

# Supporting OData Query Syntax

Brian Noyes

CTO, Solliance ([www.solliance.net](http://www.solliance.net))

@briannoyes



# Outline

- OData query syntax overview
- Adding OData Queryable support to Web APIs
- Deferred execution
- Query validation

# **OData Query Syntax**

- **Encode complex query instructions in the request URL**
- **Embed one or more operators to instruct the service to shape the results**
- **Form complex expressions as parameters to the operators**
- **Express relations to identify portions or relations to a resource that you want to return**

# OData Query Operators

- **Web API OData first release support (mid-Feb 2013)**
  - \$filter – ask for a subset of the resource collection based on an expression
  - \$orderby – order the results on the server based on one or more of the properties of the returned entities
  - \$top – only return a certain number of entities that result from the query
  - \$skip – skip a certain number of entities from the beginning of the entity set that would have been returned
  - \$inlinecount – returns a server computed count of the number of entities that would have been produced by the associated query parameters in the request
  - \$it – like a “this” reference for a resulting object in the query

# OData Query Operators

- **Currently missing in Web API OData:**
  - \$expand – include related entities in the returned payload, instead of just providing links to them
  - \$select – return just a specific set of properties from the entities in a result set instead of the entire entity
  - Batch updates
  - Media streams

# OData Expressions

- **Complex URL syntax for formulating values passed to the query operators**
  - Boolean comparisons ( =, <, >, or, and, etc.)
  - String parsing (length, substring, indexof, etc.)
  - Numeric operations (add, sub, mult, div, round, ceiling, floor, etc.)
  - Date functions (year, month, day, minutes, seconds, etc.)
  - Type operations (IsOf, Cast)
  - Grouping – parens to group and specify eval priority

# Query samples

- **/Products**
  - All products
- **/Products?\$filter=ProductName eq 'Tofu'**
  - Products with product name Tofu
- **/Products?\$filter=(UnitPrice gt 10.0m) and (UnitPrice lt 25.0m)**
  - Products with unit price between 10 and 25
- **/Products?\$filter=substringof('Crab',ProductName) eq true**
  - Products with Crab in their name
- **/Products?\$orderby=ProductName**
  - Products ordered by Product name (ascending by default)
- **/Products?\$top=2**
  - First 2 products
- **/Products?\$top=2&\$orderby=ProductName**
  - First two products ordered by ProductName
- **/Products?\$top=2&\$skip=2&\$orderby=ProductName**
  - Second page of products, 2 per page, ordered by ProductName
- **/Products?\$filter=(UnitPrice gt 10.0m) and (UnitPrice lt 25.0m)&\$top=2&\$skip=2&\$orderby=ProductName**
  - Products with unit price between 10 and 25, paged with 2 per page, second page, ordered by ProductName

# Relationship Query Samples

- **/Customers('ALFKI')**
  - Get the customer who's key is ALFKI
- **/Customers('ALFKI')/Orders**
  - Get the orders for customer ALFKI
- **/Customers('ALFKI')/Orders(10643)**
  - Get a specific order (key 10643) related to customer ALFKI
- **/Customers('ALFKI')/Orders(10643)/Order\_Details**
  - Get the order details for a particular order of customer ALFKI
- **/Categories(1)/\$links/Products**
  - Get just the hypermedia links to the products within category 1



