

# Java OData アプリケーションから Pervasive PSQL データへアクセスする

---

Java アプリケーションは、Pervasive PSQL コンポーネントがサポートされないさまざまなモバイル デバイス上で実行します。しかし、Open Data Protocol (OData) を使用するようなアプリケーションであれば、データ リポジトリとして Pervasive PSQL を使用することができます。

このドキュメントは、「[Pervasive PSQL 用の OData Java プロデューサーを作成する](#)」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、Java OData ソリューションに用いられるプロデューサー (サーバー部分) 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、Java クライアント アプリケーションを作成するために使用し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 コンシューマ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](#)
- [サンプル カスタマー アプリケーションの作成および実行](#)
- [制限に関する注記](#)

## 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"、クライアントは "カスタマー" と言います。

## 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。

このツールキットは、本ドキュメントで説明する Java OData コンシューマーの基盤となります。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。

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。また、データベースを共有するよう設定できる基本的な Java プロバイダーも実装しました。プロデューサーの詳細については、「[Pervasive PSQL 用の OData Java プロデューサーを作成する](#)」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。これは OData プロデューサーが実行されるシステムです。
- 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。カスタマー アプリケーションをこのシステムで実行しましたが、JRE がインストールされている第 3 のシステムから実行することもできます。

## サンプル カスタマー アプリケーションの作成および実行

このカスタマー アプリケーションは odata4j サンプルの 1 つに基づいています。私のアプリケーションはシンプルなもの、DEMADATA の OData メタデータの出力に続き Room テーブルの全エンタリを出力します。プロデューサーとコンシューマーのサンプルは、odata4j の Web サイトからダウンロードした odata4j v0.7 (odata4j-archive-0.7.zip) のソースアーカイブ内にあります。<http://code.google.com/p/odata4j/downloads/list> を参照してください。

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 createconsumerjar.bat という名前のバッチ ファイルを作成しました。

次の手順に従ってコンシューマー アプリケーションを作成します。

1. Eclipse で、  
PSQLODataJavaExample\PSQLODataJavaConsumer\src\odata4j\consumer\ODataPSQLDemoDataReadExample.java ファイルを開きます。
2. run 関数を探し、URL の IP アドレスをプロデューサーが実行しているシステムの IP に変更します。
3. createconsumerjar.bat を実行してコンシューマー jar を再作成し、その jar をリモート システムに配置します。JDK を私の実践例とは異なるディレクトリにインストールされた場合は、createconsumerjar.bat ファイルを編集して jar コマンドのディレクトリを変更してください。
4. プロデューサー アプリケーションを実行します。詳細については「[Pervasive PSQL 用の OData Java プロデューサーを作成する](#)」を参照してください。
5. java -jar PsqlOdataTestClient.jar というコマンドを用いてコンシューマー アプリケーションを実行します。

出力の最後は次のようなデータになります。

```
Room:Gambill Building          501
  Building_Name:Gambill Building
  Number: 501
  Capacity: 50
  Type:Classroom
Room:Gambill Building          613
  Building_Name:Gambill Building
```

```
Number: 613
Capacity: 1
Type:Office
Room:Garrison Hall          110
Building_Name:Garrison Hall
Number: 110
Capacity: 2
Type:Office
Room:Garrison Hall          120
Building_Name:Garrison Hall
Number: 120
Capacity: 35
Type:Classroom
```

## 制限に関する注記

このサンプル アプリケーションには次のような制限事項があります。

- このアプリケーションはデータを照会するだけです。データの挿入、更新、削除は行いません。OData プロトコルが構築されるエンティティ データ モデルの場合、データの挿入、更新および削除操作を実行するにはそのテーブルで主キーが定義されている必要があります。DEMODATA には定義済みの主キーがありません。挿入、更新および削除操作の例を見るには、ODataTestServiceReadWriteExample.java ファイルを見直してください。このファイルは odata4j Web サイトからダウンロードされる odata4j-archive-0.7.zip 内のソース アーカイブ odata4j-examples-0.7.0-sources.jar の org¥odata4j¥examples¥consumer に含まれています。<http://code.google.com/p/odata4j/downloads/list> を参照してください。
- この Java コンシューマーに対応する Java プロデューサーの目的は、OData プロデューサーが、JDBC を使用して Pervasive PSQL の情報を表示できることを示すことです。このプロデューサーは堅牢性が非常に高いわけではなく、OData の仕様に完全には準拠しない可能性もあります。とはいえ、現在 odata4j の団体は JDBC プロデューサーがパッケージへネイティブに含まれるよう取り組んでいます。そのプロデューサーが利用可能になれば、私が(対応関係にあるドキュメントで)作成したプロデューサーは不要になります。

Pervasive Software ソフトウェア開発技術者 Jan D.