
ログイン認証

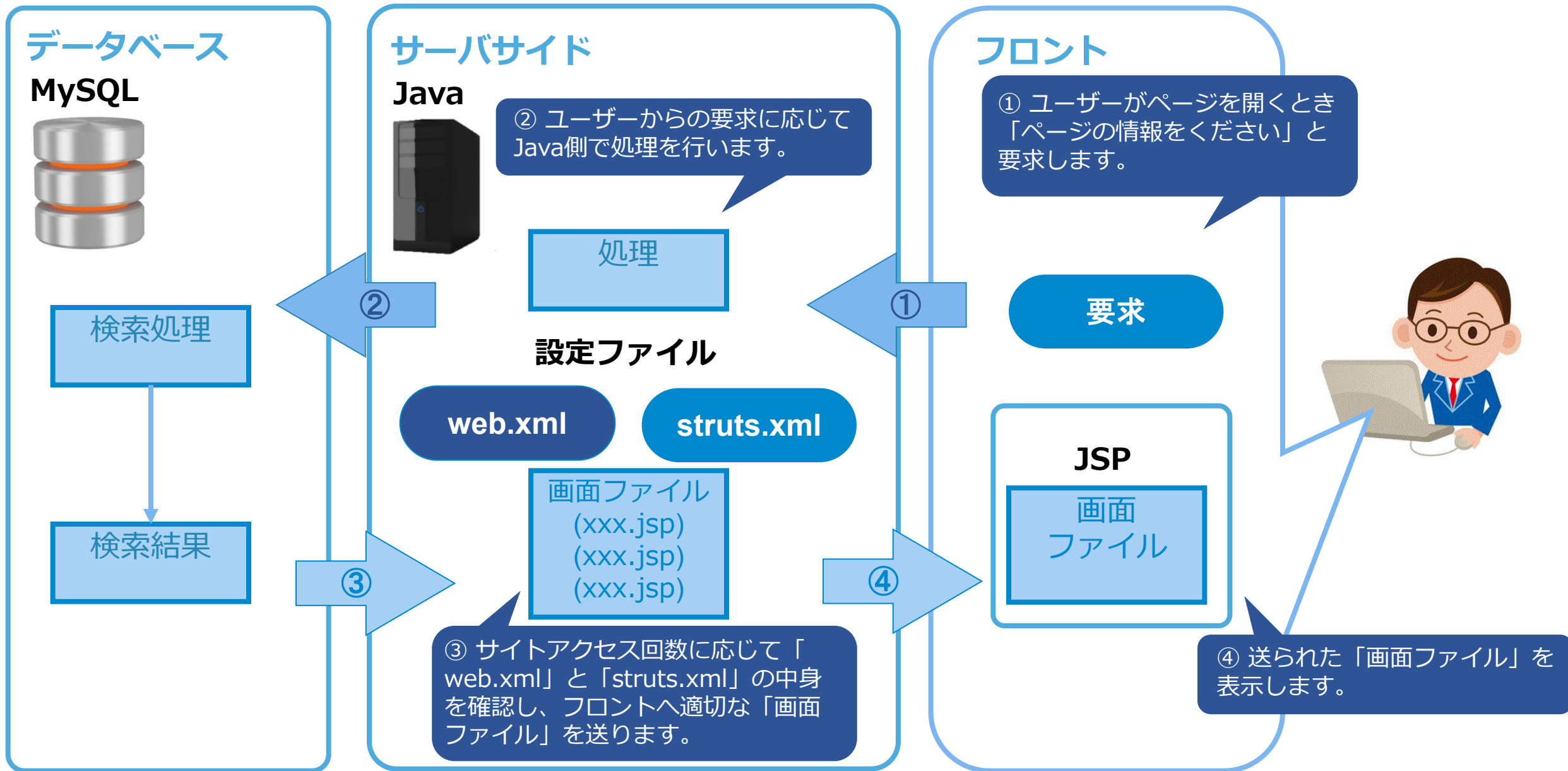
Struts設定ファイルの作成

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。

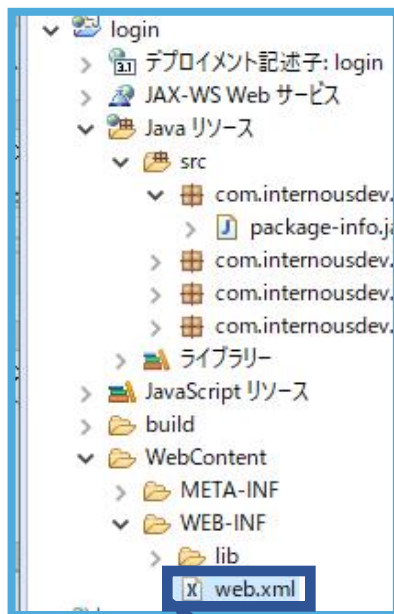
Strutsフレームワーク以外でも利用される「設定ファイル」です。

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>
```

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

④ 以下の内容に書き換えます。

web.xml(xmlファイル)

