

---

# ECサイトログイン認証

Struts設定ファイルの作成

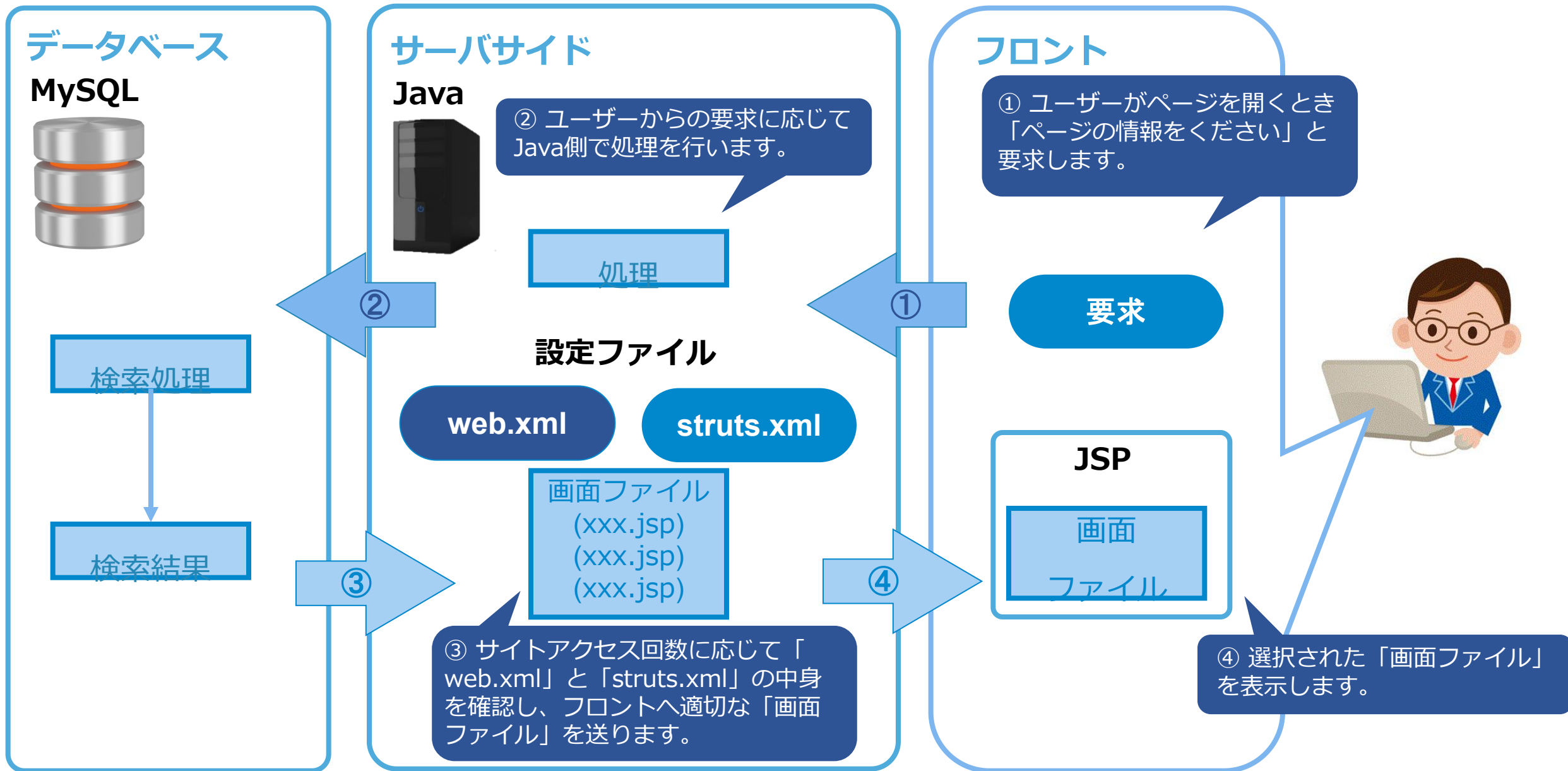
---

## 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」などの変更が行える。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。

サイトへアクセスする際「2回目以降のアクセス」時に「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)」の設定が記述されています。

「初回アクセス」の際は「web.xml」という「設定ファイル」が呼び出されます。

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「初回アクセス」時に「最初に表示する画面ファイル(JSPファイル)」の設定が記述されています。

