Aras Innovator 2026 Release

RESTful API

Document #: *D-008126*

Last Modified: *10/16/2026*

Copyright Information

Copyright © 2026 Aras Corporation. All Rights Reserved.

Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Phone: 978-691-8900

E-mail: support@aras.com

Website: https://www.aras.com/

Notice of Rights

Copyright © 2026 by Aras Corporation and/or its affiliates. All rights reserved.

This document is protected by U.S. and international copyright laws and conventions. No copyright may be obscured or removed from this document. This document may not be modified or altered, or reproduced or transmitted in any form, without the explicit permission of the copyright holder.

Aras Innovator, Aras, and the Aras Corp "A" logo are registered trademarks of Aras Corporation in the United States and other countries.

All other trademarks referenced herein are the property of their respective owners.

Notice of Liability

THIS DOCUMENT IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY, AND THE CONTENTS HEREOF ARE SUBJECT TO CHANGE WITHOUT NOTICE. THE INFORMATION CONTAINED IN THIS DOCUMENT IS DISTRIBUTED ON AN "AS IS" BASIS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR A WARRANTY OF NON-INFRINGEMENT. ARAS SHALL HAVE NO LIABILITY TO ANY PERSON OR ENTITY WITH RESPECT TO ANY LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY THE INFORMATION CONTAINED IN THIS DOCUMENT OR BY THE SOFTWARE OR HARDWARE PRODUCTS DESCRIBED HEREIN.

Table of Contents

Send Us Your Comments **5**
1 Introduction **6**
2 Aras Innovator OData Interface **6**
2.1 Request Translation 6
2.1.1 OData Request URL Format *6*
2.1.2 Entities and Properties *6*
2.1.3 Item properties *8*
2.1.4 Relationships *9*
2.1.5 File Properties *10*
2.1.6 Poly Items *11*
2.1.7 Extended Properties *11*
2.1.8 Get *11*
2.1.9 Define operations *11*
2.1.10 Update operation *12*
2.1.11 Change permission operation *12*
2.1.12 Restricted properties *12*
2.2 Server methods 13
2.3 Request options 13
2.3.1 $filter *13*
2.3.2 $expand *16*
2.3.3 $select *17*
2.3.4 $orderby *17*
2.3.5 $top *17*
2.3.6 $skip *18*
2.3.7 $count *18*
2.3.8 $format *19*
2.4 Operations 19
2.4.1 Create *20*
2.4.2 Update *22*
2.4.3 Upsert *23*
2.4.4 Delete *23*
2.4.5 File operations *24*
2.4.6 Batch Requests *24*
3 Aras Innovator specific extensions **30**
3.1 Paging 30
3.2 Multilingual strings 31
3.3 Unsupported Features 31
4 Response translation **32**
4.1 Error response 32

©2026 Aras Corporation All Copyrights Reserved.

5 Vault OData interface 33
5.1 Begin file upload transaction 33
5.2 Upload file chunk 33
5.3 Commit file upload transaction 34
6 AML Enhancements in OData 36
6.1 Retrieving extended property values and attributes 36
6.2 Filtering by Extended Property Definition 37
6.3 Filtering Items Extended condition *in* 39
6.4 Filtering Items by an Extended Class or its Descendants 40
7 Using OAuth 2.0 Tokens from the Authentication Server 42
7.1 Getting the Access Token 42
7.2 Confirming Token Authentication for Aras Innovator 45

©2026 Aras Corporation All Copyrights Reserved.

Send Us Your Comments

Aras Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for future revisions.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where and what level of detail?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, indicate the document title, and the chapter, section, and page number (if available).

You can send comments to us in the following ways:

Email:
TechDocs@aras.com
Subject: Aras Product Documentation

Or,

Postal service:
Aras Corporation
100 Brickstone Square
Suite 100
Andover, MA 01810
Attention: Aras Technical Documentation

If you would like a reply, provide your name, email address, address, and telephone number.

If you have usage issues with the software, visit https://www.aras.com/support

©2026 Aras Corporation All Copyrights Reserved.

