**OData Notes**

* The Open Data Protocol (OData) is a data access protocol built on core protocols like HTTP and commonly accepted methodologies like REST for the web.
  + World has agreed to REST
  + If you want to publish REST-based web services for accessing a structured data store like a relational database, you will end up inventing your own custom protocols on top of REST
  + OData is based on REST but adds necessary standards for a fully-fledged data APIs
* For e.g.
  + You have an API /customers which returns all customers in the DB
  + Then you have a request coming in stating, get the customers ordered by last name, age, marital status, etc.
  + You will get requests coming in as get customers who match the particular criteria….. and so on
  + You will end up creating complex and hundreds of APIs for customers
* OData takes the idea of SQL Query language to the web
* Protocol the client uses to access data locally from RDBMS: TDS (Tabular Data Stream) (JDBC is the API)
  + TDS
    - Rather old
    - Very well documented
    - Workhorse for decades for data access from DB
    - Works perfectly on reliable network connection (local DB)
    - Not good for access data from remote DB
    - It is SSL secured
    - However, the connection breaks as soon as the network connection breaks
    - Internet is not reliable
    - May be a security concern as well
    - Establishes an IP based connection. Hence if something happens to the network, even when the connection is not being used at that time, the connection is broken and you have to reconnect
  + OData
    - Relies on REST, which relies on HTTP – package based protocol - stateless
    - As long as the packages are able to reach the destination HTTP does not care if the connection is maintained.
    - Uses ports 80 and 443 – standard HTTP(S) ports – hence easy to inspect, run through firewalls
    - A data provider can support much larger number of concurrent clients since now there are no longer “active connections” required
    - Can support async queries as connections are not blocked
    - Supports XML, JSON
    - Not tied to any kind of database
    - Technology agnostic
* Need
  + A new protocol that supports all we love in query language, but runs on the HTTP protocols
  + “OData is a SQL for the Web”
    - Uses HTTP, JSON, XML,
  + Query language adopted to the needs of the web
* Any clients on any platform (as long as the client can speak HTTP and understand JSON/XML) it works. Does not have to understand complex TDS
* OData is now an OASIS standard
* Adoption of OData
  + All new Microsoft APIs, like Sharepoint, Azure, Office360 are all OData compliant
  + Azure active directory (on the cloud), Exchange server
  + <https://graphexplorer.cloudapp.net/default.aspx>
  + PowerBI,
  + SAP, Amazon are building OData compliant APIs
* Northwind Database
* OData service
  + Capable of providing metadata
  + Can use all kind of security schemes for access management
* Metadata
  + OData come metadata out-of-the-box
  + Metadata enables developers to write generic clients
  + Language is XML (data can be any format)
  + CSDL = Common Schema Definition Language
    - Used to define Metadata
    - You can build your own OData provider from scratch
* OData clients
  + Excel (2010 onwards)
    - Menu 🡪 Data 🡪 Get External Data 🡪 From Other Sources 🡪 From OData feed
  + LINQPad
    - Tool to write code in VB/C# and query from DB
    - Supports OData
* Advantages of OData
  + Usually consumers need to read documentation of your APIs to figure out which calls are possible
* Query Language functionality in OData
  + Implemented using keywords
  + Begins with $
  + Sent as a query parameter in the REST call
  + The incoming URI is translated into some kind of query tree called “Syntax tree” by the server. This “Syntax tree” is then translated into SQL and the filtering is not done on the App Server, but the RDBMS – uses grouping/aggregating on the server, uses indices, and all optimization features of RDBMS
  + Possible functionalities
    - Filters
      * Equals, not equals, greater than, etc.
      * Calculate results
      * Mathematical functions
      * Expand (joins)
      * Project (select)
      * Top and skip for paging
      * Count, search
* OData Functions
  + OData supports functions – same as stored procedure – user defined
  + An OData function can behind the scenes do whatever is necessary (custom code)
  + Allows you to use DB specific extension/functions to speed things
* By default – OData statements (simple where clauses, joins, etc.) works same for all different databases
* Creating an OData Service
  + Entity Framework
    - Object relational mapping framework for .Net applications
    - Helps define data model that are completely database agnostic

Why OData?

The challenge

* Many organizations are now opening up their data as RESTful APIs
* However, it is hard to create an application which can access data from multiple data sources as each organization provides data in a slightly different way
* Current solution – each organization provide developer documentation on how to manipulate their data sets. Needs specific code to access and manipulate those data
* Common data formats: XML, JSON, Atom
* Uniform semantics to query and manipulate the data have emerged: using REST CQRS
* For e.g.
  + You need different web browser for accessing different web pages
* OData unifies these semantics – built on REST/secure practices on web
* Can express data model via metadata

Advantages of exposing your data as OData compliant

* Compatible with a large number of existing applications designed for use with OData, e.g. PowerPivot, Excel,
* Every developer who knows how to handle OData endpoint now knows how to handle your data
* Consumers can be provided granular access for only the subset of data they need access to, reducing bandwidth
* Can leverage DBMS optimization features

Tutorials

<https://blogs.sap.com/2014/02/05/odata-producer-using-apache-olingo-with-mysql-jpa-tomcat-web-server/>

<http://download.oracle.com/oll/obe/EntityFrameworkWCF/WCFEntityFramework.htm>

<https://community.apigee.com/questions/2624/odata-version-supported.html>