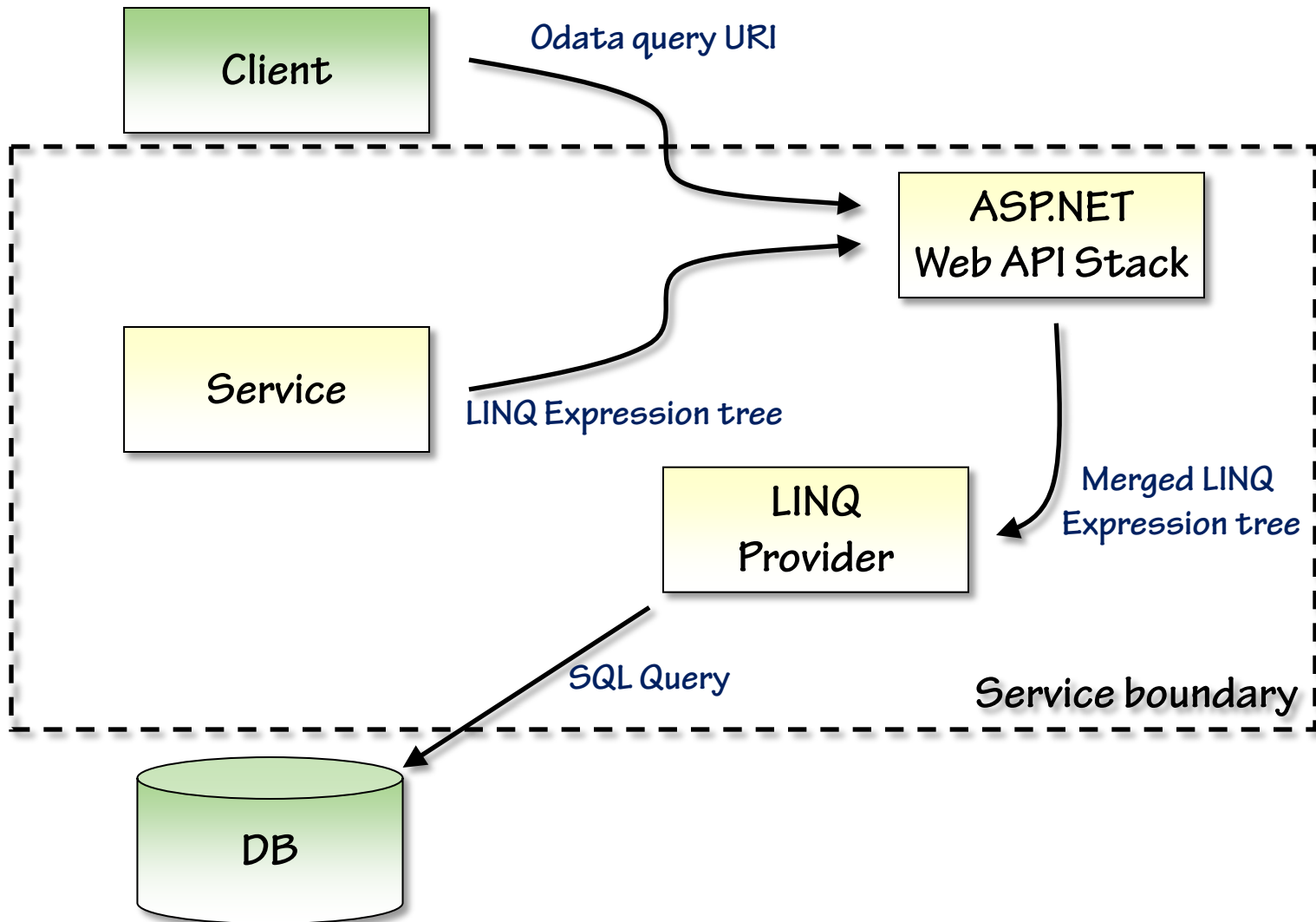
# Adding OData Queryable Support to Web API

- **Normal Web API CRUD actions (methods) return collections or single entities**
- **Can return an IQueryable instead**
  - Interface abstraction for an expression tree
  - Represents the “potential” for a result set, but does not produce a collection until executed
- **Out of the box Web API will just iterate over the IQueryable and produce the entire collection of what was returned by the method**
- **Add [Queryable] to the method**
  - Part of the OData NuGet package for Web API
- **Causes the service to support OData query syntax in the request URL**
- **Web API will merge the IQueryable returned by the service method with one produced from the client query options in the URL**

# Deferred Execution

- **An IQueryable represents an expression tree**
  - A series of operations that will run against a collection, producing new collections
  - Does not produce the collection items until asked to (i.e. iteration)
- **LINQ Providers implement IQueryable for their exposed collections**
  - Produce the results however it makes sense for them
- **LINQ providers**
  - Entity Framework (LINQ to Entities)
  - LINQ to SQL
  - LINQ to XML
  - LINQ to Objects
- **Entity Framework and LINQ to SQL produce results by formulating a SQL query based on the expression tree**

# Deferred Execution



# Query Validation

- Callers should not always be able to execute any query they like
  - Constrain maximum number of rows
  - Constrain ordering of a non-indexed column
  - Constrain what operations are supported on the underlying LINQ provider
- Can add declarative conditions to your [Queryable] attribute

GET /api/Customers

[Queryable()]

▲ 2 of 2 ▼ QueryableAttribute.QueryableAttribute(Named Parameters...)

Enables a controller action to support OData query parameters.

**Named Parameters:**

- AllowedArithmeticOperators* = *System.Web.Http.OData.Query.AllowedArithmeticOperators*
- AllowedFunctions* = *System.Web.Http.OData.Query.AllowedFunctions*
- AllowedLogicalOperators* = *System.Web.Http.OData.Query.AllowedLogicalOperators*
- AllowedOrderByProperties* = *string*
- AllowedQueryOptions* = *System.Web.Http.OData.Query.AllowedQueryOptions*
- EnableConstantParameterization* = *bool*
- EnsureStableOrdering* = *bool*
- HandleNullPropagation* = *System.Web.Http.OData.Query.HandleNullPropagationOption*
- MaxAnyAllExpressionDepth* = *int*
- MaxNodeCount* = *int*
- MaxSkip* = *int*
- MaxTop* = *int*
- PageSize* = *int*

# Summary

- OData query syntax defines a complex query language that can be embedded in a URL to express the client query desires for service execution
- OData operators instruct the service how to shape the results returned to the client
- OData expressions support complex parameters to the operators
- Adding Queryable support to Web APIs is easy and unleashes the power of OData queries for any data retrieval Web API
- With a LINQ provider under your Web API, supporting OData queries means you can leverage deferred execution and expression tree merging automatically