1 Introduction

Aras Innovator's RESTful API uses the Open Data Protocol (OData) to create and consume a RESTful API. OData is an ISO/IEC-approved OASIS standard that defines best practices for building and consuming RESTful APIs.

JSON enables the Aras Innovator client to communicate more efficiently with the Aras Innovator server using a more compact data format and faster processing. It also allows communication with other systems that support the OData protocol.

2 Aras Innovator OData Interface

The OData interface receives OData requests and translates them into Aras Markup Language (AML) requests. Each corresponding AML request is translated into an OData response before being returned to the requestor.

2.1 Request Translation

This section describes the following:

- OData Request URL format
- Entities and properties
- Item properties

2.1.1 OData Request URL Format

The OData Request URL points to the requested entity or entity collection. The URL path starts from the entity set of an action or function call. The following example shows the components of the OData URL:

```
http://host:port/path/SampleService.svc/Categories(1)/Products?$top=2&$orderby=Name
\_______________________________________/\____________________/ \_________________/
                  |                                |                      |
          service root URL                   resource path           query options
```

2.1.2 Entities and Properties

Entity sets contain predefined collections of entities. In Aras Innovator, each item type name represents an entity set. For example, the following request returns all items of type Part:

GET: http://host/server/odata/Part
Response:

```
{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    { "id": "…", "item_number": "P-01", … remaining Part item properties },
    … remaining Part items
  ]
}
```

©2026 Aras Corporation All Copyrights Reserved.

The client can request a specific item of type Part by passing the item ID in parentheses, as shown in the following example:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "048649CAF3F04D75928B1ECD702E23BC ",
  "item_number": "P-01",

  … other Part item properties
}
```

The client can use the *$filter* attribute in the query for Part items to return a result subset:

```
GET http://host/server/odata/Part?$filter=item_number eq 'P-01'

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    { "id": "…", "item_number": "P-01", … remaining Part item properties }
  ]
}
```

For requests that return a collection, the client can append *$count* to the request to get the number of elements contained in the collection. Use the AML attributes pagesize="1" and page="1" to limit the number of items returned in the response. The number of resulting items is stored in the response's *itemmax* attribute:

```
GET http://host/server/odata/Part/$count

Response (value of itemmax attribute of the returned <Item> unless "no items" Fault is returned):
12
```

Append the property name to the resource path to request a property such as item_number, as shown here:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')/item number

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity/item_number",
  "value": "P-01"
}
```

The OData service only returns the *item_number* property in the response. The request returns the property in the appropriate format (e.g., JSON). The client can add the *$value* attribute to the request to get the raw property value:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')/item_number/$value

Response:
P-01
```

This request returns the raw value of the *item_number* property.

©2026 Aras Corporation All Copyrights Reserved.

2.1.3 Item properties

An Item property contains the ID of the referenced item. When the client requests an item with an Item property, the property does not appear in the response unless its name is specified in either a *$select* or *$expand* list.

If the Item property is specified in a *$select* list, the OData service returns the property value referencing the corresponding item. All additional attributes of the corresponding AML element appear in the response with the annotation *@aras.**:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')?$select=created_by_id

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part(created_by_id)/$entity",
  "id": "048649CAF3F04D75928B1ECD702E23BC",

  "created_by_id@odata.navigationLink":"User('0CB01F8A2A93430B9A713F7196A85B20')",
  "created_by_id@aras.keyed_name":"Innovator Admin",
}
```

If the client specifies an Item property in an *$expand* list, the OData service returns the value of the property as a complete item:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')?$expand=created_by_id

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "048649CAF3F04D75928B1ECD702E23BC",

  "created_by_id":
  {
    "id": "0CB01F8A2A93430B9A713F7196A85B20",
    … other User item properties
  },
  … other Part item properties
}
```

The *$expand* and *$select* options can contain more than one property and can be combined in one request, as shown here:

©2026 Aras Corporation All Copyrights Reserved.

