OData: What’s New with REST APIs for Your Database

Sanjeev Mohan, Gartner
Nishanth Kadiyala, Progress
Mark Biamonte, OData TC Member, Progress
Audio Bridge Options & Question Submission
OData: What’s New with REST APIs for Your Database

Sanjeev Mohan, Gartner
Nishanth Kadiyala, Progress
Mark Biamonte, OData TC Member, Progress
Agenda

- Modern Data Access Patterns
- Why are companies RESTifying databases?
- OData overview
  - Exposing the Database Using OData
  - Getting Started with OData
  - Demo
Modern Data Access Patterns
Data Warehouses – still the workhorse analytical engines

Source: https://en.wikipedia.org/wiki/Data_warehouse
Data Access for Data Lakes

Analytical
SQL

Mobile / Cloud
{REST}
Common Data Lake Implementation Technologies Create Differing Requirements for Data Governance

**Hadoop distributions:**
- Simplified data ingestion and storage with several processing options
- Data lake management ecosystem emerging
- Complex deployment and management

**Cloud-based block and object stores:**
- Simplified data ingestion and storage
- Bring your own processing
- Nascent management and security ecosystem

**Database management systems:**
- Optimal for certain data types and formats
- Data processing options expanding beyond SQL
- Scaling and cost may be challenges
Virtual Data Lake by REST Enabling Sources
Goal is to “Connect” data and not “Collect”

- REST APIs are accessible over the internet and thus are much more user/coder friendly
- REST is stateless and thus can easily be horizontally scaled since there is no session dependency
- As you monetize data, APIs offer a secure way to expose your data to users outside your organization
- REST APIs are more human-readable and thus more engaging
- Decoupling the code from the infrastructure will give the enterprise the flexibility to adopt newer technologies
Why are companies RESTifying databases?
Transition from SOAP to REST
# SOAP vs REST

<table>
<thead>
<tr>
<th>SOAP</th>
<th>REST</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>JSON, XML, YAML etc.</td>
</tr>
<tr>
<td>Uses WSDL to define interface</td>
<td>Uses HTTP verbs (Get, Post, Put, Delete)</td>
</tr>
<tr>
<td>Stateful</td>
<td>Stateless</td>
</tr>
</tbody>
</table>
SOAP vs REST

Source: Google Trends
SOAP vs REST

Source: Google Trends
Use Cases for **REST** Enabling Databases
1. Expose Enterprise Data to SaaS/Cloud via HTTP

Enterprise REST APIs

Enterprise Data

HTTP / HTTPS

salesforce
Google Analytics
amazon web services
Google Cloud Platform
Microsoft Dynamics
NETSUITE
AZURE
2. Modernize Access to Enterprise Data

Your DB

REST APIs

{JSON}
2. Modernize Access to Enterprise Data
2. Modern Data Access
3. Additional Layer means Additional Control

- Limit direct access to your DB
- Throttling can help protect against resource monopolization
- Decoupling code from the database makes future transitions easier
Poll Question 1

Which of the following use cases are true for your organization?

- Your database needs to support modern languages such as JS, node.js, angular, python, etc.
- Sensitive data needs to stay within the firewall (GDPR, etc.)
- Enterprise Data needs to be available from new devices
- Modern Apps need REST end points for integration
- Other
Industry is moving towards REST APIs

<table>
<thead>
<tr>
<th>Cloud-Hosted Databases</th>
<th>On-Premises Databases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Cloud</td>
<td>Oracle REST Data Services</td>
</tr>
<tr>
<td>Azure</td>
<td>Teradata REST API</td>
</tr>
<tr>
<td>Database.com</td>
<td>IBM DB2 Rest Services</td>
</tr>
<tr>
<td>IBM Cloudant</td>
<td>MarkLogic REST API</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Poll Question 2

Which of the following databases are of interest to REST enable?

- Oracle
- SQL Server
- IBM DB2
- Postgres
- MySQL
Poll Question 2 – Part 2

Which of the following databases are of interest to REST enable?

- Amazon Redshift
- Hadoop Frameworks
- Teradata
- OpenEdge
- Other
But, every database and REST API is different… **OData** is the Answer
OData Overview
An **open protocol** to allow the creation and consumption of **queryable** and **interoperable RESTful APIs** in a **simple** and **standard** way

- Started by Microsoft in 2007
- Ratified as an ISO standard in Feb 2017
- OASIS Standard since Feb 2014

**Progress was the first member to join the OData Technical Committee**
OData is the standard for REST

OData is essentially SQL for the web built on top of standard protocols – HTTP, JSON & ATOM – while leveraging the REST architecture style.

