

[All Collections](#)[Developer & API](#)[Getting Started with the OData API](#)

Getting Started with the OData API

For developers who want to learn how to write code and queries using the OData Restful API.



Written by Andrew Ardron

Updated over a week ago

This guide is a getting started document, starting with an introduction to OData and goes onto explain how to get a token and gives examples of the many ways to query OData. If you already know that stuff, and are just looking for details of the endpoints and the OData data models, then please refer to the [API documentation](#).

What is OData?

The Technical description:

A REST-based protocol for querying and updating data built on standard technologies such as HTTP, Atom/XML and JSON.

OData is standardised by OASIS and is ISO/IEC approved.

The Useful Description:

OData is a way to query databases over the web in a uniform way, requesting the data in a variety of formats (mainly JSON and XML).

Because the OData specification is standardised and ISO/IEC approved, we can theoretically use any OData feed, and it'll respond in the same way to the defined OData functions.

Authenticating

Some OData services are open (available to anyone without authentication). You'll need a Personal Access Token (PAT) to access our OData services. Visit [this article](#) for information on generating a PAT Token.

Constructing a Query

Queries are constructed from 3 parts:

1. The base URL
2. The entity you're accessing
3. The query parameters

A complete query will look something like this:

<https://api-batch-v1.prospect365.com/Contacts>

?\$select=Title,Forename,Surname,Email,StatusFlag

&\$filter=contains(email,'@prospectsoft.com') and StatusFlag eq 'A'

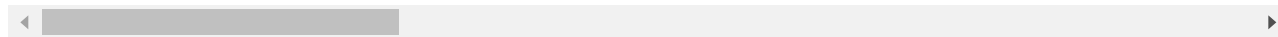
&\$expand=Leads(

\$select=Description,Created,StatusFlag;

```
$filter=StatusFlag eq 'A' and Status/DeadFlag eq 0;  
$expand=Status(  
    $select=Description  
)  
)
```

If you want to try that query later on, when you're connected and authenticated, you'll need to copy and paste it without the white space. Here's the same code as above, but this time without the formatting:

```
https://api-batch-v1.prospect365.com/Contacts?$select=Title,Forename,Surname,Email,Stat
```



Base URL

The base URL explains where you want to get your data from. Some examples are:

- The OData specification test service - <http://services.odata.org/V4/OData/OData.svc/>
- A fictitious company feed - <http://services.odata.org/Northwind/Northwind.svc/>

The services.odata example URLs are unsecured, so you can use these to test queries without needing to worry about authentication.

The Prospect OData service is secured with **bearer authentication**, which means you need to add a bearer authorisation header to your request, using a valid access token. If you don't have a token, you can create one [here](#). Once you have a token, you can access the Prospect base URL:

- The ProspectSoft OData service - <https://api-v1.prospect365.com> (requires authentication)

Important Note on Security

These access tokens allow reading and writing to the database as you, with your user ID. It allows the same access to the data as you have when you log in. You should treat this token as if it's your password. **Don't hand out your token or share it with anyone.** If a colleague, another developer or someone else needs a token, they should generate one themselves - with their own identity and permissions.

If you ever need to disable a token, you can revoke them from [here](#).

Entity

The entity is the type of CRM record you want to access. These are defined in our database and generally mirror our tables.

At the time of writing this, we don't have a full list of entities available on our service, but for the most part they are the plural of the table name.

This entity name is appended after the base URL like so:

- <https://api-batch-v1.prospect365.com/Contacts>
- <http://services.odata.org/V4/OData/OData.svc/Products>
- <http://services.odata.org/Northwind/Northwind.svc/Customers>

Please note: Running these queries without filtering will retrieve all data for those entities. If you try to retrieve 50,000 Contacts, it'll take a long time to complete.

Wherever possible, you should aim to only retrieve the data you need. Smaller chunks of data, over a larger number of queries, is almost always better.