```
GET
http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')?$expand=created_by_id,Part_CAD
       &$select=item_number,owned_by_id

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part(item_number,
owned_by_id,created_by_id,Part_CAD)/$entity",
  "id": "048649CAF3F04D75928B1ECD702E23BC",
  "item_number": "P-01",

  "created_by_id":
  {
    "id": "0CB01F8A2A93430B9A713F7196A85B20",
    … other User item properties
  },

  "owned_by_id@odata.navigationLink":"User('A035BBBCCC8D40BEBED49D71E3129E6F')",
  "owned_by_id@aras.keyed_name":"Innovator User",

  "Part_CAD": [
    { "id": "…", … remaining Part CAD item properties },
    … remaining Part_CAD items
  ]


}
```

If the client requests the Item property itself, the OData interface returns all of the properties associated with the referenced item:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')/created_by_id

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#User/$entity",
  "id": "0CB01F8A2A93430B9A713F7196A85B20",
  … other User item properties
}
```

2.1.4 Relationships

The client appends the relationship name to the resource path in the request:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')/Part CAD

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part CAD",
  "value": [
    { "id": "…", … remaining Part CAD item properties },
    … remaining Part CAD items
  ]
}
```

This request returns the collection of `Part CAD` relationship items.

Note: Aras Innovator does not allow properties to start with a capital letter. For this reason, the translator identifies property names starting with capital letters as relationship types. Property names that begin with a lowercase letter are identified as regular properties.

©2026 Aras Corporation All Copyrights Reserved.

To request a related item, the client should request the `related_id` property for a specific relationship item:

```
GET http://host/server/odata/Part('0486…23BC')/Part CAD('01AA…FF3D')/related_id

Response:
{
  "@odata.context": "http://host/server/odata/server/$metadata#CAD/$entity",
  "id": "B9D77528B11B4516BB244F75BE2E606F",
  "item_number": "CAD-01",

  … other CAD item properties
}
```

The OData service only returns the `related_id` property.

2.1.5 File Properties

A request for a File property returns the File item and the corresponding `odata.media*` annotations describing the file stream:

```
GET http://host/server/odata/Document('A035…9E6F')/Document File('536E…3114')/related id

Response:
{
  "@odata.context": "http://host/server/odata/server/$metadata#File/$entity",
  "@odata.mediaContentType": "application/pdf",
  "@odata.mediaReadLink": "File('4792…89B0')/$value",
  "id": "4792…89B0",
  "filename": "PartDescription.pdf",
  … other File item properties
}
```

The OData service only returns the received File item.

To get the file content, the client adds the *$value* attribute to the resource path. In this case, the OData service constructs a file URL and redirects the client to the Vault server. The Vault server is selected based on the Vault priority assigned to the current user. For example, if the user accessing the Vault server is located in China and the server is also in China, then the server is assigned a priority of 1.

```
GET http://host/server/odata/Document('A035…9E6F')/Document_File('536E…3114')/related_id/$value
```

©2026 Aras Corporation All Copyrights Reserved.

2.1.6 Poly Items

When the client requests either an Item property or a collection with the Poly Item type, the OData service adds an *@odata.type* annotation in the response that describes the item type for each item returned by the service:

```
GET http://host/server/odata/Affected_Item('0486…23BC')/affected_id

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Affected_Item/$entity/affected_id",
  "@odata.type": "#Part",
  "id": "47921523824141E084EA4EE4C06A89B0",
  … other Part item properties
}
```

2.1.7 Extended Properties

Aras Innovator supports Extended Properties (xProperties). Assign an xProperty to a specific item type before you use it. Once you assign the xProperty, it can be used by items of the same ItemType. You can also create xClasses. xProperties assigned to Parent xClasses are inherited by the child xClasses.

Users with the correct permissions can create and maintain xProperty Definitions and Classification Trees programmatically or through the UI. End users can assign multiple xProperties to one item. They can also:

- Classify items
- Use xProperty values to search for items.

The OData Interface enables you to perform the following operations:

- Get
- Define
- Update
- Set permissions

2.1.8 Get

You can use the Get operation to retrieve x-properties by using the $select option, as shown in the following example:

GET http://host/server/odata/Part('AF1A....8A88')?$select=xp-prop1,xp-prop2

2.1.9 Define operations

Before using an explicitly defined xProperty on an item, you must define it. Add an "explicit" value to the "set" attribute and add an "explicit" attribute with the value "1".

To set an xProperty to an undefined state, add the "explicit" value to the "set" attribute and add an "explicit" attribute with the value "0".

©2026 Aras Corporation All Copyrights Reserved.

2.1.10 Update operation

If an xProperty with a value is included in the JSON body, the "value" is automatically added to the *set* attribute.

