

Java



盛岡情報ビジネス&デザイン専門学校

★☆☆★本日の内容★☆☆★

1. PostgreSQLでのID自動採番
2. PostgreSQLでの日付関数
3. Javaのバージョンについて
4. PostgreSQLでのJDBC



◆PostgreSQLでのID自動採番

MySQL と PostgreSQL では自動採番の書き方に差があります。
PostgreSQL の場合は自動採番用の SERIAL 型が用意されています。

MySQLの場合

```
CREATE TABLE students (  
  id INT AUTO_INCREMENT,  
  name VARCHAR(32),  
  age INT,  
  PRIMARY KEY(id)  
);
```

PostgreSQLの場合

```
CREATE TABLE students (  
  id SERIAL,  
  name VARCHAR(32),  
  age INT,  
  PRIMARY KEY(id)  
);
```



◆PostgreSQLでのID自動採番

自動採番したい場合は値に **DEFAULT** を指定します。

```
INSERT INTO [テーブル名] VALUES( DEFAULT , [値2] , [値3] );
```



実際に書くと・・・

```
INSERT INTO students VALUES(DEFAULT, '高橋' , 28 );
```



◆PostgreSQLでの日付関数

現在日時を INSERT したい場合は下記のように書きます。

テーブル定義

```
CREATE TABLE teacher (  
  id SERIAL,  
  name VARCHAR(64) not null,  
  age INTEGER,  
  created_at TIMESTAMP,  
  primary key(id)  
);
```

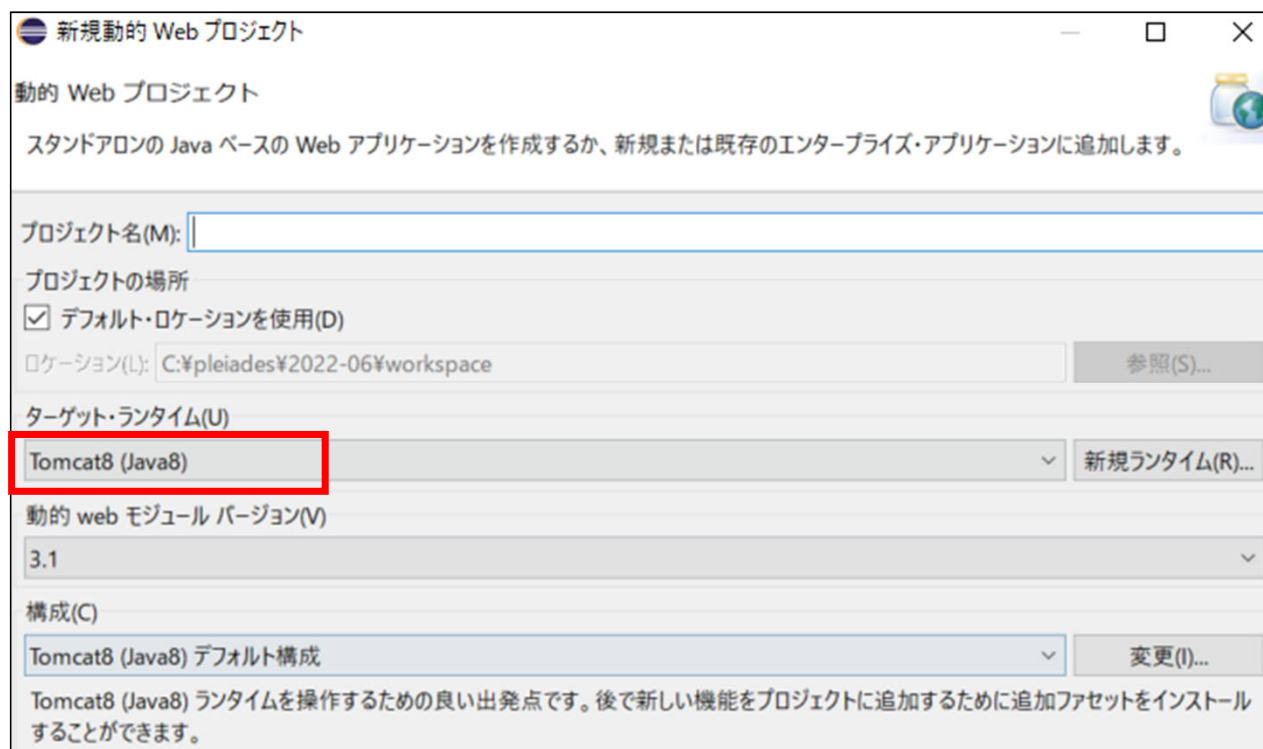
INSERT文

```
INSERT INTO teacher VALUES(default, 'takahashi', 30, current_timestamp);
```



◆Javaのバージョンについて

Heroku の Tomcat サーバが Java17 に対応していないので、今後のターゲットランタイムは Tomcat8 (Java8) を指定します。



新規動的 Web プロジェクト

動的 Web プロジェクト

スタンドアロンの Java ベースの Web アプリケーションを作成するか、新規または既存のエンタープライズ・アプリケーションに追加します。

プロジェクト名(M):

プロジェクトの場所

☒ デフォルト・ロケーションを使用(D)

ロケーション(L): C:\pleiades\2022-06\workspace 参照(S)...

ターゲット・ランタイム(U)

Tomcat8 (Java8) 新規ランタイム(R)...

動的 web モジュール バージョン(V)

3.1

構成(C)

Tomcat8 (Java8) デフォルト構成 変更(I)...

