
```

```
public class DBConnector {  
    private static String driverName = "com.mysql.jdbc.Driver";  
    private static String url = "jdbc:mysql://localhost/logindb_yamada";  
  
    private static String user = "root";  
    private static String password = "mysql";
```

```
    public Connection getConnection() {  
        Connection con = null;
```

```
        try {
```

```
            Class.forName(driverName);
```

```
            con = (Connection) DriverManager.getConnection(url,user,password);
```

MySQL接続に必要な情報を設定します。

接続先のデータベース名を指定します。
※「yamada」の部分は自分の名前に変更してください

設定した情報を使って自分のパソコンに
インストールされているMySQLサーバへ
接続するための記述です。

次へ続きます。

DBConnectorの作成

前の続きです。

```
    } catch(ClassNotFoundException e) {  
        e.printStackTrace();  
    } catch(SQLException e) {  
        e.printStackTrace();  
    }  
    return con;  
}
```

MySQLサーバに接続した結果を
メソッドの呼び出し元に渡します。

MySQL JDBC ドライバーの配備

解説

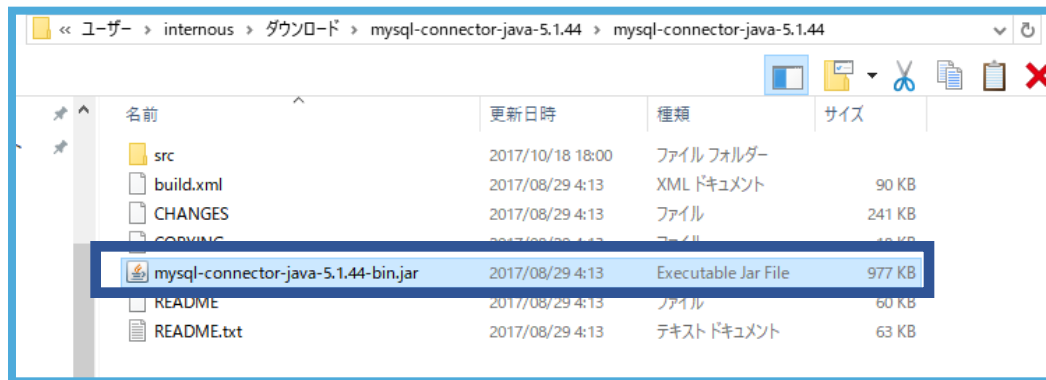
JavaファイルがMySQLデータベースに接続を行う際に利用するドライバークラスです。

4 MySQL JDBC ドライバー

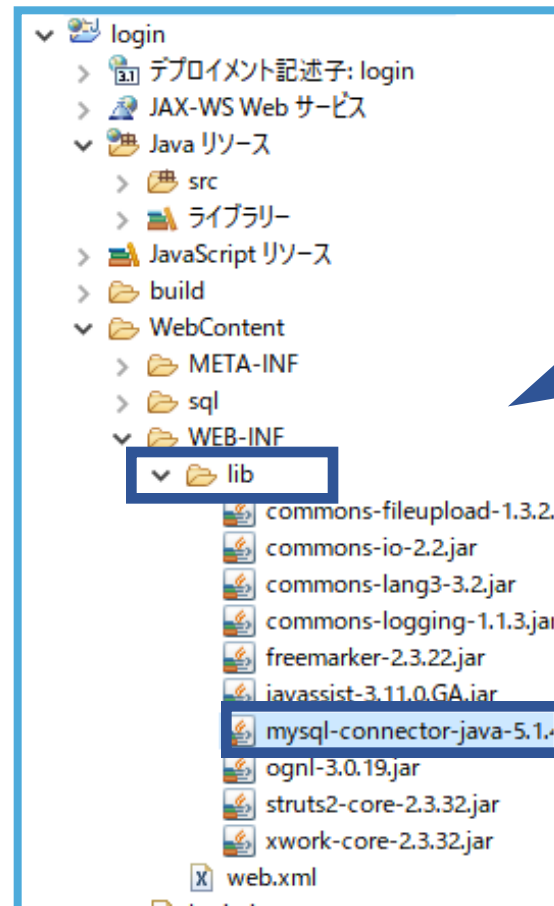
具体的なダウンロード方法は、
マイドライブ「最新教材」「9-java」フォルダの資料

MySQL JDBCドライバー(ConnectorJ)ダウンロード方法
を参照して下さい。

MySQL JDBC ドライバーの配備



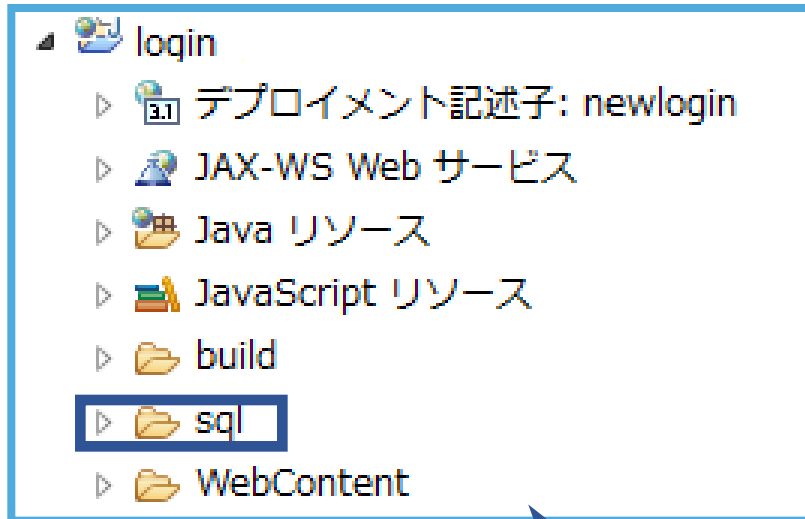
① ダウンロードした「mysql-connector-java-x.x.xx-bin.jar」ファイルをコピーします。



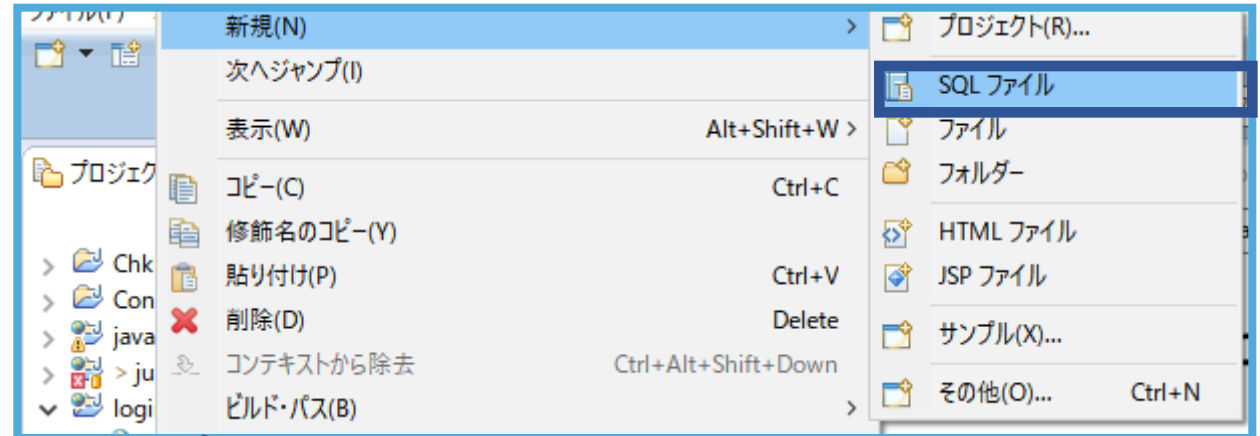
② eclipse上で、コピーした「mysql-connector-java-x.x.xx-bin.jar」を「login」「WebContent」「WEB-INF」「lib」フォルダ直下に貼り付けます。

これでドライバー配備は完了です。

SQLファイルの配備

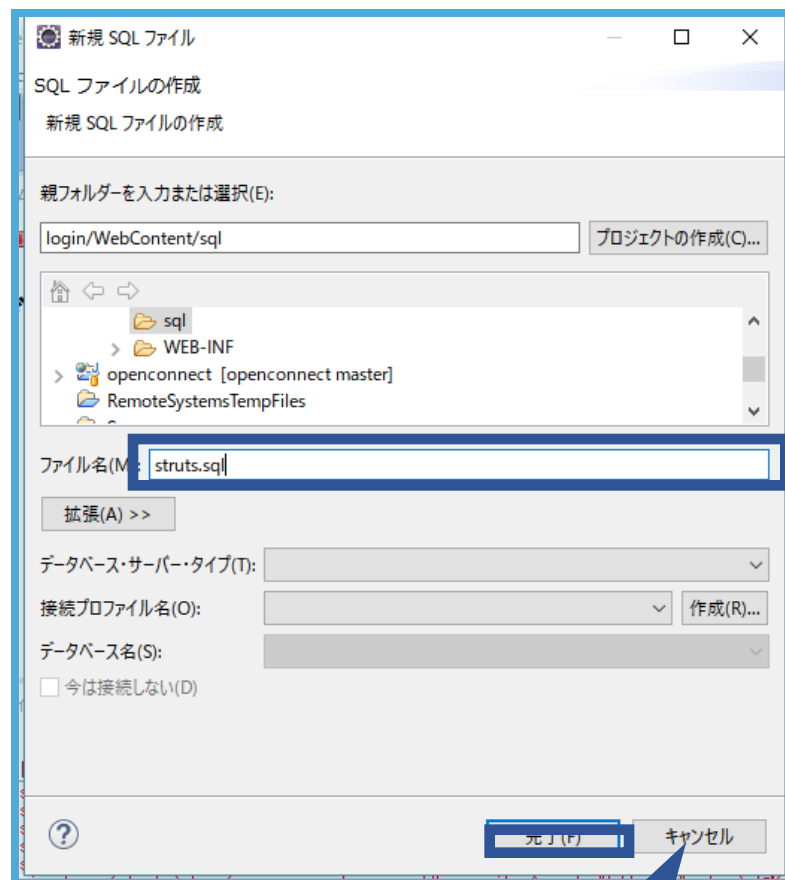


① 「プロジェクト」の直下に「sql」フォルダを作成します。

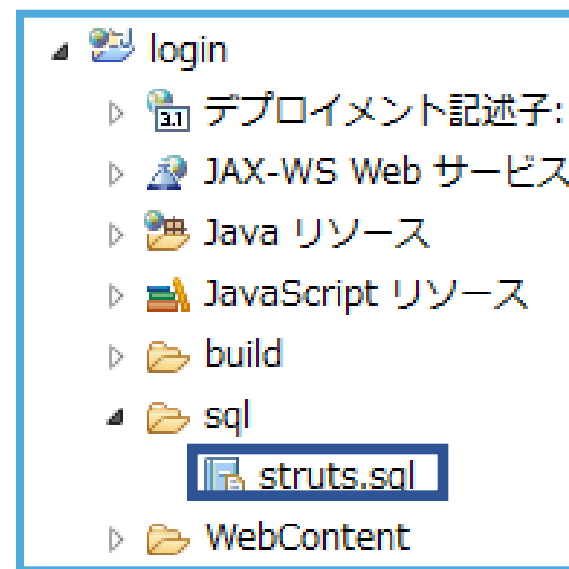


② 「sql」の上で右クリックし、「新規」「SQLファイル」を選択します。

SQLファイルの配備



③ 「ファイル名(M):」の欄に「struts.sql」と入力し、完了をクリックします。



④ SQLファイルが作成できていれば成功です。

SQLファイルの写経

①以下の内容を写経します。

struts.sql(sqlファイル)

```
set names utf8;  
set foreign_key_checks = 0;  
drop database if exists logindb_yamada;  
create database logindb_yamada;  
use logindb_yamada;
```

DBConnectorクラスで設定したデータベース名と
合わせます (logindb_自分の名前)。

```
create table user(  
id int,  
user_name varchar(255),  
password varchar(255)  
);
```

「user」がテーブル名になります。

```
insert into user values  
(1,"taro","123"),  
(2,"jiro","123"),  
(3,"hanako","123"),  
(4,"saburo","123");
```

「user」テーブルにデータを追加します。

データベースの作り方の説明

drop、create、useは、
定型文なので必ずこの3行は書きましょう。

drop database if exists **logindb_自分の名前;**

もしあれば、という条件付きで、
一度データベースを削除します。

create database **logindb_自分の名前;**

新たにデータベースを作成します。
必ず、1行目で削除したデータベース名と合わせます。

use **logindb_自分の名前;**

作成したデータベースに移動します。
ここも上記と同じデータベース名にします。