「2回目以降のアクセス」の際は「struts.xml」という「設定ファイル」が呼び出されます。

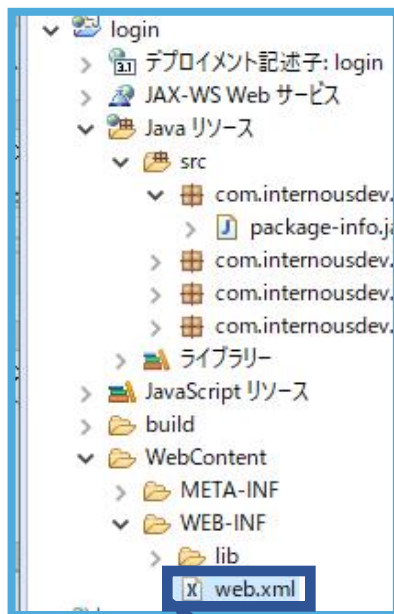
Strutsフレームワーク以外でも利用される「設定ファイル」です。

# 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。  
初回アクセスで表示させたい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>
```

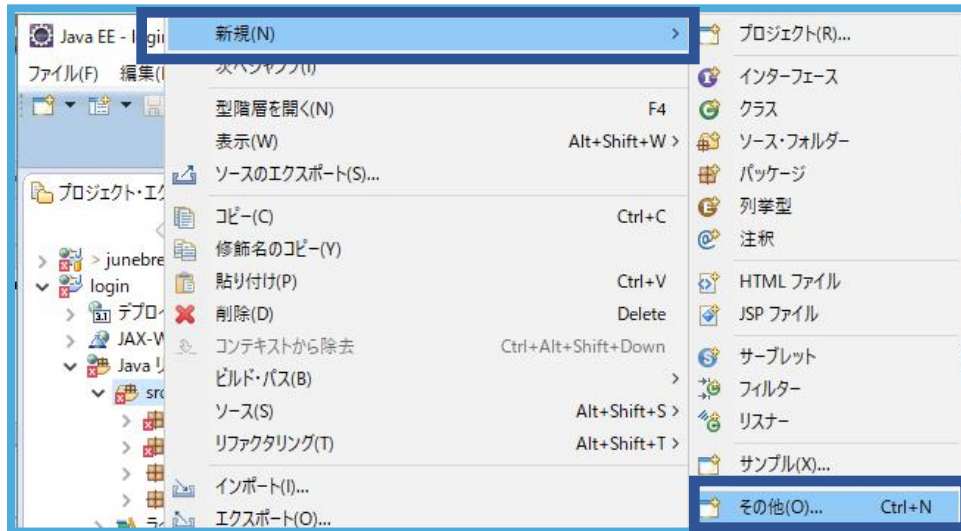


# struts.xmlの作成

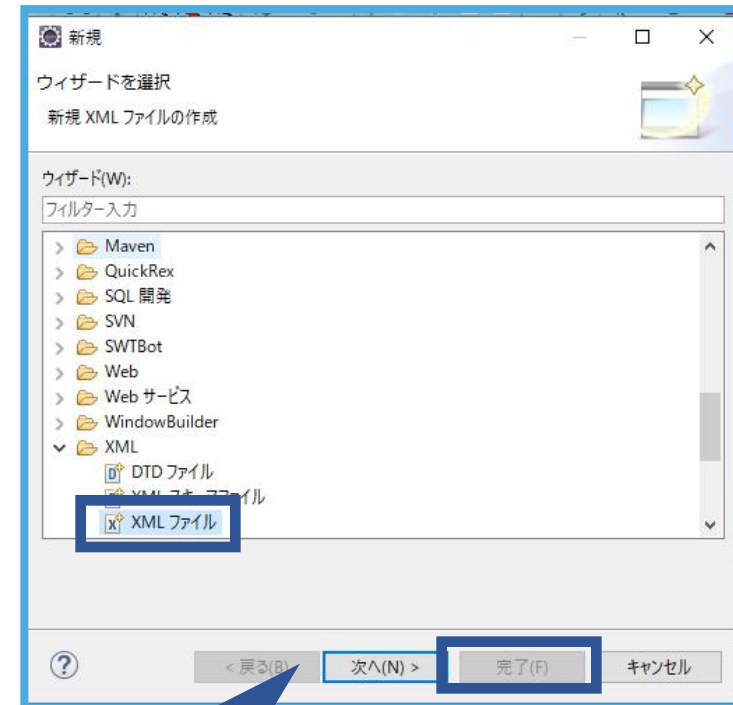
解説

struts.xmlでは、JSPとJavaファイルの連携を設定を行えます。  
JSP側から、「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

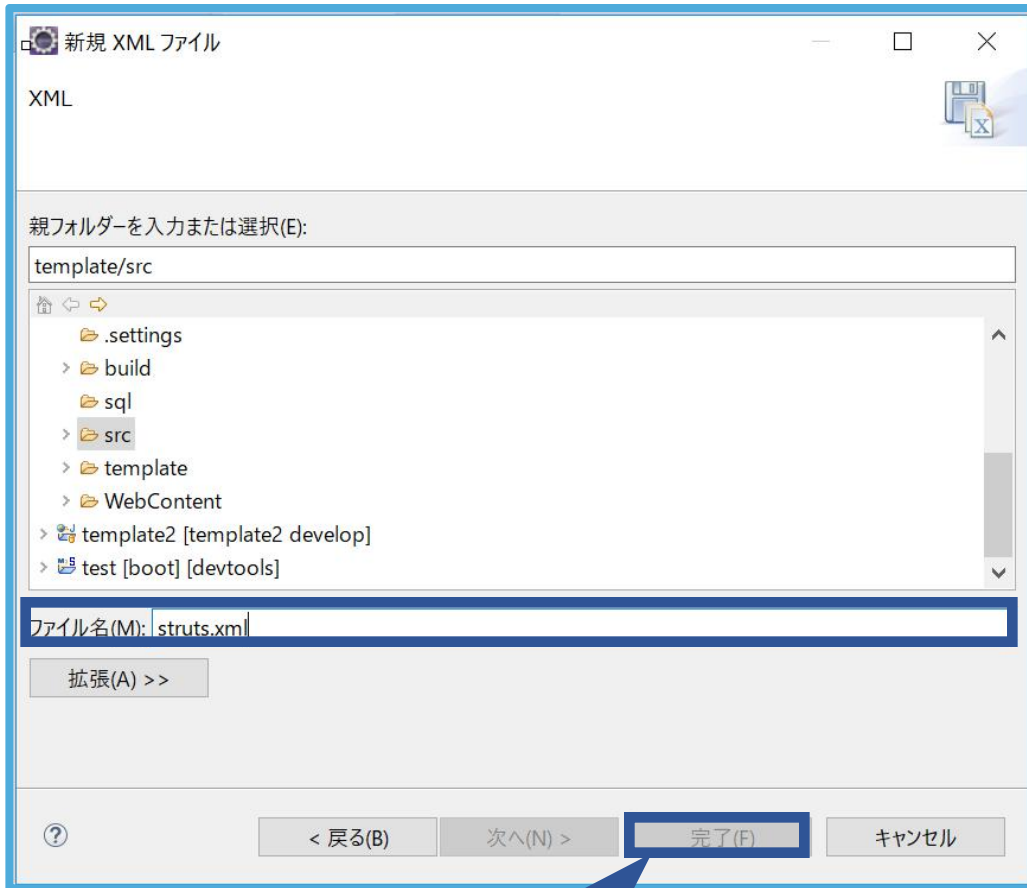


① 「Javaリソース」「src」を選択し、「プロジェクトを右クリック」「新規」「その他」を選択します。

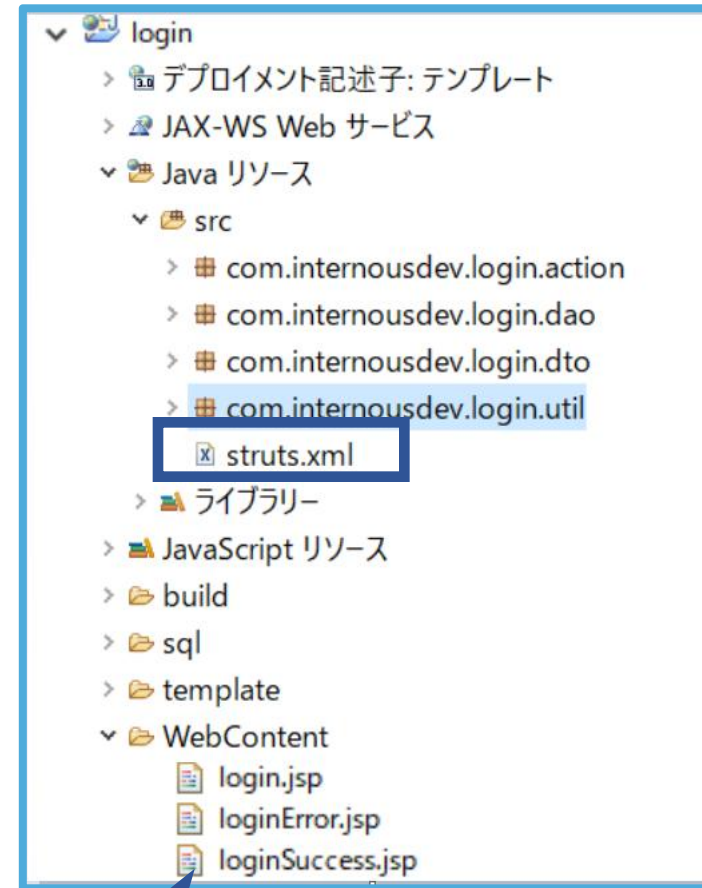


