OData Guide
ADP Copyright Information

ADP, the ADP logo, and Always Designing for People are trademarks of ADP, LLC.

All other trademarks are the property of their respective owners.

Copyright © 2019 ADP, LLC. ADP Proprietary and Confidential - All Rights Reserved. These materials may not be reproduced in any format without the express written permission of ADP, LLC.

ADP provides this publication "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. ADP is not responsible for any technical inaccuracies or typographical errors which may be contained in this publication. Changes are periodically made to the information herein, and such changes will be incorporated in new editions of this publication. ADP may make improvements and/or changes in the product and/or the programs described in this publication.

Publication Date: 7/9/2019 7:57 AM
Overview

ADP application programming interfaces (APIs) support the following options defined in Open Data Protocol (OData):

- $select
- $filter
- $top
- $skip

These options allow a client device to control the representation it gets back from the server. Multiple query parameters may be used together by separating each option with the & character.

About OData

OData is a standardized protocol built over existing Hypertext Transfer Protocol (HTTP) and Representational State Transfer (REST) protocols supporting CRUD (Create, Read, Update, Delete) operations for creating and consuming data APIs.

For more information, see: https://www.odata.org/getting-started/understand-odata-in-6-steps/.

For URL conventions, see: https://www.odata.org/documentation/odata-version-3-0/url-conventions/.

OData Parameter Types

About the $select Parameter

The $select option selects a subset of properties to include in the response.

The following are examples using $select:

<table>
<thead>
<tr>
<th>To get …</th>
<th>Use the following query …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only the legal name of each worker:</td>
<td>GET <a href="https://api.adp.com/hr/v2/workers?$select=workers/person/legalName">https://api.adp.com/hr/v2/workers?$select=workers/person/legalName</a></td>
</tr>
<tr>
<td>Both the legal name and Associate Object Identifier (AOID) of each worker:</td>
<td>GET <a href="https://api.adp.com/hr/v2/workers?$select=workers/associateOID">https://api.adp.com/hr/v2/workers?$select=workers/associateOID</a>, workers/person/legalName</td>
</tr>
</tbody>
</table>
About the $filter Parameter

The $filter option filters a collection of resources addressed by a request URL. The expression specified with $filter is evaluated for each resource in the collection, and only items where the expression evaluates to true are included in the response.

The following are examples using $filter:

<table>
<thead>
<tr>
<th>To get …</th>
<th>Use the following query …</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only workers reporting to Jane Doe:</td>
<td>GET <a href="https://api.adp.com/hr/v2/workers?$filter=workers/workAssignments/reportsTo/positionID%20eq%20%2793V000061%27">https://api.adp.com/hr/v2/workers?$filter=workers/workAssignments/reportsTo/positionID%20eq%20%2793V000061%27</a></td>
</tr>
<tr>
<td>To get workers reporting to Jane Doe and workers whose status is Active:</td>
<td>GET <a href="https://api.adp.com/hr/v2/workers?$filter=workers/workAssignments/assignmentStatus/statusCode/codeValue">https://api.adp.com/hr/v2/workers?$filter=workers/workAssignments/assignmentStatus/statusCode/codeValue</a> eq 'A' and workers/workAssignments/reportsTo/positionID eq '94N000061'</td>
</tr>
</tbody>
</table>

Note: Supported $filter resource paths will vary by ADP product.

About the $top Parameter

The $top option requests the number of items in the queried collection to be included in the result.

The following request returns the first two workers of the workers collection:

GET https://api.adp.com/hr/v2/workers?$top=2

About the $skip Parameter

The $skip option specifies the number of items in the queried collection to be skipped and not included in the result.

GET https://api.adp.com/hr/v2/workers?$skip=18
Pagination Using $top and $skip for Large Collections

The following request returns 20 workers, starting with the 51st worker of the workers collection while using the HR V2 Workers API and pagination using $top and $skip for large collections:

GET https://api.adp.com/hr/v2/workers?$top=20&$skip=50

Note: While using Pagination, when you reach the end of all the records, a 204 response code with No Content will be returned.

Checking Whether a GET Call Supports $select or $filter

OData parameters can only be used with the GET calls. You need to call the meta version of the API. This can be done by appending /meta to the end of the resource.

Sample Response Snippet

The following is a sample response snippet from hr/v2/workers. The field "queryOptionCode": "$select" shows $select can be used as an OData parameter for the API and the following objects:

- email
- fax
- landline phones
- mobile phones
In the same way you can find out about the $filter parameter. If the "/meta" response results in tags such as "queryOptionCode": "$filter", it implies the API supports the $filter parameter.

In the following sample response snippet from hr/v2/workers the $filter parameter can be used as:

/hr/v2/workers/workAssignments/reportsTo/positionID%20eq%20%2794N293601%27

In the following snippet, 94N293601 is the value on which we are applying the filter:

```json
{
  "itemID": "q2",
  "queryOptionCode": "$filter",
  "queryOptionTypeCode": "OData",
  "resourcePaths": [
    "workers/workAssignments/reportsTo/positionID"
  ],
  "queryValueCodeList": {
    "listItems": [
      {
        "codeValue": "94N293601",
        "shortName": "iattam1Manager, Test"
      }
    ]
  }
}
```
Examples of $select and $filter Parameters

The following examples are covered in this section:

- Example 1: Using the $select Parameter
- Example 2: Using the $filter Parameter

The difference between the $select and $filter parameters is you cannot use `eq` for the $select parameter.

Example 1: Using the $select Parameter

If you're using the $select parameter, then the request will look like the following:

```
hr/v2/workers?$filter=workers/workerID/idValue
```

The response contains `idValue(s)` for all the workers.
Example 2: Using the $filter Parameter

If you're using the $filter parameter, then the request will look like the following:

hr/v2/workers?$filter=workers/workerID/idValue eq '1A9HUHNNN'

When you are requesting a specific value by using the eq field name, you must use the $filter parameter. In the previous example, the response contains worker details with the idValue equal to 1A9HUHNNN.

The $filter parameter supports field values in single quotes only.

Back to the Top of Section