```
PATCH http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')
{
  "xp-prop": "xp-prop-value"
}
```

Response:

```
HTTP/1.1 200 OK
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "…",
  "xp-prop": "xp-prop-value",
  … remaining Part item properties
}
```

2.1.11 Change permission operation

Users can change the permission xProperty as shown in the following example.

```
PATCH http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')
{
  "xp-prop@aras.permission_id": "…"
}
```

Response:

```
HTTP/1.1 200 OK
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "…"
  … remaining Part item properties
}
```

2.1.12 Restricted properties

Item properties without "Get" permissions are returned with a null value, and the "*is_null*" attribute is equal to "0". To provide information about these properties, "@aras.restricted" metadata is returned in the item but will not be returned in the response. The same rules apply to xProperties.

Get:
`http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')?$select=*`
Response:

```
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "…",
  "xp-prop@aras.restricted": true,
  "team_id@aras.restricted": true,
  … remaining Part item properties
}
```

©2026 Aras Corporation All Copyrights Reserved.

2.2 Server methods

Aras Innovator's server methods are exposed as OData actions. OData functions are not used. The method call has the method namespace prepended to it to avoid conflicts with item types and property names. The client sends a POST request to the method's URI to call it:

```
POST http://host/server/odata/method.CalculateCost
```

The Translator uses the *Method* item type if the request body is empty or does not specify the payload type. The client specifies the payload type using the *@odata.type* annotation:

```
POST http://host/server/odata/method.DoSomething
{
  "@odata.type": "http://host/server/odata/$metadata#Person",
  "name": "John Doe",
  "age": 36
}
```

The client can also append a method to a collection or item URL. In this case, the item type and ID (if specified) are taken from the URI.

```
POST http://host/server/odata/Part('01AA…FF3D')/method.DoSomething
{
  "item_number": "P-001",
  "description": "Very important part"
}
```

2.3 Request options

Use the following OData query options to apply additional conditions to a returned data set.

2.3.1 $filter

The `$filter` option applies conditions to the returned data set. The `$filter` expression is translated to AML using the following AML features:

- The `condition` attribute of the property element
- The `<or>` and `<and>` elements
- The `where` attribute of an `Item` element.

Table 2 describes how each element of the `$filter` option is translated to AML.

©2026 Aras Corporation All Copyrights Reserved.

Table 1: Comparison Operators

Comparison Operators			
Operator	**Description**	**Example**	**AML**
`eq`	Equal	`name eq "Part Name"`	`condition="eq"` Note: OData equal is case sensitive.
`ne`	Not equal	`name ne "Part Name"`	`condition="ne"` Note: OData not equal is case sensitive.
`gt`	Greater than	`price gt 20`	`condition="gt"`
`ge`	Greater than or equal	`price ge 10`	`condition="ge"`
`lt`	Less than	`price lt 20`	`condition="lt"`
`le`	Less than or equal	`price le 20`	`condition="le"`

Table 2: Logical Operators

Logical Operators			
Operator	**Description**	**Example**	**AML element**
`and`	Logical and	`price le 200 and price gt 3.5`	`<and>`
`or`	Logical or	`price le 3.5 or price gt 200`	`<or>`
`not`	Logical negation	`not endswith(description, 'milk')`	`<not>`

©2026 Aras Corporation All Copyrights Reserved.

The `$filter` expression can contain a call to a built-in function. Table 4 shows how to translate a built-in function to AML.

Table 3: String Functions

String Functions		
Function	**Example**	**AML**

`contains`	`contains(company_name,'freds')`	`<company_name condition="like">`
		`%freds%`
		`</company_name>`

`endswith`	`endswith(company_name, 'freds')`	`<company_name condition="like">`
		`%freds`
		`</company_name>`

`startswith`	`startswith(company_name,'Alfr')`	`<company_name condition="like">`
		`Alfr%`
		`</company_name>`

2.3.1.1 Null Values

OData uses the special value null to indicate that a property does not have a value. Equality to null is translated using the condition "is null."

