

4-5-2.データベースへ接続するプログラム

データベースへ接続するプログラム

はじめに

データベースの種類や接続方法によって細かな設定の違いはありますが、本章で学ぶ記述をマスターすれば、どのようなデータベースでもスムーズに接続を行うことができます。Javaからデータベースに接続するための基本的かつ一般的な記述を学んで行きましょう！

Step1 : 接続するプログラム

DBAccessクラスは、データベースへの接続と切断の処理を行なうプログラムです。クラス名に関しては、DBAccess にする必要はありません。クラス名は現場によって変わると思います。重要なのは中の処理です。例題を元にデータベースとの接続の仕方と切断の仕方を学んで行きましょう。

例題

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBAccess {

    /** ドライバーのクラス名 */
    private static final String POSTGRES_DRIVER = "org.postgresql.Driver";
    /** ・JDBC接続先情報 */
    private static final String JDBC_CONNECTION = "jdbc:postgresql://localhost:5432/lesson_db";
    /** ・ユーザー名 */
    private static final String USER = "postgres";
    /** ・パスワード */
    private static final String PASS = "postgres";

    public static void main(String[] args) {

        Connection connection = null;

        try {
            // データベースに接続する準備
            Class.forName(POSTGRES_DRIVER);
            // 接続先の情報
            connection = DriverManager.getConnection(JDBC_CONNECTION, USER, PASS);

            // forName()で例外発生
        } catch (ClassNotFoundException e) {
            e.printStackTrace();

            // getConnection()で例外発生
        } catch (SQLException e) {
            e.printStackTrace();

        } finally {
            try {
                if (connection != null) {
                    // データベースを切断
                    connection.close();
                }

            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Step2 : DBAccessを読み解く

```
Class.forName(POSTGRES_DRIVER);
Connection connection = DriverManager.getConnection(JDBC_CONNECTION,USER,PASS);
```



Class.forName()

上記の1行目のコードで`Class.forName()`メソッドにJDBCドライバ名を文字列で与え、JDBCドライバをロードしています。つまり、データベースへ接続する下準備ということです。

DriverManagerクラス

2行目のコードは、`DriverManager`クラスの`getConnection()`メソッドを使用してデータベースへの接続を行います。この`getConnection()`メソッドには、**3つの引数**を与えます。
`jdbc:postgresql://【ホスト名】/【データベース名】【ユーザ名】【パスワード】`
これで接続したいデータベースを指定することができます。

また、**Connectionインタフェース**や**DriverManagerクラス**は、`java.sql`パッケージ内のクラスです。なので一行目・二行目で、`import java.sql.Connection`と`import java.sql.DriverManager`をインポートしています。

connection.close()

接続の時に作られた`connection`オブジェクトを切断（開放）するために`close()`メソッドを実行します。これで、データベースから切断されます。
確実に切断を行うには、`finally`句に記述し、`Connection`が接続されているか（`null`でないか）の判定を行うよう記述します。
切断する理由はSQL文がデータベース上に滞留し処理が進まなくなり、エラーになったりします。なので、DBに接続したら、必ず**切断**するよう記述します。

補足

```
/** ドライバーのクラス名 */
private static final String POSTGRES_DRIVER = "org.postgresql.Driver";
/** ・JDBC接続先情報 */
private static final String JDBC_CONNECTION = "jdbc:postgresql://localhost:5432/lesson_db";
/** ・ユーザ名 */
private static final String USER = "postgres";
/** ・パスワード */
private static final String PASS = "postgres";
```

上記のように、あらかじめ定数として記述することがベターです。
理由はいくつかあります。
1,変更の多い値ではない。
2,なんとも同じように、ホスト名やデータベース名を記述する必要がない。
3,コーディングのケアレスミスをなくす。
このあたりもポイントになるので、しっかり抑えて起きましょう。

まとめ

1. `Class.forName()` でDB接続の準備
 2. 接続処理
 3. 切断処理
- 上記3つの流れはしっかり頭に入れておきましょう！

課題

提出課題はありませんので、一通り学習が終わったら次の章に進んで下さい。

最終更新日時: 2022年 09月 10日(土曜日) 08:21