Tomcat8 (Java8) ランタイムを操作するための良い出発点です。後で新しい機能をプロジェクトに追加するために追加ファセットをインストールすることができます。



◆PostgreSQLでのJDBC

基本的にやり方は変わりません。

今後は下記の2環境を使ってアプリを構築していきます。

テスト環境(localhostのPostgreSQL)

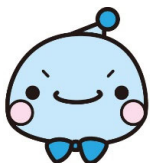
本番環境(HerokuのPostgreSQL)

そのため、JDBC 接続のコードを環境変数を使うように変更します。

なお、JDBCドライバは下記のフォルダにあります。

C:¥Program Files (x86)¥PostgreSQL¥pgJDBC¥

postgresql-42.2.18.jar



◆PostgreSQLでのJDBC

システム環境変数の登録

① 変数名を入力

ex) DATABASE_URL

② PostgreSQLサーバへのURIを登録

ex)

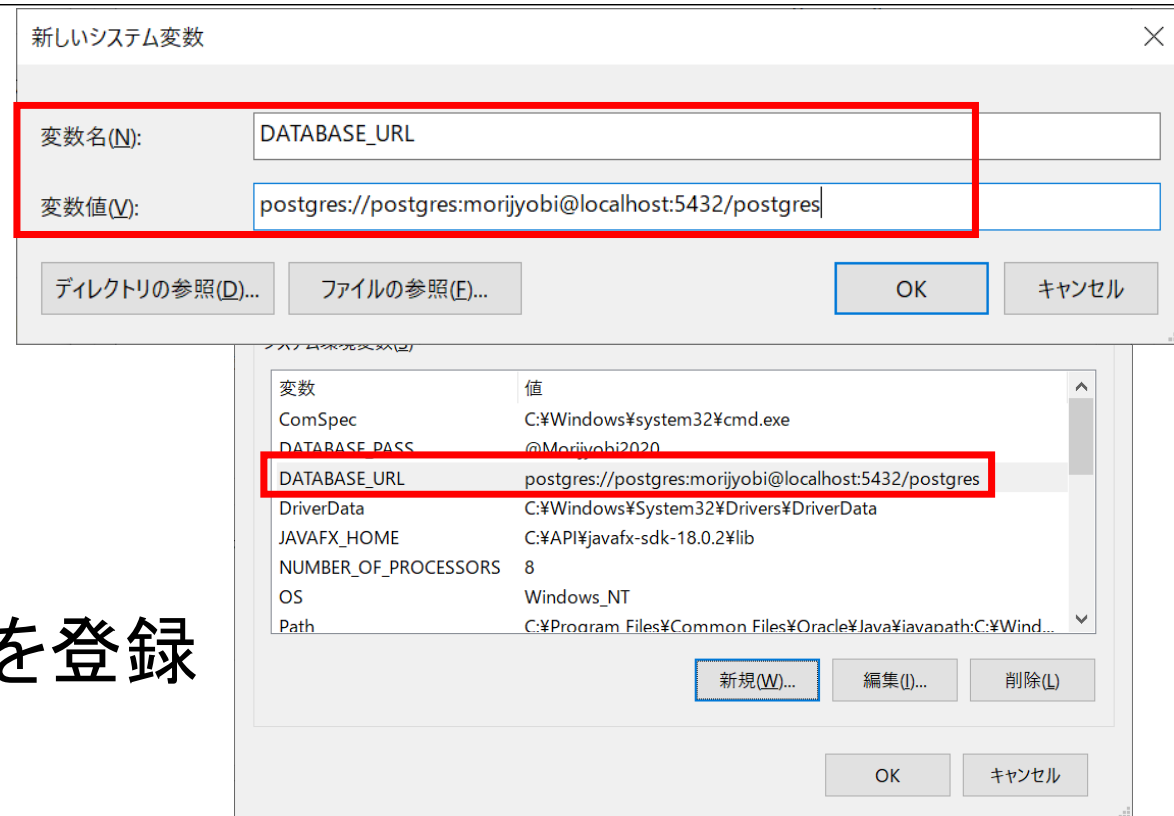
postgres://[ユーザ名]:[PW]@[ホスト名]:[ポート]/[データベース名]

postgres://postgres:password@localhost:5432/postgres

※ローカルのPostgreSQLのデフォルトのDB名は postgres です。

※passwordは自分のpostgresユーザのパスワードに置き換えてください。

③ 反映するために Windows を再起動します。



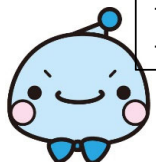
◆PostgreSQLでのJDBC

コードはこんな感じ(詳細はサンプルコード参照)

```
// DBへのConnectionを取得 (本番環境/テスト環境 切り替え用)
private static Connection getConnection() throws URISyntaxException, SQLException {
    try {
        Class.forName("org.postgresql.Driver"); // ドライバのロード
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
    URI dbUri = new URI(System.getenv("DATABASE_URL")); // 環境変数からURIを取得

    String username = dbUri.getUserInfo().split(":")[0];
    String password = dbUri.getUserInfo().split(":")[1];
    String dbUrl = "jdbc:postgresql://" + dbUri.getHost() + ':' + dbUri.getPort()
        + dbUri.getPath();

    return DriverManager.getConnection(dbUrl, username, password);
}
```



◆PostgreSQLでのJDBC

Heroku 上でも同様に同じ名前の環境変数を設定することができます。

そうすることにより、ソースコードを変えずに動いている環境(localhost / Heroku)によって、適切に接続先のDBサーバを切り替えることができます。

また、ソースコード上にユーザID、PWの情報も書かないため、GitHubなどでソースコードを公開するときにも便利です。

Config Vars

Config vars change the way your app behaves. In addition to creating your own, some add-ons come with their own.

Config Vars

Hide Config Vars

DATABASE_URL

postgres://asetnfuvknphcm:3fa3da474e08ca0



KEY

VALUE

Add

参考: Heroku の環境変数の設定画面

