

記事

[Toshihiko Minamoto](#) · 2021年11月16日 4m read[Open Exchange](#)

ODataとInterSystems IRIS

ODataとは？

OData (Open Data Protocol) は、RESTful APIの構築と消費に関する一連のベストプラクティスを定義する[ISO/IEC認定](#)の[OASIS標準](#)です。ODataは、リクエストとレスポンスのヘッダー、ステータスコード、HTTPメソッド、URL変換、メディアタイプ、ペイロード形式、クエリオプションなどを定義するための多様なアプローチを気にせずに、RESTful APIを構築しながらビジネスロジックに専念するのに役立ちます。また、ODataには変更の追跡、再利用可能なプロシージャの関数/アクションの定義、および非同期/バッチリクエストの送信に関するガイダンスも提供されています（出典: OData.org）。

ODataのユースケース

開発作業なしで、相互運用可能な形式でデータをRESTサービスとしてデプロイする。

開発作業なしで、RESTを使って、BI、データ視覚化、ERP、CRM、ESB、ワークフローツール、およびエンジンでデータを消費できるようにする。

API管理ツールで企業データを仮想化する。

ODataは、これらのRESTful APIを使用して、SQLのようなクエリ機能を可能にするREST APIを実装するための標準的な方法を提唱しています。ODataは基本的に、RESTアーキテクチャスタイルを利用しながら、HTTP、JSON、およびATOMの標準プロトコルを土台に構築されたWeb用のSQLです（出典: [progress.com](#)）。

ODataは幅広く採用されています。以下をご覧ください。

Broad Adoption for OData



ODataはFHIRの実装に役立ちます。FHIR（Fast Healthcare Interoperability Resources Specification）は、医療情報を電子的に交換するための標準です。

FHIRを実質的に相互運用可能にため、\$searchパラメーターには[OData仕様で規定されたルールをシステムが使用することが推奨](#)されます。さらに、FHIRは、クライアントとの信頼関係を築くため、追加のセキュリティレイヤにOAuthも使用しています（出典: progress.com）。

ODataは、ページネーション、バッチリクエスト、およびJSON、ATOM、XMLなどのさまざまな形式をサポートしています。

ODataとInterSystems IRIS

InterSystems IRISはODataをサポートしていませんが、OData Server for InterSystems IRISを使用して、永続クラスをRESTとして公開することが可能です。

次の手順に従ってください。

IRIS OData Serverのソースコードのクローンを次のようにして作成します: `git clone https://github.com/yurimarx/isc-iris-odata.git`

次のフォルダに移動します: `isc-iris-odata` folder

次を実行します: `mvnw install` (MS Windows) または `./mvnw install` (Linuxまたはmac)

次を実行します: `docker build -t odata:1.0.0`

次を実行します: `docker run -p 8080:8080 odata:1.0.0` OData Serverが起動します:

6. 任意の永続クラスでInterSystems IRISインスタンスを起動します。
7. ブラウザで次にアクセスします: <http://localhost:8080/> 画面でパラメーターを設定します:
8. これは私のインスタンスのパラメーターです。
ご利用のIRISインスタンスにパラメーターを設定してください。ネームスペースにあなたのIRISのネームスペース、スキーマにSQLテーブルスキーマ、ポートにJDBCデータベース接続へのポートを設定します。
9. [送信] を押すと、OData Server
Dockerインスタンスが再読み込みされてパラメーターが適用されます。
10. <http://localhost:8080/odata.svc/>にアクセスし、IRISスキーマのすべての永続クラスを表示します。
私の場合は次のようになります。
11. 永続クラスに移動するには、次を参照します: <http://localhost:8080/odata.svc/<PersistentClass>> 例:
<http://localhost:8080/odata.svc/Animal>
12. ODataサーバーは次のようにAnimalデータをリスト表示します。
13. JSON形式で表示するには、次を参照します:
[http://localhost:8080/odata.svc/Animal?\\$format=application/json](http://localhost:8080/odata.svc/Animal?$format=application/json) 次に例を示します。
14. 行の詳細を表示するには、次を参照します:
[http://localhost:8080/odata.svc/Animal\(8\)?\\$format=application/json](http://localhost:8080/odata.svc/Animal(8)?$format=application/json)
15. 削除するには、Postmanで次を指定してDELETEを送信します: [http://localhost:8080/odata.svc/Animal\(8\)](http://localhost:8080/odata.svc/Animal(8))
16. 挿入するには、次のようにPostmanで <http://localhost:8080/odata.svc/Animal>
とプロパティ名と値のペアを持つJSON本体を指定してPOSTを送信します。
17. 永続クラスを使用して、すべてのCRUD操作を実行できます。
18. IRIS OData
Serverがコミュニティで採用されるようになれば、今後、その他多数の機能がリリースされるでしょう。

それでは！

ODataとInterSystems IRIS

Published on InterSystems Developer Community (<https://community.intersystems.com>)

[#コンテスト](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#) [#Open Exchange](#)
[InterSystems Open Exchangeで関連アプリケーションを確認してください](#)

ソースURL:<https://jp.community.intersystems.com/post/odata%E3%81%A8intersystems-iris>