```
GET http://host/server/odata/Part?$filter=name eq null
```

Inequality is translated using the condition "*is not null*":

```
GET http://host/server/odata/Part?$filter=name ne null
```

You can use the previous condition with negation:

```
GET http://host/server/odata/Part?$filter=not name eq null
```

2.3.1.2 Filtering by Related Item Properties

The OData interface supports filtering using a property associated with the related item:

```
GET http://host/server/odata/Part?$filter=created_by_id/name eq 'Admin'
```

In Aras Innovator, item type names may contain spaces. This conflicts with the filter expression syntax, where a space character is used as a delimiter. To resolve the conflict, names containing a space character should be enclosed by square brackets in filter expressions, as shown:

```
GET http://host/server/odata/Part?$filter=[Part BOM]/quantity eq 2
```

©2026 Aras Corporation All Copyrights Reserved.

2.3.2 $expand

The $expand option specifies which Item property or relationship must be included in a response. If you do not specify the $expand option, Item properties are included in the response as an Item with a single id property. Relationships are not included at all.

Expanding Item property:

```
GET http://host/server/odata/Part?$expand=created_by_id
```

Expanding relationship:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')?$expand=Part CAD

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "id": "…",

  "Part_CAD":
  [
    { "id": "…", … remaining Part CAD item properties },
    { "id": "…", … remaining Part CAD item properties },
    … remaining Part CAD items
   ],
  … remaining Part item properties
}
```

Expanding property of a related item:

```
GET http://host/server/odata/Part('048649CAF3F04D75928B1ECD702E23BC')?$expand=Part
CAD($expand=related_id)

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "id": "048649CAF3F04D75928B1ECD702E23BC",

  "Part_CAD":
  [
    {
      "related_id":
      {
        "item_number": "CAD-1",
        … remaining CAD item properties
      },
      … remaining Part CAD item properties
    },
    … remaining Part CAD items
   ],
  … remaining Part item properties
}
```

Properties specified using the *$expand* option are implicitly included in the *$select* list.

©2026 Aras Corporation All Copyrights Reserved.

2.3.3 $select

The $select option defines which properties should be included in a result set. The ID property must be included:

```
GET http://host/server/odata/Part?$select=item_number

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part(item_number)",
  "value":
  [
    {"id": "…", "item number": "P-01"},
    {"id": "…", "item_number": "P-02"},
    … remaining Part items
  ]
}
```

You can also apply the *$select* option to related items:

```
GET http://host/server/odata/Part?$expand=Part CAD($select=related id)

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value":
  [
    {
      "id": "…",
      "item_number": "P-01",
      "Part_CAD":
      [
        {
          "id": "…",
          "related_id@odata.navigationLink": "CAD('CCF205347C814DD1AF056875E0A880AC')",
          "related_id@aras.keyed_name": "CAD-01",
        },
        … remaining Part CAD items
      ],
      … remaining Part item properties
    },
    … remaining Part items
  ]
}
```

If you omit the *$select* option, the OData service does not return properties with a null value. To return all properties, the client must use the star "*" value for the *$select* query option in the request.