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

プロジェクト名を設定します。
「login」と書いてください。

最初に画面に表示する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。

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

Struts2を使うので、以下の設定を書いてください。

次へ続きます。

前の続きです。

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

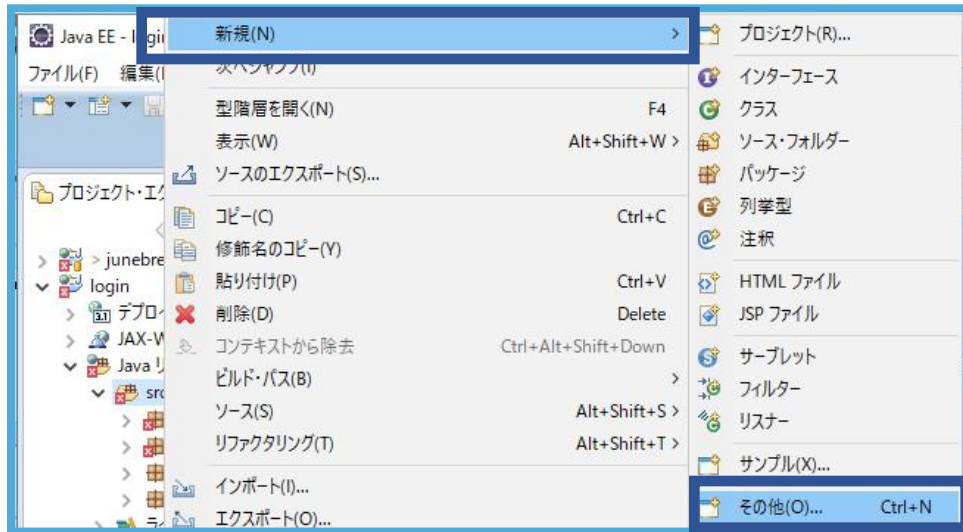

struts.xmlの作成

解説

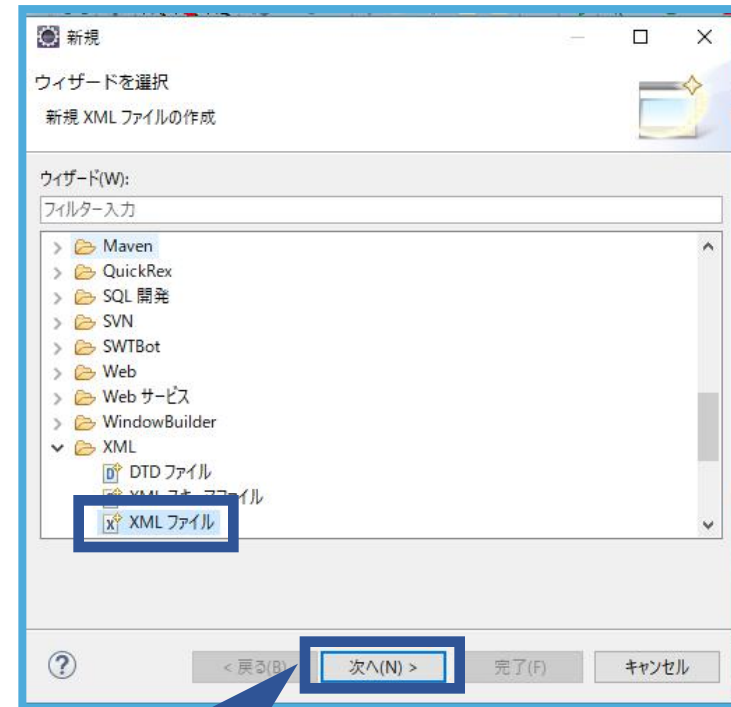