- ODATA - The standard REST API
- JSON
- XML
- Mime
- HTTP(S)
Broad Adoption for OData
Exposing the Database Using OData
OData Data Model

Customers

Id: 2
Name: Mark
Status: active
Orders: Customers(1)/Orders

Id: 1
Name: Mark
Status: active
Orders: Customers(1)/Orders
## Mapping Databases to OData

<table>
<thead>
<tr>
<th>Database Construct</th>
<th>OData Construct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table</td>
<td>Entity Set</td>
</tr>
<tr>
<td>Row</td>
<td>Entity</td>
</tr>
<tr>
<td>Column</td>
<td>Property</td>
</tr>
<tr>
<td>Stored Procedure</td>
<td>Action (or Function)</td>
</tr>
<tr>
<td>User Defined Function</td>
<td>Function</td>
</tr>
<tr>
<td>Primary Key</td>
<td>Entity Key</td>
</tr>
<tr>
<td>Foreign Key</td>
<td>Navigation Property</td>
</tr>
</tbody>
</table>
# Mapping SQL to OData

<table>
<thead>
<tr>
<th>HTTP Method</th>
<th>OData REST Operation</th>
<th>SQL Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST</td>
<td>Create a new entity</td>
<td>INSERT</td>
</tr>
<tr>
<td>GET</td>
<td>Retrieve zero or more entities</td>
<td>SELECT</td>
</tr>
<tr>
<td>PUT</td>
<td>Update an Entity (replace semantics)</td>
<td>UPDATE</td>
</tr>
<tr>
<td>PATCH</td>
<td>Update an Entity (update semantics)</td>
<td>UPDATE</td>
</tr>
<tr>
<td>DELETE</td>
<td>Destroy an Entity</td>
<td>DELETE</td>
</tr>
<tr>
<td>GET</td>
<td>Invoke a function and get its results</td>
<td>CALL / EXEC</td>
</tr>
<tr>
<td>POST</td>
<td>Invoke and action and get its results</td>
<td>CALL</td>
</tr>
</tbody>
</table>
### Self Describing Service – Metadata Document

- Detailed description of the service

**URL: `<service_root>/$metadata`**

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex Type</td>
<td>Action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type Properties</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Property Attributes</td>
<td>Parameter Attributes</td>
</tr>
<tr>
<td>Name, Type, MaxLength, Precision, Scale, Nullable, more</td>
<td>Name, Type, MaxLength, Precision, Scale, Nullable, more</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Navigation Property</th>
<th>Return Type (Function)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key (EntityType)</td>
<td>Type, Nullable</td>
</tr>
</tbody>
</table>

- Information equivalent to SQL Tables, Columns, Primary Key, Foreign Key, Procedures, ProcedureColumns, Functions, FunctionColumns metadata calls
URL Query Conventions

https://myodataserver/northwind
|________________________/|
| Service Root |

https://myodataserver/northwind/Customers('ALFKI')/Orders?$expand=OrderDetails
|_________________________/
|_____________________/
|_________________/
| Service Root | Resource Path | Query Options |
## ODATA Query Options

<table>
<thead>
<tr>
<th>System Query Option</th>
<th>Description</th>
<th>SQL Construct</th>
</tr>
</thead>
<tbody>
<tr>
<td>$filter</td>
<td>Restrict the entities returned when querying an entity set to those matching the filter criteria</td>
<td>WHERE clause</td>
</tr>
<tr>
<td>$select</td>
<td>Specify the properties to be included in the returned entities</td>
<td>SELECT list</td>
</tr>
<tr>
<td>$orderby</td>
<td>Specify the sort order of the returned entities</td>
<td>ORDER BY clause</td>
</tr>
<tr>
<td>$expand</td>
<td>Include related entities and complex types nested in the returned entities</td>
<td>INNER JOIN</td>
</tr>
<tr>
<td>$top and $skip</td>
<td>Enable client to page through results</td>
<td>TOP/SKIP or LIMIT/OFFSET</td>
</tr>
<tr>
<td>$count</td>
<td>Include the count of the number of entities returned in the result</td>
<td>COUNT(*)</td>
</tr>
<tr>
<td>$search</td>
<td>Restrict the entities returned when querying an entity set to those matching the search expression</td>
<td>Full Text Search</td>
</tr>
<tr>
<td>$format</td>
<td>Specify the desired data format for the response</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Batch Insert and Transactions

URL: <service_root>/$batch

- Execute multiple operations in a single OData request.