② 「ウィザード(W):」欄の「XML」「XMLファイル」を選択し、完了ボタンをクリックします。

# 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。

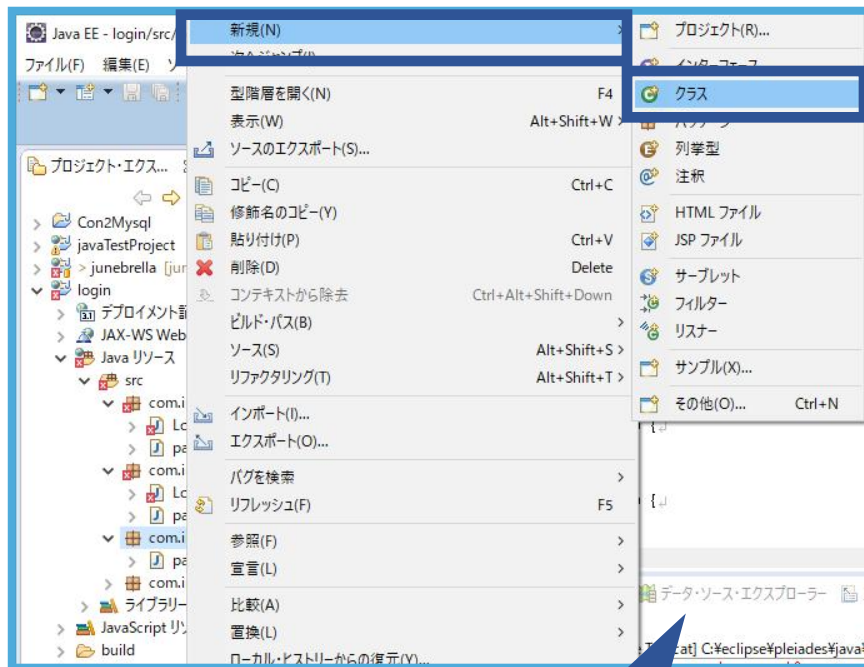
```
</action>
```

# DBConnectorの作成

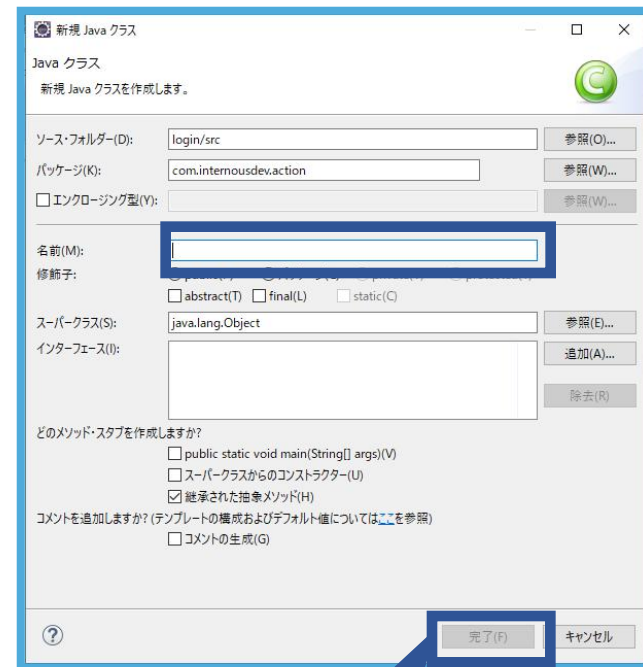
## 解説

DBに接続するときに接続開始(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「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.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";
```

```
    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。

接続情報から自分のパソコンにインストールされているMySQLへ接続する準備が整います。

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;  
}  
}
```

