

OData Query Syntax

Query String Options

The Query Options section of an OData URI specifies three types of information: [System Query Options](#), [Custom Query Options](#), and [Service Operation Parameters](#). All OData services must follow the query string parsing and construction rules defined in this section and its subsections.

4.1. System Query Options

System Query Options are query string parameters a client may specify to control the amount and order of the data that an OData service returns for the resource identified by the URI. The names of all System Query Options are prefixed with a "\$" character.

An OData service may support some or all of the System Query Options defined. If a data service does not support a System Query Option, it must reject any requests which contain the unsupported option as defined by the request processing rules in [\[OData:Operations\]](#).

4.2. Orderby System Query Option (\$orderby)

A data service URI with a \$orderby System Query Option specifies an expression for determining what values are used to order the collection of Entries identified by the Resource Path section of the URI. This query option is only supported when the resource path identifies a Collection of Entries.

The \$orderby section of the normative OData specification outlines the full expression syntax supported by this query option. The examples below represent the most commonly supported subset of that expression syntax.

Examples

[https://services.odata.org/OData/OData.svc/Products?\\$orderby=Rating](https://services.odata.org/OData/OData.svc/Products?$orderby=Rating)

- All Product Entries returned in ascending order when sorted by the Rating Property.

[https://services.odata.org/OData/OData.svc/Products?\\$orderby=Rating asc](https://services.odata.org/OData/OData.svc/Products?$orderby=Rating asc)

- Same as the example above.

[https://services.odata.org/OData/OData.svc/Products?\\$orderby=Rating,Category/Name desc](https://services.odata.org/OData/OData.svc/Products?$orderby=Rating,Category/Name desc)

- Same as the URI above except the set of Products is subsequently sorted (in descending order) by the Name property of the related Category Entry.

4.3. Top System Query Option (\$top)

A data service URI with a \$top System Query Option identifies a subset of the Entries in the Collection of Entries identified by the Resource Path section of the URI. This subset is formed by selecting only the first N items of the set, where N is an integer greater than or equal to zero specified by this query option. If a value less than zero is specified, the URI should be considered malformed.

If the data service URI contains a \$top query option, but does not contain a \$orderby option, then the Entries in the set needs to first be fully ordered by the data service. While no ordering semantics are mandated, to ensure repeatable results, a data service must always use the same semantics to obtain a full ordering across requests.

Examples

[https://services.odata.org/OData/OData.svc/Products?\\$top=5](https://services.odata.org/OData/OData.svc/Products?$top=5)

- The first 5 Product Entries returned where the Collection of Products are sorted using a scheme determined by the OData service.

[https://services.odata.org/OData/OData.svc/Products?\\$top=5&\\$orderby=Name desc](https://services.odata.org/OData/OData.svc/Products?$top=5&$orderby=Name desc)

- The first 5 Product Entries returned in descending order when sorted by the Name property.

4.4. Skip System Query Option (\$skip)

A data service URI with a \$skip System Query Option identifies a subset of the Entries in the Collection of Entries identified by the Resource Path section of the URI. That subset is defined by seeking N Entries into the Collection and selecting only the remaining Entries (starting with Entry N+1). N is an integer greater than or equal to zero specified by this query option. If a value less than zero is specified, the URI should be considered malformed.

If the data service URI contains a \$skip query option, but does not contain a \$orderby option, then the Entries in the Collection must first be fully ordered by the data service. While no ordering semantics are mandated, to ensure repeatable results a data service must always use the same semantics to obtain a full ordering across requests.

Examples

[https://services.odata.org/OData/OData.svc/Categories\(1\)/Products?\\$skip=2](https://services.odata.org/OData/OData.svc/Categories(1)/Products?$skip=2)

- The set of Product Entries (associated with the Category Entry identified by key value 1) starting with the third product.

[https://services.odata.org/OData/OData.svc/Products?\\$skip=2&\\$top=2&\\$orderby=Rating](https://services.odata.org/OData/OData.svc/Products?$skip=2&$top=2&$orderby=Rating)

- The third and fourth Product Entry from the collection of all products when the collection is sorted by Rating (ascending).

4.5. Filter System Query Option (\$filter)

A URI with a \$filter System Query Option identifies a subset of the Entries from the Collection of Entries identified by the **Resource Path** section of the URI. The subset is determined by selecting only the Entries that satisfy the predicate expression specified by the query option.