struts.xmlでは、JSPとJavaファイルの連携を設定します。

「JSP側が次にどのJavaファイルを呼び出すのか」「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

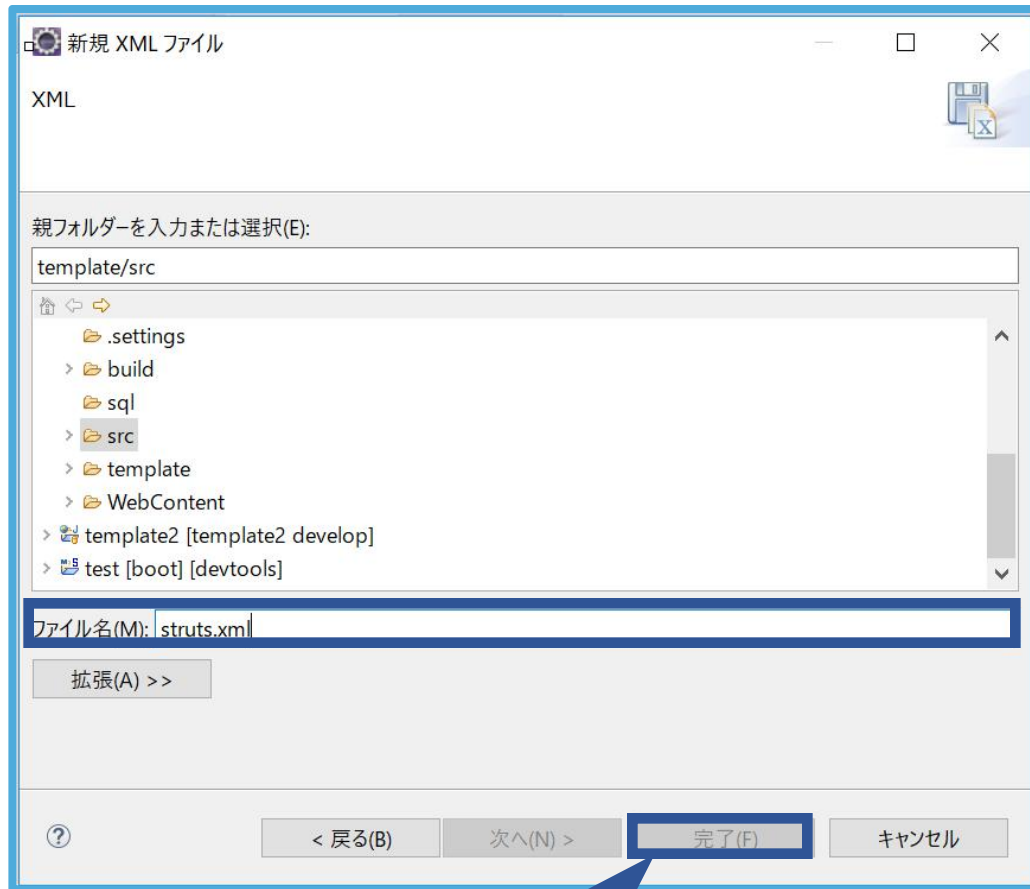


- ① 「Javaリソース」「src」を右クリックし「新規」「その他」を選択します。

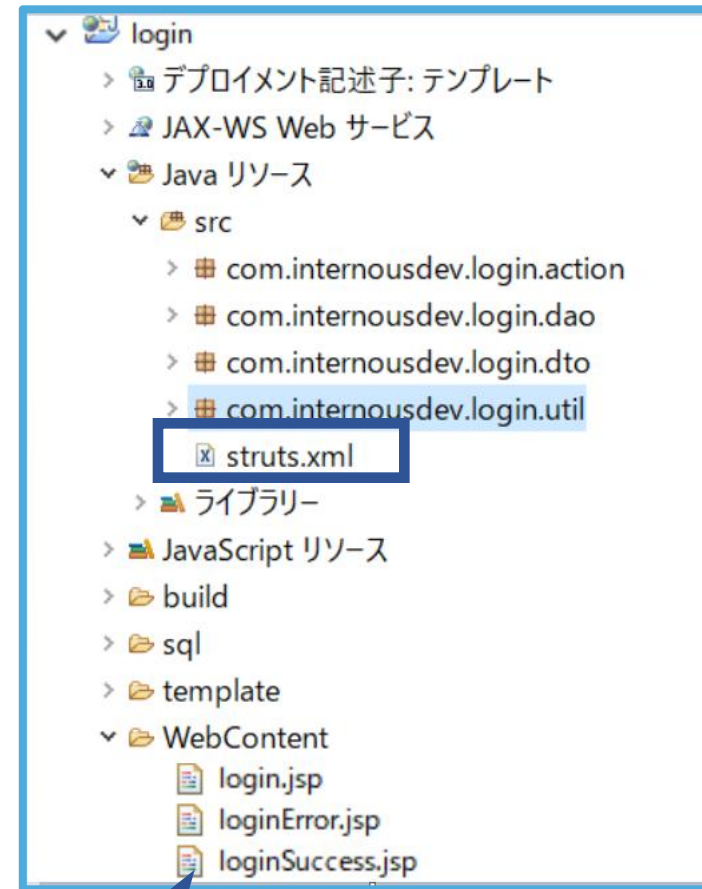


- ② 「ウィザード(W):」欄の「XML」「XMLファイル」を選択し、「次へ」ボタンをクリックします。

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)