2.3.4 $orderby

The $orderby option specifies the order of items in a result set. The Translator converts the sorting expression to AML syntax and sets the *orderBy* attribute of the *Item* element:

```
GET http://host/server/odata/Part?$orderby=item_numer desc
```

2.3.5 $top

The $top option limits the result set to the first *$top* items. In the following example, the query specifies that Part items 01-10 be returned. The Translator adds the *fetch* attribute to the AML request and specifies the *GetItemWithPropertyConditions* method as the action:

©2026 Aras Corporation All Copyrights Reserved.

```
GET http://host/server/odata/Part?$top=10

Response:
{
  "@odata.context": "serviceRoot/$metadata#Part",
  "value":
  [
    {"id": "…", "item_number": "P-01",…},
    {"id": "…", "item_number": "P-02",…},
    … remaining 8 Part items
  ]
}
```

2.3.6 $skip

The $skip option allows a specified number of items in the resulting collection to be skipped. The following example excludes parts 01-10 from the collection.

The Translator adds the *offset* attribute to the AML request and specifies the *GetItemWithPropertyConditions* method as the action:

```
GET http://host/server/odata/Part?$skip=10

Response:
{
  "@odata.context": "serviceRoot/$metadata#Part",
  "value":
  [
    … Parts starting from 11th item remaining 8 Part items
    {"id": "…", "item_number": "P-01",…},
    {"id": "…", "item_number": "P-02",…},
  ]
}
```

2.3.7 $count

If the $count option is present and set to true, the translator adds the @odata.count property to the resulting data set with a value equal to the number of items to be included in the response:

```
GET http://host/server/odata/Part?$count=true

Response:
{
  "@odata.context": "serviceRoot/$metadata#Part",
  "@odata.count": 10
  "value":
  [
    {"id": "…", "item_number": "P-01",…},
    {"id": "…", "item_number": "P-02",…},
    …
  ]
}
```

If the result does not return a data set *$count* is ignored.

©2026 Aras Corporation All Copyrights Reserved.

2.3.8 $format

The $format option only supports the JSON format for communication.

2.4 Operations

In OData, operations are specified as HTTP methods. Any Aras Innovator action that cannot be specified as an HTTP method is passed using the custom *@aras.action* annotation in the request body. Actions are prepended with the action namespace to avoid conflicts with item types and property names. The actions use the POST HTTP method.

Table 4: HTTP Methods

HTTP method	Aras Innovator action	Notes
GET	get	
POST	add	It can be sent only to a collection.
PATCH	edit	Can be sent only to a specific entity
PUT	edit	Update simple property or single navigation property
DELETE	delete	Can be sent only to a specific entity
PATCH	merge	Uses the `@aras.action=merge` annotation
POST	create	Uses the `@aras.action=create` annotation
PATCH	update	Uses the `@aras.action=update` annotation
PATCH	lock	Uses the `@aras.action=lock` annotation
PATCH	unlock	Uses the `@aras.action=unlock` annotation
DELETE	purge	Uses the `@aras.action=purge` annotation

The following example shows how the action can be used:

```
PATCH http://host/server/odata/Part('0486…23BC')
Prefer: return=minimal
{
  "@aras.action": "edit",
  "item_number": "P-025"
}
```

©2026 Aras Corporation All Copyrights Reserved.

```
Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('0486…23BC')
```

If the action is successful, all data modification operations except DELETE return a modified item representation if the client specified *return=representation* in the *Prefer* header. When the client specified *return=minimal*, the OData service returns the *204 No Content* status code.

The OData specification does not define service behavior if the *return* preference is omitted. To be consistent with Aras Innovator, the OData service uses *return=representation* as the default.

The response's *Location* header must point to a created or modified item.

2.4.1 Create

The client must send a POST request to the collection URL to create an item. The client can also call a create or merge action. The request body must contain a single item containing all the required properties. If the HTTP Prefer header is set to return=minimal in the request, the OData service returns a 204 No Content status code. Otherwise, the service returns a 201 Created status code along with the item's properties:

```
POST http://host/server/odata/Part
{
  "item_number": "P-01"
}

Response:
HTTP/1.1 201 Created
Location: http://host/server/odata/Part('0486…23BC')
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "0486…23BC",
  "item_number": "P-01",
  … other Part item properties
}
```

The client can create an item referenced by an item property in a single request (in OData, this is referred to as a deep insert):

```
POST http://host/server/odata/Action
Prefer: return=minimal
{
 "name": "New Action",
 "type": "Item",
 "location": "Client",
 "target": "Main",
 "method":
    {
      "name": "NewMethod",
      "method_type": "C#",
      "method_text": "test"
    }
}
Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Action('01AA…FF3D')
```

©2026 Aras Corporation All Copyrights Reserved.