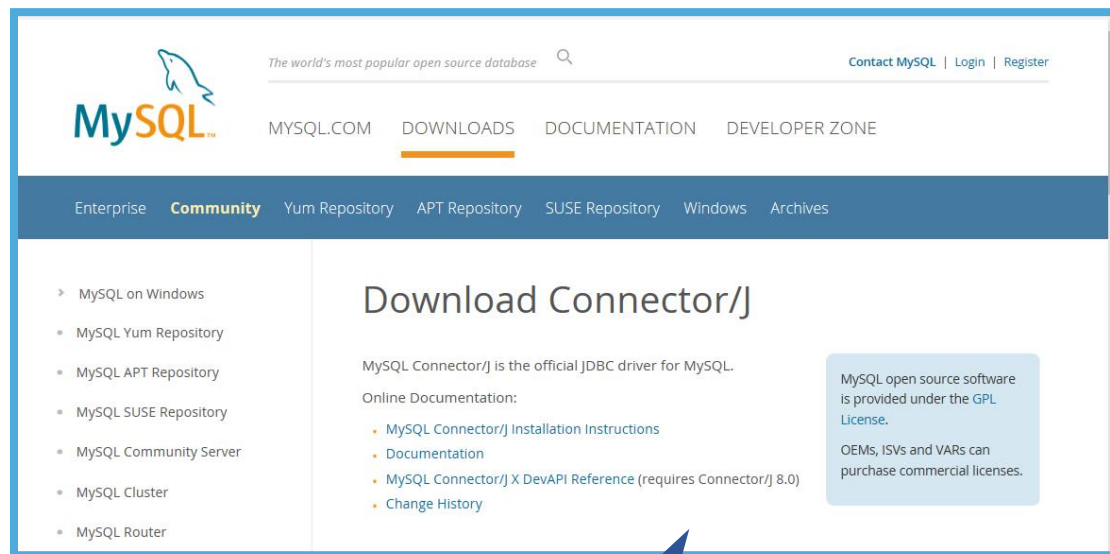
Mysqlに接続できたか情報を渡します。

# MySQL JDBC ドライバーの配備

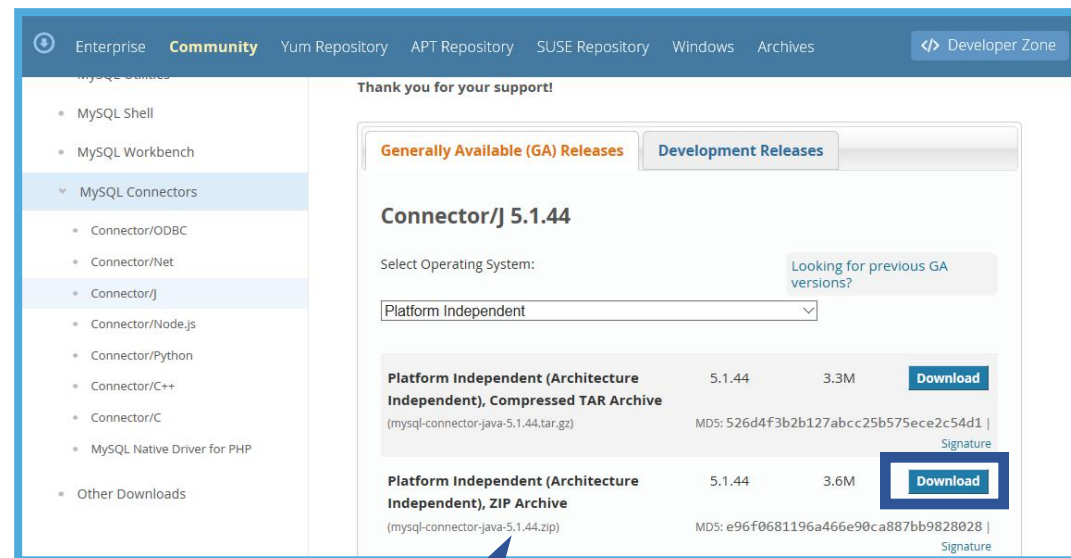
解説

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



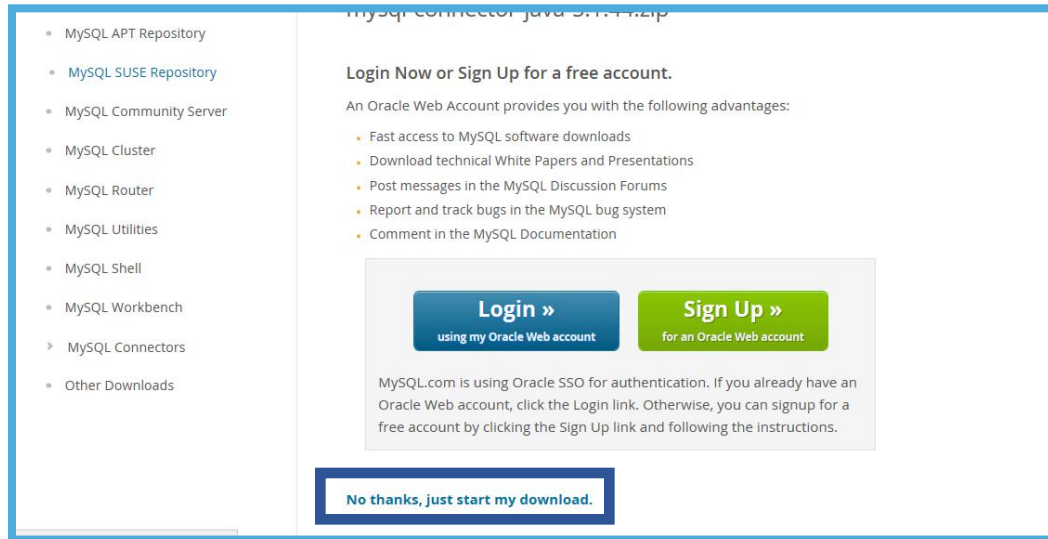
① 「<https://dev.mysql.com/downloads/connector/j/>」にアクセスします。