```
<?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">

    <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>
</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登録します。

struts.xmlの作成

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

「LoginAction」クラスの中で「execute」メソッドが最初に呼び出されます。

「com.internousdev.login.action」パッケージの「LoginAction」に設定します。

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

「execute」メソッドの処理結果に応じてフロントに送る「JSPファイル」を設定しています。

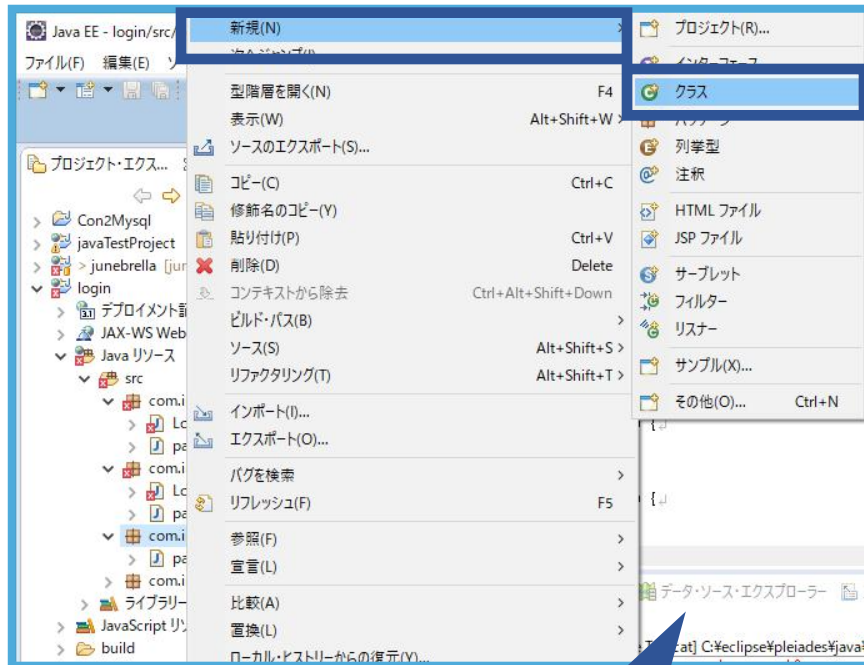
```
</action>
```

DBConnectorの作成

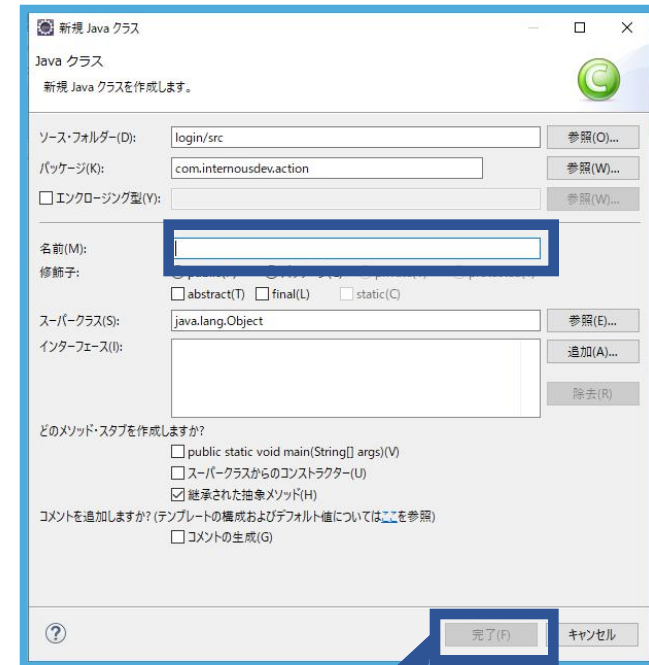
解説

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。

DBConnector.java(javaファイル)

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

MySQL接続に必要な情報を設定します。

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

```
    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);
```