To retrieve a specific record by ID, you can append it to the entity name in parentheses:

- [https://api-batch-v1.prospect365.com/Contacts\(272970\)](https://api-batch-v1.prospect365.com/Contacts(272970))
- [http://services.odata.org/V4/OData/OData.svc/Products\(1\)](http://services.odata.org/V4/OData/OData.svc/Products(1))
- [http://services.odata.org/Northwind/Northwind.svc/Customers\('ALFKI'\)](http://services.odata.org/Northwind/Northwind.svc/Customers('ALFKI'))

Parameters

Parameters are made up of the various OData functions for selecting and filtering entities.

The main functions are:

- \$Select
- \$Expand
- \$Filter
- \$OrderBy
- \$Top
- \$Skip

These parameters are chained together after the entity using '&':

- Get the top 50 Contacts showing ContactId, Forename and Surname - [https://api-batch-v1.prospect365.com/Contacts?\\$select=ContactId,Forename,Surname&\\$top=50](https://api-batch-v1.prospect365.com/Contacts?$select=ContactId,Forename,Surname&$top=50)
- Get the product with ID 1, showing Name, Description and Rating - [http://services.odata.org/V4/OData/OData.svc/Products\(1\)/?](http://services.odata.org/V4/OData/OData.svc/Products(1)/?)

[\\$select=Name,Description,Rating](#)

- Get the country and company name of all customers where country is 'UK' - [http://services.odata.org/Northwind/Northwind.svc/Customers/?\\$select=CompanyName,Country&\\$filter=Country_eq_'UK'](http://services.odata.org/Northwind/Northwind.svc/Customers/?$select=CompanyName,Country&$filter=Country_eq_'UK')

Please note: For most services, the entity has a '/' before the parameters. On our service we opted to omit this.

OData Functions

OData functions are used in the parameters section to manipulate the data before it's returned to the caller. These functions can be used alone, or chained together to make more complex queries.

\$Select

This allows you to reduce the data returned by only picking the fields you want to see. This is synonymous to SQL's select, where the OData default is all (select * from...).

The select looks like this: \$Select=[Field 1],[Field 2]...where Field 1, Field 2 and so on are replaced with field names.

Each field you want to select is separated by a comma:

- Select the surname of every Contact - [https://api-batch-v1.prospect365.com/Contacts?\\$select=Surname](https://api-batch-v1.prospect365.com/Contacts?$select=Surname)
- Select the Name, Description and Rating of all Products - [http://services.odata.org/V4/OData/OData.svc/Products/?\\$select=Name,Description,Rating](http://services.odata.org/V4/OData/OData.svc/Products/?$select=Name,Description,Rating)

- Select the CompanyName and Country of all Customers - [http://services.odata.org/Northwind/Northwind.svc/Customers/?\\$select=CompanyName,Country](http://services.odata.org/Northwind/Northwind.svc/Customers/?$select=CompanyName,Country)

\$Expand

Expand allows you to move from the base entity out to other levels of the hierarchy.

The syntax for expand is: \$Expand=[Navigation Property]

Navigation properties are the links between entities, and therefore tend to match foreign keys.

These links can be used in both directions, E.g. Contact to Division is Contact.Division. Division to Contact is Division.Contacts.

Compared to SQL, these are most similar to the Join syntax.

- Select all contacts and their Division - [https://api-batch-v1.prospect365.com/Contacts?\\$expand=Division](https://api-batch-v1.prospect365.com/Contacts?$expand=Division)
- Select all Products and their category - [http://services.odata.org/V4/OData/OData.svc/Products/?\\$expand=Categories](http://services.odata.org/V4/OData/OData.svc/Products/?$expand=Categories)
- Select all customers and their orders - [http://services.odata.org/Northwind/Northwind.svc/Customers/?\\$expand=Orders](http://services.odata.org/Northwind/Northwind.svc/Customers/?$expand=Orders)

Expand is the most complex function, as it allows you to chain other functions inside of the expand e.g. selecting Contact forename and Division name:

```
https://api-batch-v1.prospect365.com/Contacts?$select=Surname&$expand=Division($select=
```



Once inside an expand, you can chain together functions by using ';'. For example, to select every field on Contact, the Division name and everything from Company:

```
https://api-batch-v1.prospect365.com/Contacts?$expand=Division($select=Name;$expand=Comp
```



\$Filter

Filters allow you to reduce the data set by specifying a field and a requirement. There are 2 formats for filter:

- \$Filter=[Field Name] [Operator] [Value]
- \$Filter=[Operator]([Field],[Value])

The correct format to use depends on the operator you are using.