The expression language that is used in \$filter operators supports references to properties and literals. The literal values can be strings enclosed in single quotes, numbers and boolean values (true or false) or any of the additional literal representations shown in the **Abstract Type System** section.

Note: The \$filter section of the normative OData specification provides an ABNF grammar for the expression language supported by this query option.

The operators supported in the expression language are shown in the following table.

| Operator | Description | Example |
|----------------------|-----------------------|---|
| Logical Operators | | |
| Eq | Equal | /Suppliers?\$filter=Address/City eq 'Redmond' |
| Ne | Not equal | /Suppliers?\$filter=Address/City ne 'London' |
| Gt | Greater than | /Products?\$filter=Price gt 20 |
| Ge | Greater than or equal | /Products?\$filter=Price ge 10 |
| Lt | Less than | /Products?\$filter=Price lt 20 |
| Le | Less than or equal | /Products?\$filter=Price le 100 |
| And | Logical and | /Products?\$filter=Price le 200 and Price gt 3.5 |
| Or | Logical or | /Products?\$filter=Price le 3.5 or Price gt 200 |
| Not | Logical negation | /Products?\$filter=not endswith(Description,'milk') |
| Arithmetic Operators | | |
| Add | Addition | /Products?\$filter=Price add 5 gt 10 |
| Sub | Subtraction | /Products?\$filter=Price sub 5 gt 10 |
| Mul | Multiplication | /Products?\$filter=Price mul 2 gt 2000 |
| Div | Division | /Products?\$filter=Price div 2 gt 4 |
| Mod | Modulo | /Products?\$filter=Price mod 2 eq 0 |
| Grouping Operators | | |
| () | Precedence grouping | /Products?\$filter=(Price sub 5) gt 10 |

In addition to operators, a set of functions are also defined for use with the filter query string operator. The following table lists the available functions. Note: ISNULL or COALESCE operators are not defined. Instead, there is a null literal which can be used in comparisons.

| Function | Example |
|---|---|
| String Functions | |
| bool substringof(string p0, string p1) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq true |
| bool endswith(string p0, string p1) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq true |
| bool startswith(string p0, string p1) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq true |
| int length(string p0) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq 19 |
| int indexof(string p0, string p1) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq 1 |
| string replace(string p0, string find, string replace) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq 'AlfredsFutterkiste' |
| string substring(string p0, int pos) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq 'Alfreds Futterkiste' |
| string substring(string p0, int pos, int length) | https://services.odata.org/Northwind/Northwind.svc/Customers(CompanyName) eq 'lf' |

| | |
|--|--|
| string tolower(string p0) | https://services.odata.org/Northwind/Northwind.svc/Customers?eq 'alfreds futterkiste' |
| string toupper(string p0) | https://services.odata.org/Northwind/Northwind.svc/Customers?eq 'ALFREDS FUTTERKISTE' |
| string trim(string p0) | https://services.odata.org/Northwind/Northwind.svc/Customers?eq 'Alfreds Futterkiste' |
| string concat(string p0, string p1) | https://services.odata.org/Northwind/Northwind.svc/Customers', '), Country) eq 'Berlin, Germany' |
| Date Functions | |
| int day(DateTime p0) | https://services.odata.org/Northwind/Northwind.svc/Employees?eq 8 |
| int hour(DateTime p0) | https://services.odata.org/Northwind/Northwind.svc/Employees?eq 0 |
| int minute(DateTime p0) | https://services.odata.org/Northwind/Northwind.svc/Employees?eq 0 |
| int month(DateTime p0) | https://services.odata.org/Northwind/Northwind.svc/Employees?eq 12 |
| int second(DateTime p0) | https://services.odata.org/Northwind/Northwind.svc/Employees?eq 0 |
| int year(DateTime p0) | https://services.odata.org/Northwind/Northwind.svc/Employees?eq 1948 |
| Math Functions | |
| double round(double p0) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi?eq 32d |
| decimal round(decimal p0) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi?eq 32 |
| double floor(double p0) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi?eq 32d |
| decimal floor(decimal p0) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi?eq 32 |
| double ceiling(double p0) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi?eq 33d |
| decimal ceiling(decimal p0) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi?eq 33 |
| Type Functions | |
| bool IsOf(type p0) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi |
| bool IsOf(expression p0, type p1) | https://services.odata.org/Northwind/Northwind.svc/Orders?\$fi'Edm.String') |

4.6. Expand System Query Option (\$expand)

A URI with a \$expand System Query Option indicates that Entries associated with the Entry or Collection of Entries identified by the Resource Path section of the URI must be represented inline (i.e. eagerly loaded). For

example, if you want to identify a category and its products, you could use two URIs (and execute two requests), one for /Categories(1) and one for /Categories(1)/Products. The '\$expand' option allows you to identify related Entries with a single URI such that a graph of Entries could be retrieved with a single HTTP request.