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。

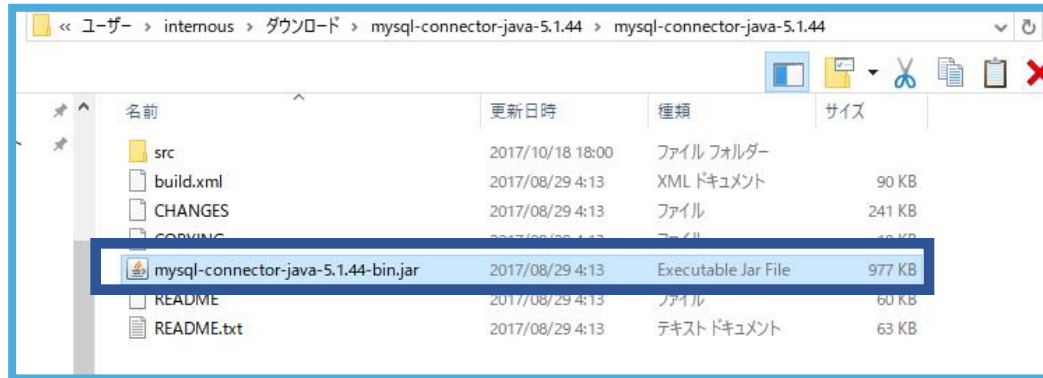
4 MySQL JDBC ドライバー

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

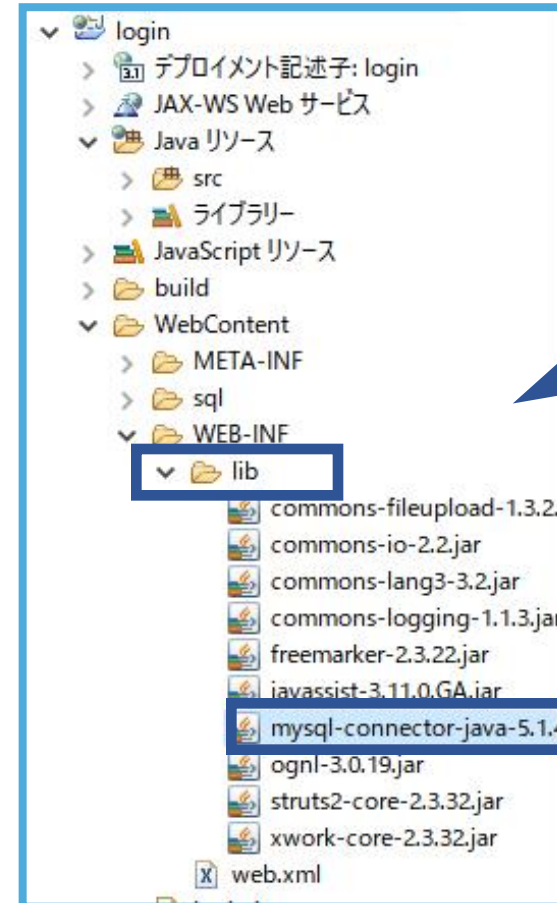
MySQL JDBCドライバー(ConnectorJ)ダウンロード方法

を参照して下さい。

