

Pervasive PSQL 用の OData Java プロデューサーを作成する

このドキュメントは、「[Java OData アプリケーションから Pervasive PSQL データへアクセスする](#)」ドキュメントと対応関係にあります。そのドキュメントでは、OData ソリューションに用いられるコンシューマー (クライアント アプリケーション) を作成する方法について説明しています。一方、このドキュメントでは、Pervasive PSQL データを Java クライアント アプリケーションで利用できるようにするために用いた手順の概要を説明します。これは OData 用語で表現すると、Java OData プロデューサーを作成する手順です。

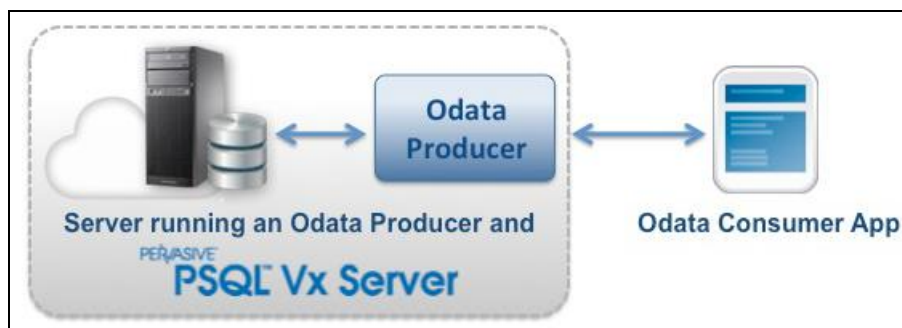
ここでは、以下の項目について説明します。

- [Open Data Protocol および Odata4j について](#)
- 環境を設定する
- プロデューサーを実行する
- プロデューサーを編集する
- プロデューサーを再コンパイルする
- 最後に

Open Data Protocol および Odata4j について

OData は、Microsoft Open Specification Promise に従ってリリースされた Web ベースのプロトコルです。これは、標準化された方法でさまざまなソースのデータにアクセスするよう設計されています。OData は、HTTP、Atompub (Atom Publishing Protocol) および JSON (JavaScript Object Notation) など既存の標準 Web テクノロジーに基づいて構築されています。OData クライアント ライブラリは Java 仮想マシンをサポートするデバイス用に存在しています。このクライアント ライブラリは OData の詳細を抽象化するので、Java を使用したアプリケーションを作成することができます (OData プロトコルやライブラリなどの詳細については、OData Web サイト (<http://www.odata.org/>) を参照してください)。

OData ソリューションは従来のクライアント/サーバー アーキテクチャに似ています。OData の用語で、サーバーは "プロデューサー" と言い、そのデータを利用できるクライアントを "カスタマー" と言います。Pervasive PSQL の場合、プロデューサーは Pervasive PSQL データベース定義を行い、コンシューマー アプリケーションがそのデータを利用できるようにします。



Odata4j

Odata4j はオープンソース ツールキットで、Java アプリケーション用の OData プロデューサーとコンシューマー フレームワークの両方を提供します。このツールキットの情報については、Web サイト (<http://code.google.com/p/odata4j/>) を参照してください。

現在、odata4j ツールキットはデータ リポジトリとしてリレーショナル データベースはサポートしません。odata4j のコミュニティは、JDBC によるデータベース アクセスに対応するよう要求しています。この対応が実現するのを待つことなく、Pervasive Software は odata4j を基盤として使用したサンプル プロデューサーを提供します。

Java 用の完全な OData ソリューションには、プロデューサーとコンシューマーの両方が必要であることに注意してください。私はデータの取得を示すサンプル コンソール アプリケーションを作成しました。コンシューマーの詳細については、「[Java OData アプリケーションから Pervasive PSQL データへアクセスする](#)」を参照してください。

環境を設定する

使用した環境設定は次のとおりです。

- Pervasive PSQL Vx Server 11 SP3 がインストールされているシステム。このシステムには JRE (Java Runtime Environment) もインストールされている必要があります。この JRE は、Pervasive PSQL でインストールされるバージョン、あるいは単独の別バージョンのどちらでも使用できます。
- Eclipse および JDK (Java Development Kit) がインストールされている開発システム。私は JDK 1.6.0_31 を使用しました。

プロデューサーを実行する

次の手順に従ってプロデューサーを実行してください。

- 1) ダウンロードしたアーカイブ ファイル [ODataJava.zip](#) をローカルのハードドライブに解凍します。
- 2) プロデューサーを実行する場所へ以下のアイテムをコピーします。
 - conf ディレクトリ
 - lib ディレクトリ
 - PsqLOdataServer.jar
- 3) conf ディレクトリにある psql-config.properties ファイルを編集します。ホスト名または IP アドレス部分を、Pervasive PSQL Vx Server が実行しているマシンのホスト名または IP アドレスに変更します。DATABASE 設定は DEMODATA のままにしておきます。