The client can also create relationships between items:

```
POST http://host/server/odata/Part
Prefer: return=minimal
{
  "item_number": "P-01",
  "Part_CAD": [{
    "related_id":
    {
      "item_number": "CAD-01"
    }
  }]
}

Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('0486…23BC')
```

The client can use @odata.bind annotation to add a reference to an existing item:

```
POST http://host/server/odata/Part
Prefer: return=minimal
{
  "item_number": "P-01",
  "Part_CAD": [{
    "related_id@odata.bind": "CAD('01AA…FF3D')"
  }]
}

Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('0486…23BC')
```

The client can send a POST request to a relationship property to add a new relationship item to an existing item:

```
POST http://host/server/odata/Part('C542F…EF6B')/Part_CAD
Prefer: return=minimal
{
  "related_id@odata.bind": "CAD('01AA…FF3D')"
}

Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('C542F…EF6B')
```

If the Part item type is versionable, the POST request creates a new version of the item.

The client can also create a new relationship and related item in “Innovator style”. In this case, a new item version is not created:

```
POST http://host/server/odata/Part_CAD
Prefer: return=minimal
{
  "source_id@odata.bind": "Part('01AA…FF3D')"
  "related_id":
  {
    "item_number": "CAD-01"
  }
}

Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part_CAD('C542F…EF6B')
```

©2026 Aras Corporation All Copyrights Reserved.

2.4.2 Update

To update an existing item, the client sends a PATCH request to the specific item URL. The request body must contain a single item with updatable properties:

```
PATCH http://host/server/odata/Part('01AA…FF3D')
{
  "item_number": "P-100",
}

Response:
HTTP/1.1 200 OK
Location: http://host/server/odata/Part('01AA…FF3D')
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "01AA…FF3D",
  "item_number": "P-01",
  … other Part item properties
}
```

The client can also include *edit* or *update* actions in the PATCH request for data modification:

```
POST http://host/server/odata/Part('01AA112937924D5383FB4A101B2AFF3D')
Prefer: return=minimal
{
  "@aras.action": "update",
  "item_number": "P-100"
}

Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('01AA…FF3D')
```

To specify a value for a simple property, the client must send a PUT request to the corresponding property URL:

```
PUT http://host/server/odata/Part('01AA112937924D5383FB4A101B2AFF3D')/item_number
Prefer: return=minimal
{
  "value": "P-001"
}

Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('01AA…FF3D')/item_number
```

The client can also send a PUT request to the $value endpoint of a simple property where the body of the request contains a raw property value:

```
PUT http://host/server/odata/Part('01AA112937924D5383FB4A101B2AFF3D')/item_number/$value
Prefer: return=minimal

P-001


Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('01AA…FF3D')/item number/$value
```

To set a value for an Item property, the client must send a PUT request to the Item property reference:

©2026 Aras Corporation All Copyrights Reserved.

```
PUT http://host/server/odata/Part('01AA112937924D5383FB4A101B2AFF3D')/owned_by_id/$ref
Prefer: return=minimal
{
  "@odata.id": "http://host/server/odata/User('C542FC153AE647A59CD6F6967295EF6B')"
}

Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Part('01AA…FF3D')
```

Update actions can only modify one item at a time.