The syntax of a \$expand query option is a comma-separated list of Navigation Properties. Additionally each Navigation Property can be followed by a forward slash and another Navigation Property to enable identifying a multi-level relationship.

Note: The \$filter section of the normative OData specification provides an ABNF grammar for the expression language supported by this query option.

Examples

[https://services.odata.org/OData/OData.svc/Categories?\\$expand=Products](https://services.odata.org/OData/OData.svc/Categories?$expand=Products)

- Identifies the Collection of Categories as well as each of the Products associated with each Category.
- Is described by the Entity Set named "Categories" and the "Products" Navigation Property on the "Category" Entity Type in the service metadata document.

[https://services.odata.org/OData/OData.svc/Categories?\\$expand=Products/Suppliers](https://services.odata.org/OData/OData.svc/Categories?$expand=Products/Suppliers)

- Identifies the Collection of Categories as well as each of the Products associated with each Category. In addition, the URI also identifies the Suppliers associated with each Product.
- Is described by the Entity Set named "Categories", the "Products" Navigation Property on the "Category" Entity Type, and the "Suppliers" Navigation Property on the "Product" Entity Type in the service metadata document.

[https://services.odata.org/OData/OData.svc/Products?\\$expand=Category,Suppliers](https://services.odata.org/OData/OData.svc/Products?$expand=Category,Suppliers)

- Identifies the set of Products as well as the category and suppliers associated with each product.
- Is described by the Entity Set named "Products" as well as the "Category" and "Suppliers" Navigation Property on the "Product" Entity Type in the service metadata document.

4.7. Format System Query Option (\$format)

A URI with a \$format System Query Option specifies that a response to the request MUST use the media type specified by the query option. If the \$format query option is present in a request URI it takes precedence over the value(s) specified in the Accept request header. Valid values for the \$format query string option are listed in the following table.

| \$format Value | Response Media Type |
|---|-------------------------------|
| Atom | application/atom+xml |
| Xml | application/xml |
| Json | application/json |
| Any other IANA-defined content type | Any IANA-defined content type |
| A service-specific value indicating a format specific to the specific OData service | Any IANA-defined content type |

Examples

[https://services.odata.org/OData/OData.svc/Products?\\$format=atom](https://services.odata.org/OData/OData.svc/Products?$format=atom)

- Identifies all Product Entries represented using the AtomPub format as defined in [OData:Atom]

[https://services.odata.org/OData/OData.svc/Products?\\$format=json](https://services.odata.org/OData/OData.svc/Products?$format=json)

- Identifies all Product Entries represented using the JSON format as defined in [OData:JSON]

4.8. Select System Query Option (\$select)

A data service URI with a \$select System Query Option identifies the same set of Entries as a URI without a \$select query option; however, the value of \$select specifies that a response from an OData service should return a subset of the Properties which would have been returned had the URI not included a \$select query option.

Version Note: This query option is only supported in OData version 2.0 and above.

The value of a \$select System Query Option is a comma-separated list of selection clauses. Each selection clause may be a Property name, Navigation Property name, or the "*" character. The following set of examples uses the data sample data model available at [https://services.odata.org/OData/OData.svc/\\$metadata](https://services.odata.org/OData/OData.svc/$metadata) to describe the semantics for a base set of URIs using the \$select system query option. From these base cases, the semantics of longer URIs are defined by composing the rules below.

Examples

[https://services.odata.org/OData/OData.svc/Products?\\$select=Price,Name](https://services.odata.org/OData/OData.svc/Products?$select=Price,Name)

- In a response from an OData service, only the Price and Name Property values are returned for each Product Entry within the Collection of products identified.
- If the \$select query option had listed a Property that identified a Complex Type, then all Properties defined on the Complex Type must be returned.