② 下へスクロールして、「zip」形式のDownloadボタンをクリックします。



# MySQL JDBC ドライバーの配備

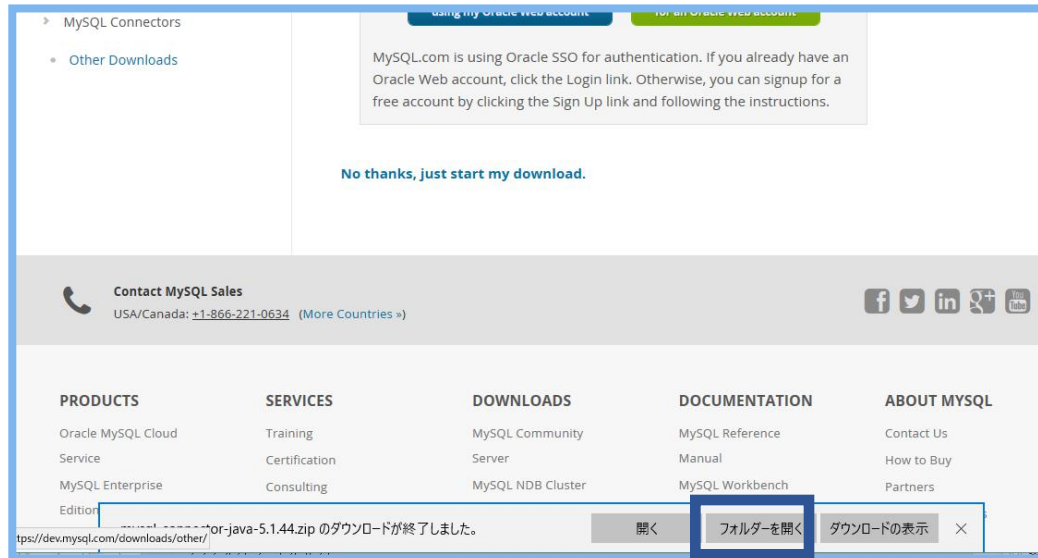


③ 遷移した先の画面でも、下へスクロールし、「No thanks, just start my download.」リンクをクリックします。

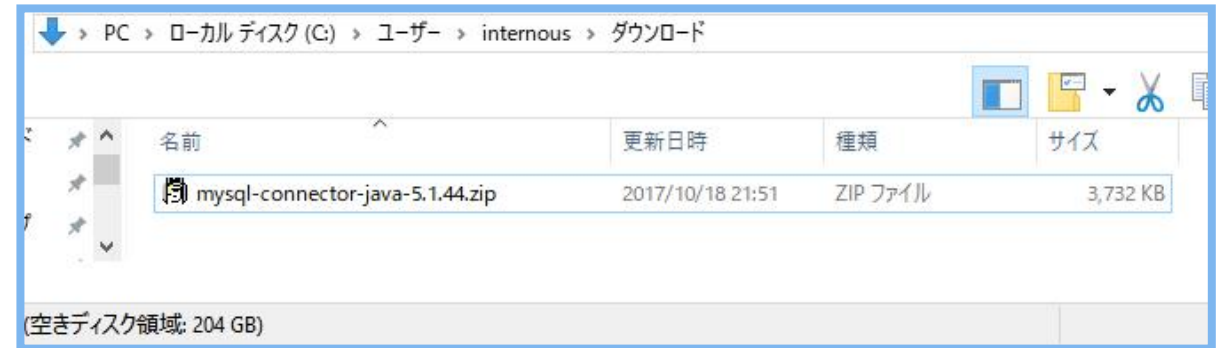


④ 「保存」ボタンをクリックします。