- 4) 次のコマンドを実行してプロデューサーを実行します。

```
java -jar PsqlOdataServer.jar
```

その結果、以下のメッセージが表示されます。

```
Please direct your browser to  
http://127.0.0.1:8020/JdbcOdataProducer3.svc Customers
```

```
3 07, 2013 6:01:24 午後  
com.sun.jersey.server.impl.application.WebApplicationImpl initiate
```

```
情報:Initiating Jersey application, version 'Jersey:1.1.5 01/20/2010  
03:55 PM'
```

```
3 07, 2013 6:01:25 午後  
com.sun.jersey.server.impl.application.WebApplicationImpl initiate
```

```
情報:Initiating Jersey application, version 'Jersey:1.1.5 01/20/2010  
03:55 PM'
```

```
3 07, 2013 6:01:25 午後  
org.odata4j.jersey.producer.server.ODataJerseyServer start
```

```
情報:Jersey app started with WADL available at  
http://127.0.0.1:8020/JdbcOdataProducer3.svcapplication.wadl
```

```
Press any key to exit
```

これでプロデューサーはリクエストを受け付けることができるようになります。たとえば、私が作成したサンプル コンソール アプリケーションを使用できます。「[Java OData アプリケーションから Pervasive PSQl データへアクセスする](#)」を参照してください。

プロデューサーを編集する

プロデューサーを編集したい場合、その設定ファイルにいくつかの変更を実装することができます。それ以外の変更はソースコードで行って再コンパイルする必要があります。

設定ファイルの変更

conf ディレクトリにある `psql-config.properties` ファイルで以下の項目を設定することができます。

- **CONNECT_STRING**
この設定は、Pervasive PSQl Vx Server への接続に必要な JDBC 接続文字列を指定します。Pervasive PSQl Vx Server が実行しているシステムと、プロデューサーが実行しているシステムが異なる場合は、URI のホスト名部分を `localhost` から正しいホスト名に変更できます。
- **DATABASE**
この設定はプロデューサーが公開するデータベースの名前を指定します。

- **USER**
この設定は、データベースへログインする際に使用するユーザー名を指定します。
- **PASSWORD**
この設定は、データベースへログインする際に必要となるパスワードを指定します。
- **TABLE_PATTERN**
この設定は、プロデューサーによって使用できるようにするテーブルを指定する正規表現を提供します。

ソースコードの変更

ソースコードで以下の項目を変更する必要があります。

- コンシューマーが接続する IP ポート番号
- OData サービス エンドポイントの名前

これらの変更はいずれも `JdbcOdataProducer.java` ファイルで行うことができます。このファイルは `PSQLODataJavaExample¥PSQLODataJavaProducer¥src¥odata4j¥producer` にあります。

次の定義を変更します。

```
public static final String endpointUri =  
"http://127.0.0.1:8020/JdbcOdataProducer3.svc";
```

この 8020 はポート番号で、`JdbcOdataProducer3.svc` は OData サービス エンドポイントの名前です。

また、設定ファイルでこれらに対応するオプションを追加することもできるでしょう。

ライブラリの更新

ダウンロード アーカイブ ファイル [ODataJava.zip](#) には Pervasive PSQJ JDBC と `odata4j.jar` ファイルが含まれています。これらのファイルを更新したい場合は、その更新したバージョンを `lib` ディレクトリにコピーしてください。その後、`PsqlOdataServer.jar` ファイルを再作成してください。

`odata4j` はそのコミュニティによって継続的に修正されているため、`odata4j` が更新されたライブラリで動作するよう、その他のコードの変更が必要となるかもしれません。

プロデューサーを再コンパイルする

Eclipse の自動ビルド機能を使用して、`.java` ファイルを自動的にビルドしました。次に、`jar` を作成するために `createproducerjar.bat` という名前のバッチ ファイルを作成しました。この `createproducerjar.bat` ファイルで指定されるパスは、ご自分の環境に合わせて変更してください。

最後に

ここでは OData プロデューサーを介して Pervasive PSQL データへ簡単にアクセスする方法を示しました。この方法を使用すれば、従来のデータベース アクセスをサポートしないモバイル デバイス向けのアプリケーションで、Pervasive PSQL をデータリポジトリとして機能させることができます。コンシューマー (クライアント アプリケーション) の作成方法については、関連ドキュメント「[Java OData アプリケーションから Pervasive PSQL データへアクセスする](#)」を参照してください。

Pervasive Software ソフトウェア開発技術者 Jan D.