[https://services.odata.org/OData/OData.svc/Products?\\$select=Name,Category](https://services.odata.org/OData/OData.svc/Products?$select=Name,Category)

- In a response from an OData service only the Name Property value and a link to the related Category Entry should be returned for each product.

[https://services.odata.org/OData/OData.svc/Categories?\\$select=Name,Products&\\$expand=Products/Suppliers](https://services.odata.org/OData/OData.svc/Categories?$select=Name,Products&$expand=Products/Suppliers)

- In a response from an OData service, only the Name of the Category Entries should be returned, but all the properties of the Entries identified by the Products and Suppliers Navigation Properties should be returned.

[https://services.odata.org/OData/OData.svc/Products?\\$select=*](https://services.odata.org/OData/OData.svc/Products?$select=*)

- In a response from an OData service, all Properties are returned for each Product Entry within the Products Entity Set.
- Note: The star syntax is used to reference all properties of the Entry or Collection of Entries identified by the path of the URI or all properties of a Navigation Property. In other words, the "*" syntax causes all Properties on an Entry to be included without traversing associations.

[https://services.odata.org/OData/OData.svc/Categories?\\$select=Name,Products&\\$expand=Products](https://services.odata.org/OData/OData.svc/Categories?$select=Name,Products&$expand=Products)

- In a response from an OData service, the Name property is included and Product Entries with all Properties are included; however, rather than including the fully expanded Supplier Entries referenced in the expand clause, each Product will contain a link that references the corresponding Collection of Supplier Entries.

Note: The \$select section of the [normative OData specification](#) provides an ABNF grammar for the expression language supported by this query option.

4.9. Inlinecount System Query Option (\$inlinecount)

A URI with a \$inlinecount System Query Option specifies that the response to the request includes a count of the number of Entries in the Collection of Entries identified by the [Resource Path](#) section of the URI. The count must

be calculated after applying any **\$filter System Query Options** present in the URI. The set of valid values for the \$inlinecount query option are shown in the table below. If a value other than one shown in Table 4 is specified the URI is considered malformed.

Version Note: This query option is only supported in OData version 2.0 and above

| \$inlinecount value | Description |
|----------------------------|--|
| allpages | The OData MUST include a count of the number of entities in the collection identified by the URI (after applying any \$filter System Query Options present on the URI) |
| none | The OData service MUST NOT include a count in the response. This is equivalence to a URI that does not include a \$inlinecount query string parameter. |

Examples

[https://services.odata.org/OData/OData.svc/Products?\\$inlinecount=allpages](https://services.odata.org/OData/OData.svc/Products?$inlinecount=allpages)

- Identifies all Product Entries and the count of all products.

[https://services.odata.org/OData/OData.svc/Products?\\$inlinecount=allpages&\\$top=10&\\$filter=Price gt 200](https://services.odata.org/OData/OData.svc/Products?$inlinecount=allpages&$top=10&$filter=Price gt 200)

- Identifies the first 10 Product Entries that cost more than 200 and includes a count of the total number of Product Entries that cost more than 200.

5. Custom Query Options

Custom Query Options provide an extension point for OData service-specific information to be placed in the query string portion of a URI. A Custom Query String option is defined as any name/value pair query string parameter where the name of the parameter does not begin with the "\$" character. Any URI exposed by an OData service may include one or more Custom Query Options.

Examples

<https://services.odata.org/OData/OData.svc/Products?x=y>

- Identifies all Product entities. Includes a Custom Query Option "x" whose meaning is service specific.

6. Service Operation Parameters

Service Operations represent functions exposed by an OData service. These functions may accept zero or more primitive type parameters. If a Service Operation requires an input parameter those parameters are passed via query string name/value pairs appended to the URI which identify the Service Operation as described in the **Addressing Service Operations** section. For nullable type parameters, a null value may be specified by not including the parameter in the query string of the URI.

Examples

<https://services.odata.org/OData/OData.svc/GetProductsByRating?rating=5>

- Identifies the "GetProductsByRating" Service Operation and specifies a value of 5 for the "rating" input parameter.

Revision #1

Created Mon, Apr 26, 2021 2:26 PM by [Nelson](#)

Updated Mon, Apr 26, 2021 2:26 PM by [Nelson](#)