# MySQL JDBC ドライバーの配備

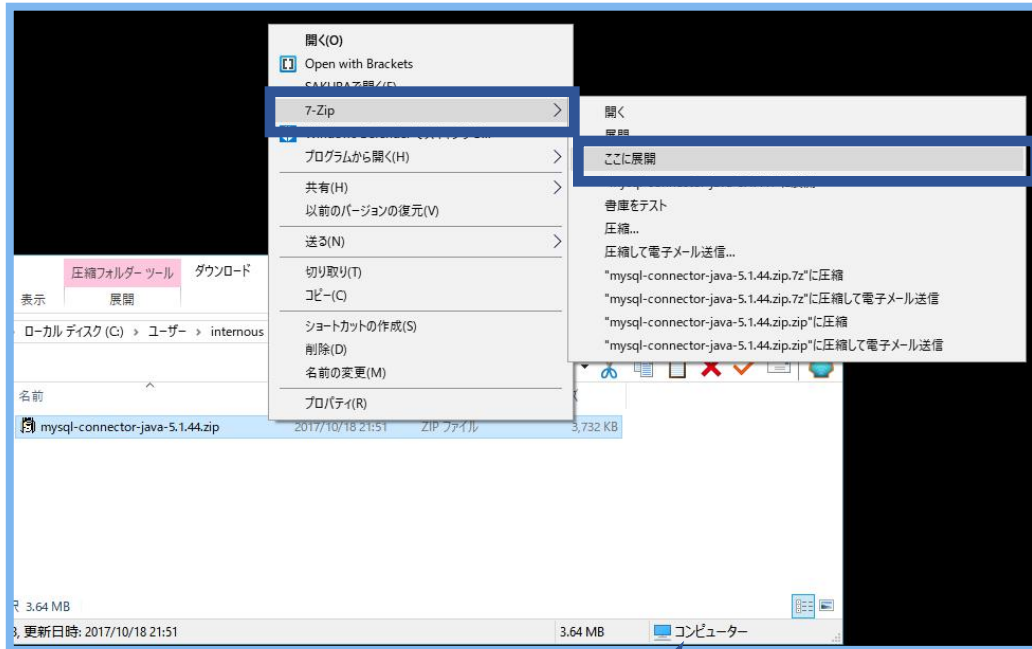


⑤ ダウンロードが完了後「フォルダを開く」ボタンをクリックします。

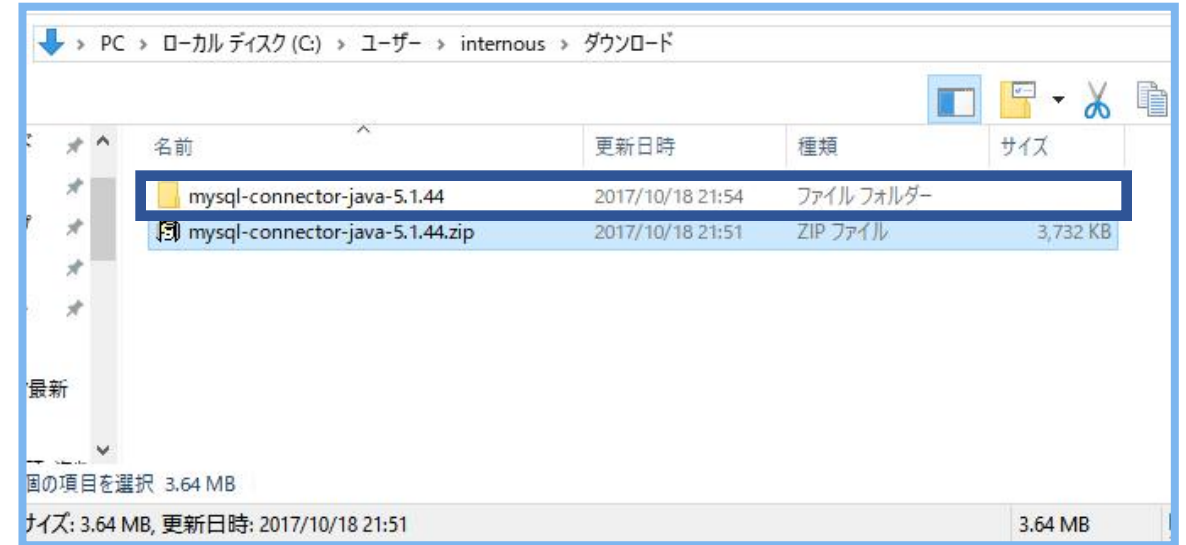


⑥ ダウンロードファイルが格納されている「ディレクトリ」が開きます。対象のディレクトリの中に「mysql-connector-java-x.x.xx.zip」を探します。

