OData Version 4.0 – Quick reference

Introduction 1

Data Model 1

Service Model 2

Context URL 2

Common Schema Definition Language (CSDL) 2

Topics 4

Resource Operations 5

OData V2 vs V4 5

# Introduction

OData is a protocol for the creation and consumption of RESTful APIs.

# Data Model

* **-Entity Data Model (EDM)-:** the abstract data model that is used to describe the data exposed by an OData service.
  + The central concepts in the EDM are entities, relationships, entity sets, actions, and functions.
* **-Entities-** are instances of entity types (e.g. Customer, Employee, etc.).
* **-Entity types-** are named structured types with a key. They define the named properties and relationships of an entity. Entity types may derive by single inheritance from other entity types.
  + declared properties
  + dynamic properties
  + navigation properties
* **-Complex types-** are keyless named structured types
* **-Enumeration types-** are named primitive types whose values are named constants with underlying integer values.
* -Type definitions- can be used in place of primitive typed properties, for example, within property definitions.
* **-Entity sets-** are named collections of entities (e.g. Customers is an entity set containing Customer entities)
* **-Operations-** allow the execution of custom logic on parts of a data model.
  + **Functions** are operations that do not have side effects
  + **Actions** are operations that allow side effects, such as data modification
    - **Bound** > bound to a type, enabling them to be called as members of an instance of that type
    - **Unbound** > called as static operations
* -**Singletons**- are single entities which are accessed as children of the entity container.
* **-OData resource-** is anything in the model that can be addressed (an entity set, entity, property, or operation).

# Service Model

The -**service document**- lists entity sets, functions, and singletons that can be retrieved.

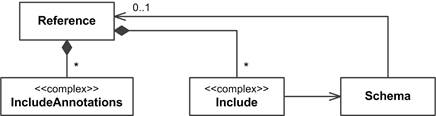
The -**metadata document**- describes the types, sets, functions and actions understood by the OData service.

# Context URL

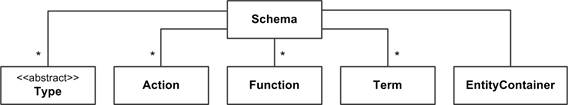
- context URL describes the content of the payload.

# Common Schema Definition Language (CSDL)

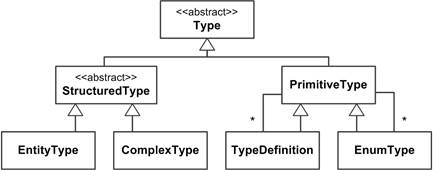
* Entity Model Wrapper



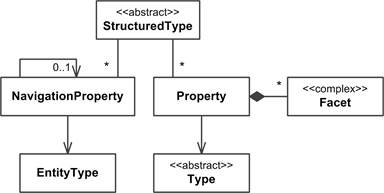
* **root** edmx:Edmx
* edmx:DataServices element MUST contain one or more edm:Schema
* edm:**Schema** 
  + acts as a namespace for elements of the entity model such as entity types, complex types, enumerations and terms.



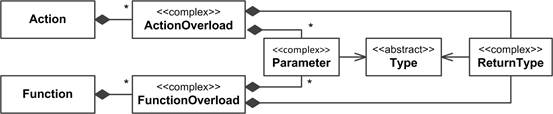
* Types



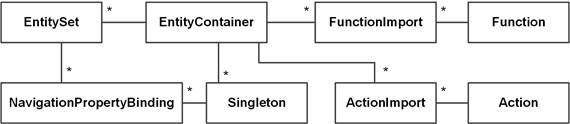
* Structured Types
  + Entity types and complex types
  + composed of zero or more -structural properties- and -navigation properties-



* Entity Type
  + template for an Entity
  + consists of one or more references to structural properties
    - **-structural property-** is a named reference to a primitive, complex, or enumeration type, or a collection of primitive, complex, or enumeration types
    - **-navigation property-** is a named reference to another entity type or collection of entity types.
* Complex Type
  + keyless nominal structured types
* Enumeration Type
  + nominal types that represent a series of related values
  + example
    - *<EnumType Name="FileAccess" UnderlyingType="Edm.Int32" IsFlags="true">*
      * *<Member Name="Read" Value="1" />*
      * *<Member Name="Write" Value="2" />*
      * *<Member Name="Create" Value="4" />*
      * *<Member Name="Delete" Value="8" />*
    - *</EnumType>*
* Action and Function
  + Todo



* Entity Container
  + an OData service **MUST** define exactly one entity container
  + define the entity sets, singletons, function and action imports exposed by the service.
  + **-entity set-** allows access to entity type instances



* Metadata Service Schema

# Topics

* Requesting Data
  + Requesting Entity Collections
  + Requesting an Individual Entity by ID
  + Requesting an Individual Property
  + Requesting an Individual Property Raw Value
* Querying Data
  + System Query Option $filter
  + System Query Option $orderby
  + System Query Option $top and $skip
  + System Query Option $count
  + System Query Option $expand
  + System Query Option $select
  + System Query Option $search
  + Lambda Operators
* Data Modification
  + Create an Entity
  + Delete an Entity
  + Update an Entity
  + Relationship Operations
  + Functions and Actions
  + ETag

# Resource Operations

**GET**: Get the resource (a collection of entities, a single entity, a structural property, a navigation property, a stream, etc.).

**POST**: Create a new resource.

**PUT**: Update an existing resource by replacing it with a complete instance.

**PATCH**: Update an existing resource by replacing part of its properties with a partial instance.

**DELETE**: Remove the resource.

# OData V2 vs V4

|  |  |
| --- | --- |
| V2 | V4 |
| [Microsoft Open Specification Promise](https://en.wikipedia.org/wiki/Microsoft_Open_Specification_Promise) | [OASIS](https://en.wikipedia.org/wiki/OASIS_(organization)) |
| JSON, Atom Pub | JSON as standard |
|  | actions, functions, collection values, navigation properties on derived types, stream properties |
|  | $search |
|  | <http://docs.oasis-open.org/odata/new-in-odata/v4.0/cn01/new-in-odata-v4.0-cn01.html> |