

スッキリわかるサーブレット&JSP 入門 第 13 章を MySQL を使ってやってみる Ver1.4

Seiichi Nukayama

2023 年 12 月 30 日

目次

1	JDBC ドライバーをインストールする	1
1.1	ダウンロード	1
1.2	jdbc ドライバのインストール	1
1.3	補足	1
2	データを準備する	2
2.1	MySQL: ユーザーの作成	2
2.2	データベースの作成	2
2.3	サンプルデータを入れる	3
3	サンプルプログラム (p383) の作成	4

1 JDBC ドライバーをインストールする

JDBC ドライバーを Windows にインストールするのは、ネットで調べてもちょっとわかりにくいかもしれない。情報が古いこともある。また、Oracle が Web サイトをけっこう頻繁に模様替えしてるのもある。

1.1 ダウンロード

まず、ここに行く。

<https://www.mysql.com/jp/products/connector/>

JDBC Driver for MySQL(Connector/J) の行の ダウンロード をクリックする。

<https://dev.mysql.com/downloads/connector/j/> のページに遷移するので、このところからインストールする。

“Connector/J 8.2.0” と書かれているところの “Select Operating System...” から、Platform independent を選択する。

二つの Download ボタンがある。

上は、“TAR Archive” 用である。

下は、“ZIP Archive” 用なので、これをダウンロードする。

1.2 jdbc ドライバのインストール

ダウンロードした “mysql-connector-j-8.2.0.zip” を適当なフォルダで展開する。すると、中に “mysql-connector-j-8.2.0.jar” がある。これがドライバファイルである。

このドライバファイルは、“src/main/webapp/WEB-INF/lib” に置く。そして、ビルドパス — ビルドパスに追加 としておく。

1.3 補足

MySQL Connector/J 8.0 については、以下のように書かれてある。

MySQL Connector/J 8.0 is highly recommended for use with MySQL Server 8.0, 5.7, 5.6, and 5.5. Please upgrade to MySQL Connector/J 8.0. (MySQL Connector / J 8.0 は、MySQL Server 8.0、5.7、5.6、および 5.5 で使用することを強くお勧めします。 MySQL Connector / J 8.0 にアップグレードしてください。by Google 翻訳)

MySQL5.7 などでも使えるようである。

2 データを準備する

p383 のコードを入力する前に、このコードで使っているデータを MySQL で準備しておかなくてはならない。

2.1 MySQL: ユーザーの作成

XAMPPなどで、MySQLを動作させておく。

まず、MySQL で使用するユーザーを用意しておく。ROOT のままだと、他のデータベースにもアクセス可能なので、実際にはそのデータベースにのみアクセス権限が与えられているユーザーを作成することになる。しかし今はお試しでプログラムを作成しているので、ユーザー名とパスワードは簡単なものにしておく。

ユーザー名: sa
パスワード: (なし)

本 (p383) ではパスワードは設定されていないので、同じようにしておく。

1. MySQL にルートでログインする。

Windows のコマンドプロンプトで、以下のようにする。

```
> mysql -u root -p  
Password: ****
```

(多くの場合、パスワードは設定されていないか、もしくは root である)

2. ユーザーを作成する。

```
mysql> create user 'sa'@'localhost' identified by '';
```

(ユーザー”sa”を作成し、パスワードを””としている)

3. そのユーザーにこれから作成するデータベースへの権限を与える。

```
mysql> grant all on example.* to 'sa'@'localhost';
```

(データベース名は “example” で、それに関連する全てのファイルにアクセス権を与える)

4. MySQL をログアウトする

```
mysql> quit;
```

以下のようにすると、ユーザーの作成とデータベースへの権限付与は同時に行うことができる。

```
mysql> grant all on example.* 'sa'@'localhost' identified by '';
```

2.2 データベースの作成

先ほど作成したユーザーで MySQL にログインする。

```
> mysql -u sa -p
Password: (そのまま Enter)
```

データベース (example) を作成する。

```
mysql> create database example ;
```

テーブル (employee) を作成する。

```
mysql> create table employee (
-> id char(6) primary key,
-> name varchar(100) not null,
-> age int not null) ;
```

2.3 サンプルデータを入れる

サンプルデータを入れる。データベースアプリを作るときは、最初にサンプルデータを入れておくようにする。

```
mysql> insert into employee values ( 'EMP001', '湊 雄輔', 23);
mysql> insert into employee values ( 'EMP002', '綾部 みゆき', 22) ;
```

確認する。

```
mysql> select * from employee;
+-----+-----+-----+
| id      | name          | age |
+-----+-----+-----+
| EMP001  | 湊 雄輔        | 23  |
| EMP002  | 綾部 みゆき    | 22  |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

データがちゃんと入っている。^{*1}

^{*1} Windows では文字コードが Shift-JIS なので、漢字がうまく表示できない場合がある。そのときは、とりあえずローマ字など英字で入れておく。プログラムを実行したときに入力に漢字 (UTF-8) が使えればよい。

3 サンプルプログラム (p383) の作成

p383 に掲載されているサンプルプログラムの MySQL 版を書いてみる。

新規動的 Web プロジェクトを作成する。プロジェクト名は “mysql”。

”src” フォルダで以下のコードを書く。パッケージを作るなら ”terminal” とでもしておく。

Listing 1 src/SelectEmployeeSample.java

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.sql.SQLException;
6
7 public class SelectEmployeeSample {
8
9     static final String USERNAME = "sa";
10    static final String PASSWORD = "";
11    static final String CONNECT =
12        "jdbc:mysql://localhost:3306/example";
13
14    public static void main( String[] args ) {
15        // MySQLドライバーをDriverManagerに登録
16        try {
17            Class.forName("com.mysql.cj.jdbc.Driver");
18        } catch (ClassNotFoundException e) {
19            throw new IllegalStateException("ドライバーが見つかりません");
20        }
21        // データベースに接続
22        try (Connection conn =
23            DriverManager.getConnection( CONNECT, USERNAME, PASSWORD )) {
24            // select 文
25            String sql = "SELECT id, name, age FROM employee";
26            PreparedStatement pstmt = conn.prepareStatement( sql );
27            ResultSet rs = pstmt.executeQuery();
28
29            while( rs.next() ) {
30                String id = rs.getString("id");
31                String name = rs.getString("name");
32                int age = rs.getInt("age");
33
34                System.out.println("ID:" + id);
35                System.out.println("名前:" + name);
36                System.out.println("年齢:" + age + "\n");
37            }
38        } catch (SQLException e) {
39            e.printStackTrace();
40        }
41    }
42 }
```

Timezone が設定されていないというエラーが出る場合、以下のようにする。

```
static final String CONNECT = "jdbc:mysql://localhost:3306/example?serverTimezone=JST";
```

これを 実行 – Java アプリケーション とすると、以下のように出力される。

ID:EMP001

名前: 湊 雄輔

年齢:23

ID:EMP002

名前: 綾部 みゆき

年齢:22