# MySQL JDBC ドライバーの配備

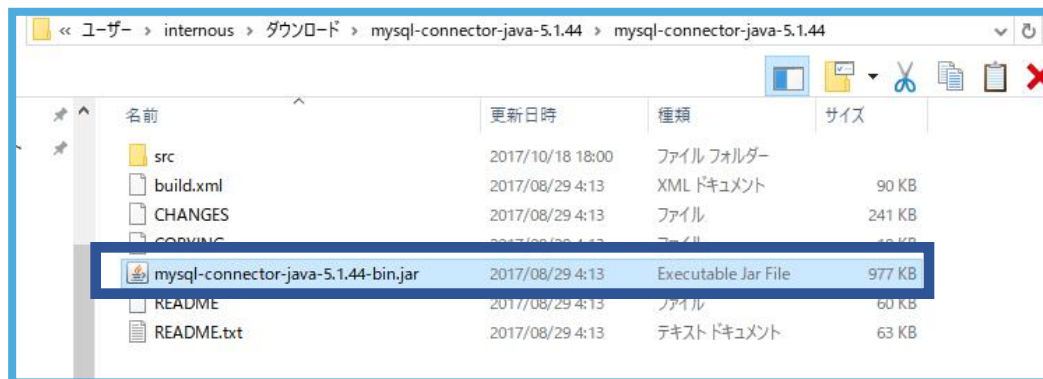


⑦ 「mysql-connector-java-x.x.xx.zip」を  
右クリックし、「解凍」または「展開」します。

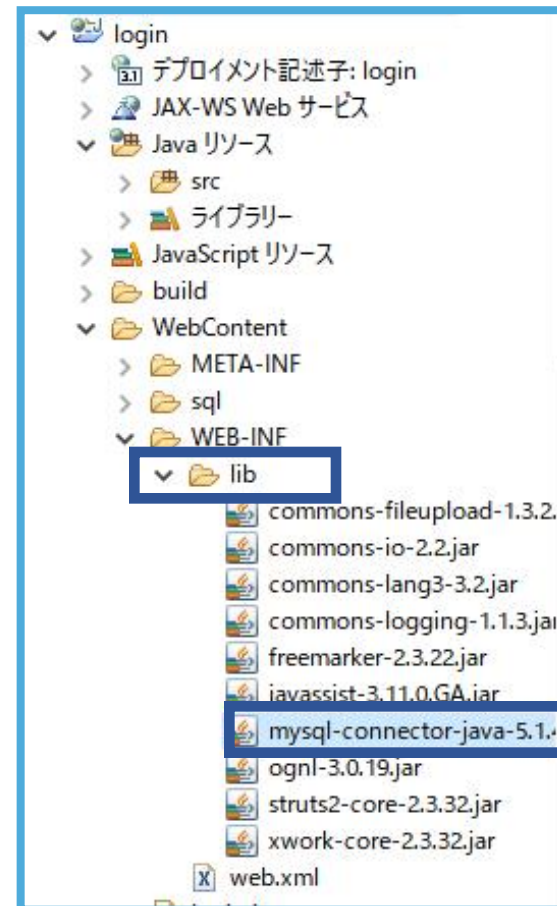


⑧ 解凍が終わると「mysql-connector-  
java-x.x.xx」という名前のディレクトリ  
が増えます。  
「mysql-connector-java-x.x.xx」をダ  
ブルクリックします。

# MySQL JDBC ドライバーの配備

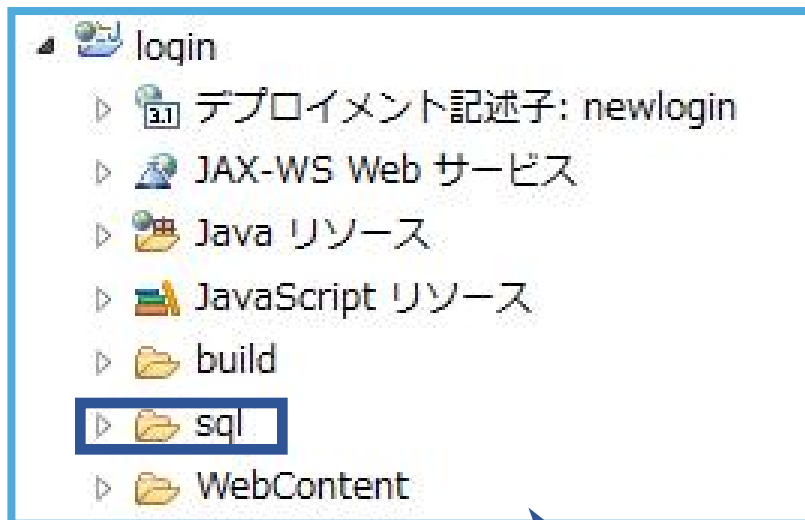


⑨ 「mysql-connector-java-x.x.xx」 ディレクトリの中に「mysql-connector-java-x.x.xx-bin.jar」があることを確認します。