Operators for first format:

- eq - Exactly equal to
- ne - Inverse of eq
- lt - Less than but not including
- gt - Greater than but not including
- le - Less than or exactly equal to
- ge - Greater than or exactly equal to

Operators for second format:

- contains - Value is contained somewhere in the field
- not contains - Inverse of contains
- startswith - Field starts with value
- not startswith - Inverse of startswith

- endswith - Field ends with value
- not endswith - Inverse of endswith

There are other more specific expressions which you can find at the [Microsoft Documentation](#) on filters.

Finally, filters can be chained together internally with 'and' or 'or' to create more complex expressions:

- \$Filter=[Field] [Operator] [Value] or/and [Operator]([Field], [Value])

Either style of filter can be chained together this way.

Some examples:

- Get every Contact with the surname Johnson - [https://api-batch-v1.prospect365.com/Contacts?\\$filter=Forename eq 'Johnson'](https://api-batch-v1.prospect365.com/Contacts?$filter=Forename eq 'Johnson')
- Get all Products excluding Milk - [http://services.odata.org/V4/OData/OData.svc/Products/?\\$filter=Name ne 'Milk'](http://services.odata.org/V4/OData/OData.svc/Products/?$filter=Name ne 'Milk')
- Get every customer in the UK where the Company name ends with an S - [http://services.odata.org/Northwind/Northwind.svc/Customers/?\\$filter=Country eq 'UK' and endswith\(CompanyName,'s'\)](http://services.odata.org/Northwind/Northwind.svc/Customers/?$filter=Country eq 'UK' and endswith(CompanyName,'s'))

\$OrderBy

The orderBy function allows you to specify a sort for your data. The syntax for this is:

- \$OrderBy=[Field Name] [Direction]

Direction is either ascending or descending. Some examples:

- Every Contact at ProspectSoft, ordered by the date the record was created - [https://api-batch-v1.prospect365.com/Contacts?\\$filter=contains\(Email,'@prospectsoft.com'\)&\\$orderby=created](https://api-batch-v1.prospect365.com/Contacts?$filter=contains(Email,'@prospectsoft.com')&$orderby=created)
- Every Product ordered by release date - [http://services.odata.org/V4/OData/OData.svc/Products/?\\$orderby=ReleaseDate asc](http://services.odata.org/V4/OData/OData.svc/Products/?$orderby=ReleaseDate asc)
- Summary of sales by year, largest subtotal first - [http://services.odata.org/Northwind/Northwind.svc/Summary_of_Sales_by_Years/?\\$orderby=Subtotal desc](http://services.odata.org/Northwind/Northwind.svc/Summary_of_Sales_by_Years/?$orderby=Subtotal desc)

\$Top and \$Skip

Top and Skip tend to go hand in hand. Top can be used to limit the number of records returned, while skip allows you to start from further down your list of results.

The syntax for these is:

- \$Top=[number]
- \$Skip=[number]

Some examples:

- Get the first 50 Contacts, skipping the first Contact - [https://api-batch-v1.prospect365.com/Contacts?\\$top=50&\\$skip=1](https://api-batch-v1.prospect365.com/Contacts?$top=50&$skip=1)
- First 5 Products ordered by release date - [http://services.odata.org/V4/OData/OData.svc/Products/?\\$orderby=ReleaseDate asc&\\$top=5](http://services.odata.org/V4/OData/OData.svc/Products/?$orderby=ReleaseDate asc&$top=5)
- The 10 biggest sales - [http://services.odata.org/Northwind/Northwind.svc/Summary_of_Sales_by_Years/?\\$orderby=Subtotal desc&\\$top=10](http://services.odata.org/Northwind/Northwind.svc/Summary_of_Sales_by_Years/?$orderby=Subtotal desc&$top=10)

Further Information

For details of the endpoints and the OData data models, please refer to the [API documentation](#).

Did this answer your question?



The #1 Stock-Aware CRM



 We run on Intercom