

Zen v14 Java Class Library サンプル

Zen v14 Java Class Library を使用した簡単なサンプルを提示します。

このサンプルは、デモのデータベース (DEMODATA) の Course テーブルを参照し、キー (Course_Name) 値が「THE 1 0 1」およびそれ以降のデータを読み込んで表示します。

◆環境設定

環境変数 PATH に JAVA ランタイムが存在するフォルダーのパスを追加し、環境変数 CLASSPATH に Zen v14 Java Class Library (psql.jar) が存在するフォルダーのパスを設定します。

JAVA ランタイムは、Zen v14 をインストールしたフォルダーにコピーされています。

```
set path=%path%"C:¥Program Files (x86)¥Actian¥Zen¥bin¥jre¥bin";
set zenpath="C:¥Program Files (x86)¥Actian"
set classpath=.;%zenpath%¥Zen¥bin¥psql.jar;
```

◆コンパイル

コンパイルを行うには、JDK 8 等をインストールしておく必要があります。

次のようなコマンドでコンパイルを行います。

```
C:¥test¥java-se-8u41-ri¥bin¥javac.exe -cp "C:¥Program Files(x86)¥Actian¥Zen¥bin¥psql.jar" jclsample
idx.java
```

※JDK 8 を C:¥test に展開していることを前提としています。

◆実行

次のようなコマンドで実行します。

```
java -cp .;"C:¥Program Files (x86)¥Actian¥Zen¥bin¥psql.jar" JCLSampleIdx
```

◆サンプルコード

```
// Sample Java Class Library Application for PSQL
```

```
import pervasive.database.*;
```

```
public class JCLSampleIdx {
```

```
    public static void main(String[] args)
```

```
    {
```

```
        if (args.length == 0) {
```

```
            new JCLSampleIdx("btrv:///demodata");
```

```
        }
```

```
        else
```

```
            new JCLSampleIdx(args[0]);
```

```
    }
```

```
    public JCLSampleIdx(String dbNameURI) {
```

```
        System.out.println("¥n¥nHello World JCL Example for Pervasive PSQL¥n");
```

```
        System.out.print("This application uses the Pervasive PSQL Java Class Library");
```

```
        System.out.println(" to connect to the sample DEMODATA db and display the Course table.¥n");
```

```
        // String dbNameURI = "btrv:///demodata";
```

```
        // セッションオブジェクトを生成します。
```

```
        Session session = Driver.establishSession();
```

```
        try {
```

```
            // データベース URI でデータベースに接続します。
```

```
            Database db = session.connectToDatabase(dbNameURI);
```

```
            // アクセスするテーブルを i 指定します。
```

```
            Table table = db.getTable("Course");
```

```
            // レコードを格納する領域（オブジェクト）を生成します。
```

```

RowSet rowset = table.createRowSet();

// レコードのメタデータを取得します。(インデックス使用に必要)
RowSetMetaData rsmd =rowset.getRowSetMetaData();

// インデックスを使用して読み込む際に使用するインデックスを指定します。
rsmd.setAccessPath("Course_Name");

// Index 要の領域 (オブジェクト) を生成します。
Index index = rowset.createIndex("Course_Name");

// インデックスを使用して読み込む際の値を設定します。
// インデックスが複数項目で構成される場合、全ての項目を設定します。
index.setString("Name", "THE 101");
try {
// インデックスを使用して、条件に一致するデータ (Name が"THE 101"のデータ)
// を読み込みます。
    Row row = rowset.getByIndex(pervasive.database.Consts.BTR_EQ, index);
    System.out.println("Courses at University of Pervasive");
    System.out.println("=====¥n¥n");

    String cname    = row.getString(0);
    String cdesc    = row.getString(1);
    String credits  = row.getString(2);
    String deptname = row.getString(3);
    System.out.println(cname+" "+credits+" "+deptname+"¥t"+cdesc);
}
catch(PsqlOperationException ex) {
    if (ex.getErrorCode() != 4)    throw(ex);
}
try {

// PsqlEOFException がスローされるまで (EOF になるまで) 繰り返します。
    while(true) {
// 次のレコードを読み込みます。
        Row row = rowset.getNext();
        String cname    = row.getString(0);
        String cdesc    = row.getString(1);

```

```
        String credits = row.getString(2);
        String deptname = row.getString(3);
        System.out.println(cname+" "+credits+" "+deptname+"¥t"+cdesc);
    }
}
```

```
catch(PsqlEOFException ex) {
    System.out.println("Encountered end of table.");
}
```

```
}
```

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

```
// No more rows
```

```
System.out.println("Ending sample...");
```

```
}
```

```
}
```