MySQL JDBC ドライバーの配備



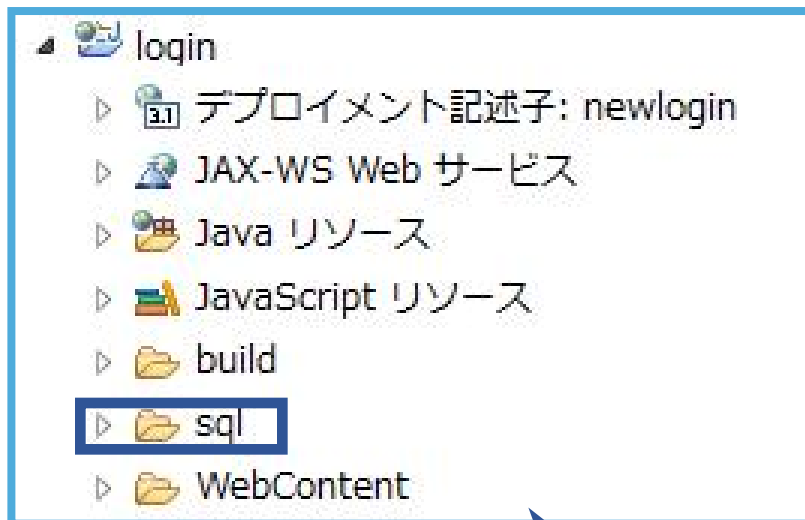
① ダウンロードした「mysql-connector-java-x.x.xx-bin.jar」ファイルをコピーします。



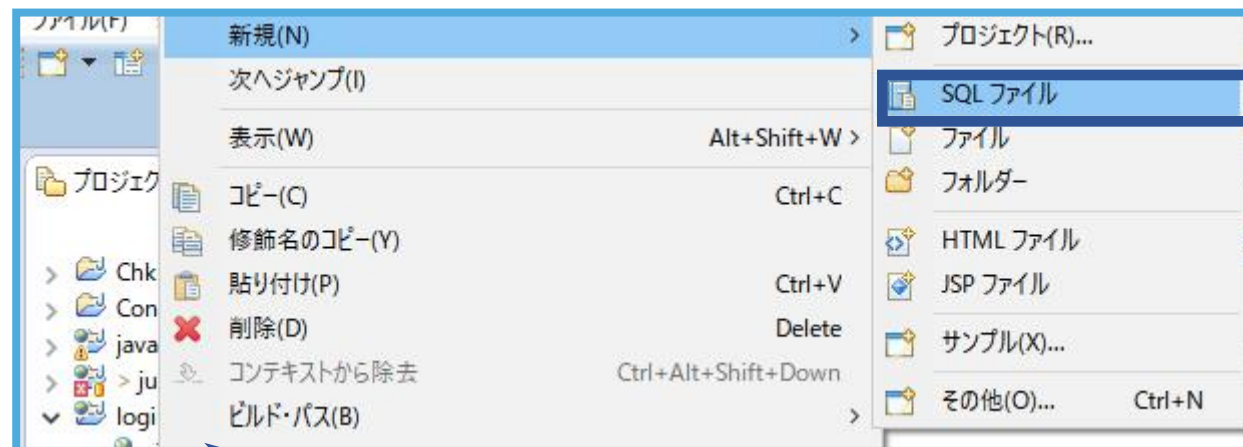
② eclipse上で、コピーした「mysql-connector-java-x.x.xx-bin.jar」を「login」「WebContent」「WEB-INF」「lib」フォルダ直下に貼り付けます。