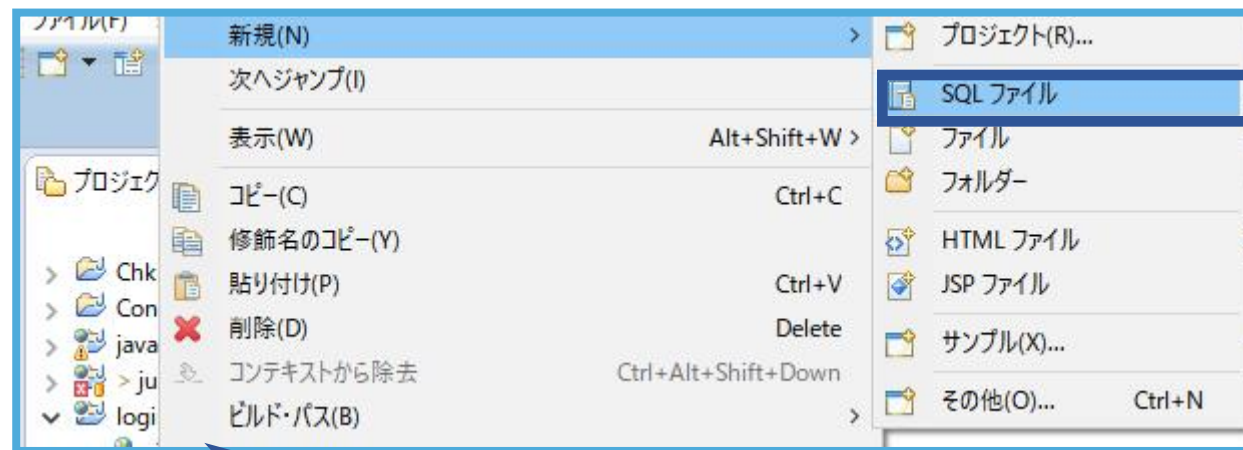


⑩ 「プロジェクト」「WebContent」「WEB-INF」「lib」の中に「mysql-connector-java-x.x.xx-bin.jar」をコピーします。

# SQLファイルの配備

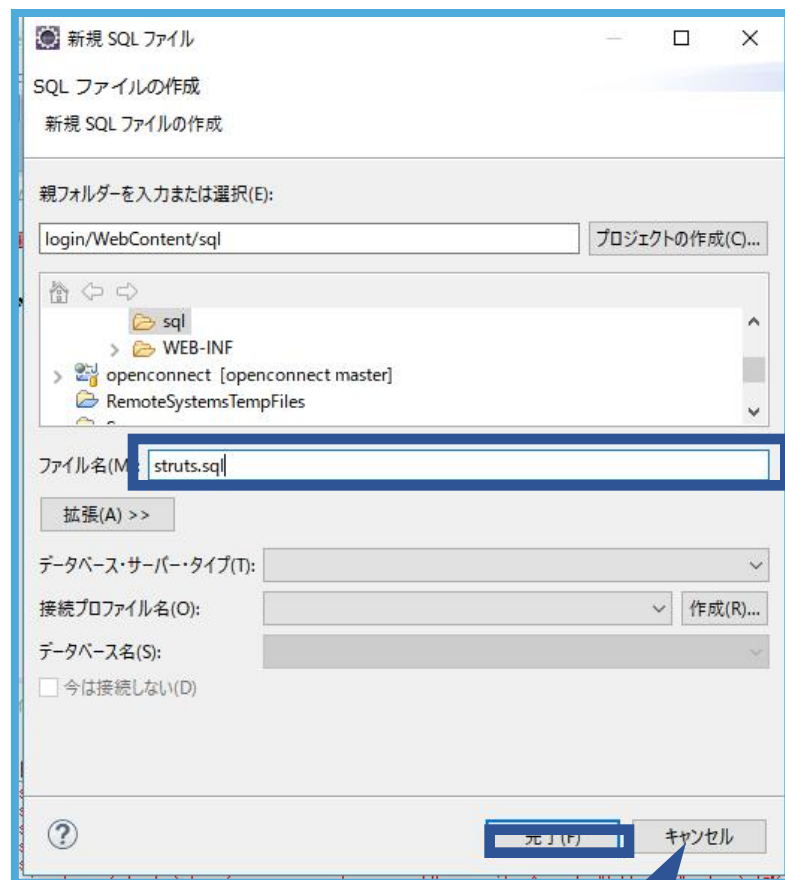


① 「プロジェクト」の直下に「sql」ディレクトリを作成します。

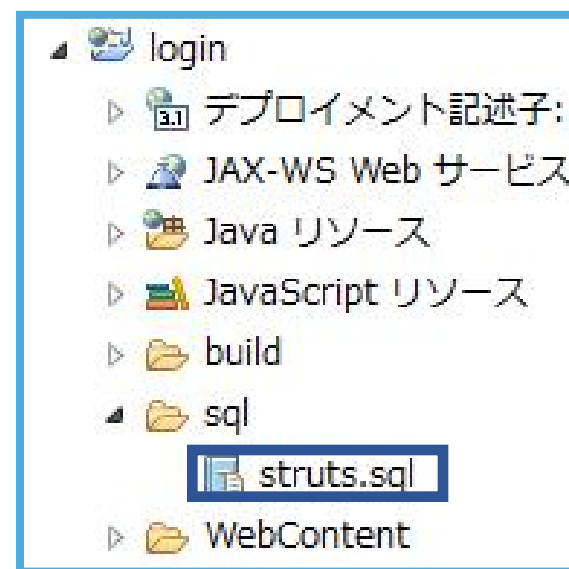


② 「sql」の上で「右クリック」「新規」「SQLファイル」を選択します。

# 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

③ 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」テーブルに情報を保存します。

**drop database if exists logindb;**

もし「logindb」というデータベースがあったら、削除します。

**create database logindb;**

「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。