- Operations in a Change Set are executed atomically
Getting Started with OData

<table>
<thead>
<tr>
<th>.NET</th>
<th>Java</th>
<th>JavaScript</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTier</td>
<td>Apache Olingo</td>
<td>Node-odata</td>
<td>ODataCpp (C++)</td>
</tr>
<tr>
<td>ODataLib</td>
<td>SDL OData Frameworks</td>
<td>DevExtreme</td>
<td>Pyslet Python Package</td>
</tr>
<tr>
<td>Edmlib</td>
<td>Odata4j</td>
<td>o.js</td>
<td>ODataStore for CoreData (iOS)</td>
</tr>
<tr>
<td>ASP.NET Web API OData</td>
<td>Jello Framework</td>
<td>OpenUI5</td>
<td>OData4ObjC (iOS)</td>
</tr>
<tr>
<td>AdaptiveLINQ</td>
<td>ODataJClient</td>
<td>JayData</td>
<td>OData Client Library for Tcl/Tk</td>
</tr>
<tr>
<td>Microsoft.Spatial</td>
<td></td>
<td>Breeze.js</td>
<td></td>
</tr>
</tbody>
</table>
Or use Hybrid Data Pipeline to create OData REST API from your enterprise databases

Why Hybrid Data Pipeline?

- OData from any data source
- No Coding
- No version control needed
- Patented on-premises gateway technology
Resources

- Understanding OData in 6 steps: [http://www.odata.org/getting-started/understand-odata-in-6-steps/](http://www.odata.org/getting-started/understand-odata-in-6-steps/)
- Getting Started Tutorial on OData.org: [http://www.odata.org/getting-started/basic-tutorial/](http://www.odata.org/getting-started/basic-tutorial/)
- Getting Started Tutorial with OData for your enterprise databases: [https://www.progress.com/tutorials/odata/rest-api-for-sql-server-oracle-or-postgres-via-odata](https://www.progress.com/tutorials/odata/rest-api-for-sql-server-oracle-or-postgres-via-odata)
- REST API Debate across OData, GraphQL, ORDS: [https://dzone.com/articles/rest-api-industry-debate-odata-vs-graphql-vs-ords](https://dzone.com/articles/rest-api-industry-debate-odata-vs-graphql-vs-ords)
- FAQs we hear on OData: [https://www.progress.com/blogs/odata-faqs-why-should-rest-api-developers-use-odata](https://www.progress.com/blogs/odata-faqs-why-should-rest-api-developers-use-odata)
Appendix
### Query Examples

Service Root: http://myodataserver/northwind

<table>
<thead>
<tr>
<th>Resource</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td><a href="http://myodataserver/northwind/Customer">http://myodataserver/northwind/Customer</a></td>
</tr>
<tr>
<td>$select=x,y</td>
<td><a href="http://myodataserver/northwind/Customer?select=CompanyName,CustomerID,Country">http://myodataserver/northwind/Customer?select=CompanyName,CustomerID,Country</a></td>
</tr>
<tr>
<td>$filter</td>
<td><a href="http://myodataserver/northwind/Customer?$filter=Country">http://myodataserver/northwind/Customer?$filter=Country</a> eq 'Germany'</td>
</tr>
<tr>
<td>$orderby</td>
<td><a href="http://myodataserver/northwind/Customer?$orderby=Country,City">http://myodataserver/northwind/Customer?$orderby=Country,City</a> desc</td>
</tr>
<tr>
<td>$expand</td>
<td><a href="http://myodataserver/northwind/Customer?$expand=Orders">http://myodataserver/northwind/Customer?$expand=Orders</a></td>
</tr>
<tr>
<td>$top and $skip</td>
<td><a href="http://myodataserver/northwind/Customer?$orderby=Country,City">http://myodataserver/northwind/Customer?$orderby=Country,City</a> desc&amp;$top=5&amp;$skip=5</td>
</tr>
<tr>
<td>$count</td>
<td><a href="http://myodataserver/northwind/Customer?$count=true">http://myodataserver/northwind/Customer?$count=true</a></td>
</tr>
</tbody>
</table>
Putting it altogether

```
http://myodataserver/northwind/Customer?
    $expand=Orders($select=OrderID,EmployeeID,OrderDate;$filter=EmployeeID eq 4)
    &$filter=Country eq 'Germany' and (City eq 'Berlin' or contains(City,'Frankfurt'))
    &$select=CustomerID,CompanyName,ContactName,City,Country,Phone
    &$orderby=CompanyName desc
    &$count=true
```