これでドライバー配備は完了です。

SQLファイルの配備

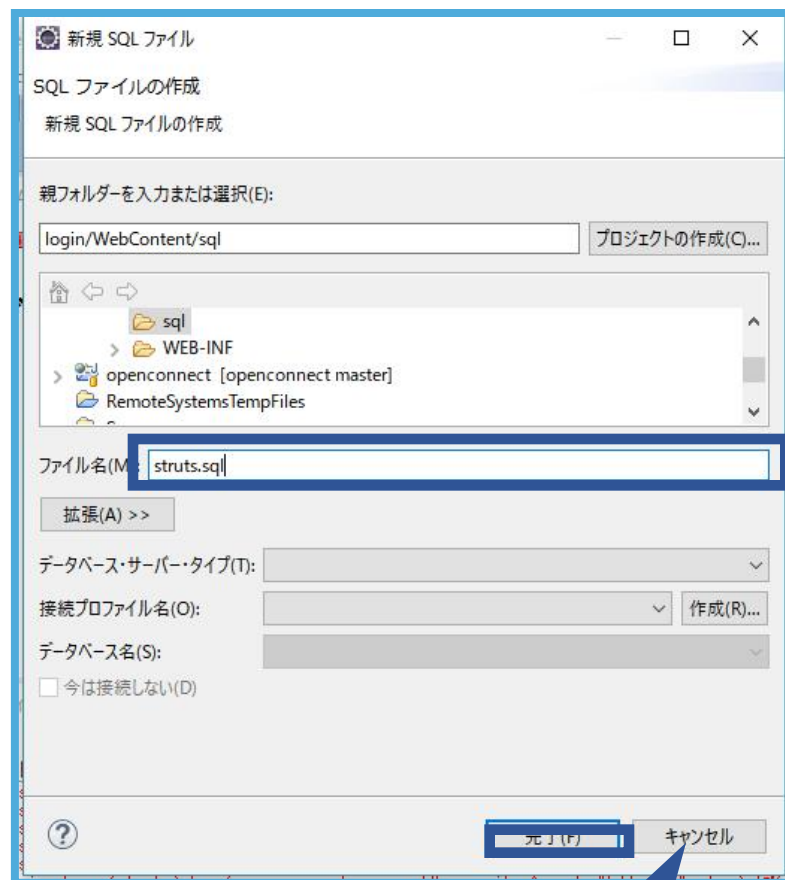


① 「プロジェクト」の直下に「sql」フォルダを作成します。

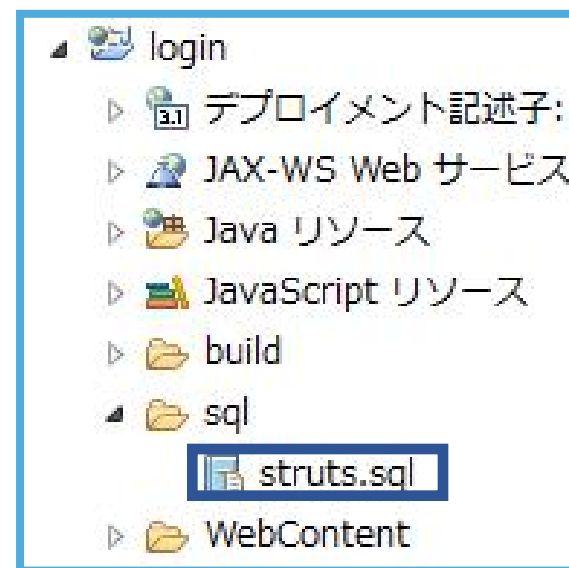


② 「sql」の上で右クリックし、「新規」「SQLファイル」を選択します。

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;  
create database logindb;  
use logindb;
```

「logindb」がデータベース名になります。

「user」がテーブル名になります。

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

```
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保存します。

drop database if exists logindb;

もしすでに「logindb」というデータベースがあったら、削除します。

create database logindb;

新たに「logindb」データベースを作成します。

use logindb;

作成した「logindb」データベースを利用します。