2.4.3 Upsert

To perform an upsert operation, the client sends a PATCH request to a specific entry and include an HTTP header *If-Match* with the value "*" in the request. The OData service translates it to AML using the *merge* action:

```
PATCH http://host/server/odata/Part('01AA…FF3D')
If-Match: *
{
  "item_number": "P-100",
}

Response:
HTTP/1.1 200 OK
Location: http://host/server/odata/Part('01AA…FF3D')
{
  "@odata.context": "http://host/server/odata/$metadata#Part/$entity",
  "id": "01AA…FF3D",
  "item_number": "P-01",
  … other Part item properties
}
```

2.4.4 Delete

To delete an existing item, the client must send a DELETE request to the specific item URL:

```
DELETE http://host/server/odata/Part('01AA112937924D5383FB4A101B2AFF3D')

Response:
HTTP/1.1 204 No Content
```

The client uses the *@aras.action=purge* annotation within the body of the query to delete only one version of the item:

```
DELETE http://host/server/odata/Part('01AA112937924D5383FB4A101B2AFF3D')
Prefer: return=minimal
{
  "@aras.action": "purge"
}

Response:
HTTP/1.1 204 No Content
```

To remove a relationship, the client sends a DELETE request to the *$ref* URL of the corresponding relationship property, passing the $id query string option:

©2026 Aras Corporation All Copyrights Reserved.

```
DELETE http://host/server/odata/Part('01AA…FF3D')/Part_CAD/$ref?$id=Part_CAD('B9D7…606F')

Response:
HTTP/1.1 204 No Content
```

To clear an item property, the client sends a DELETE request to the *$ref* URL for the corresponding property. The Purge action cannot be used for this request:

```
DELETE http://host/server/odata/Part('01AA112937924D5383FB4A101B2AFF3D')/owned_by_id/$ref

Response:
HTTP/1.1 204 No Content
```

A Delete action can only remove one item at a time.

Note: Because Aras Innovator sends a success response for a delete operation, even if the item being deleted cannot be found, the OData service also sends a success response.

2.4.5 File operations

The Aras Innovator OData interface only supports get and delete operations for File items. You must use the Vault OData interface to perform a Create operation. The Update operation is not supported.

2.4.6 Batch Requests

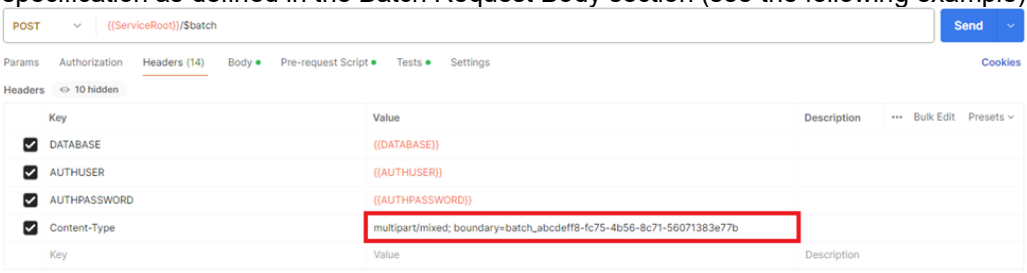
In OData, batching enables developers to bundle multiple requests into a single call to the *$batch* endpoint (see OData documentation). A batch request is represented as a Multipart MIME message, a standard format allowing the representation of multiple parts, each of which may have a different content type, within a single request.

Batch requests are submitted as a single **HTTP POST** request to the $batch endpoint of a Web Service. Any requests nested in a batch are executed sequentially and synchronously.

2.4.6.1 Batch Request Headers

This section describes the headers for an HTTP POST request to the $batch endpoint.

- **Content-Type**: Required. This header must have the *multipart/mixed* value and a boundary specification as defined in the Batch Request Body section (see the following example).



- **OData-Version**: Optional. If using this header, set the value to *4.0*.
- **Prefer**: Optional. By default, the CWS engine stops processing a batch request when any of the nested requests return an error. To continue processing the batch in case of an error in a nested request, include the Prefer header with the value *odata.continue-on-error*.

2.4.6.2 Batch Request Body

The body of a batch request comprises a series of individual requests and Change Sets, each represented as a distinct MIME part (i.e., separated by the boundary defined in the *Content-Type* header).

©2026 Aras Corporation All Copyrights Reserved.

An individual request in the context of a batch request can be

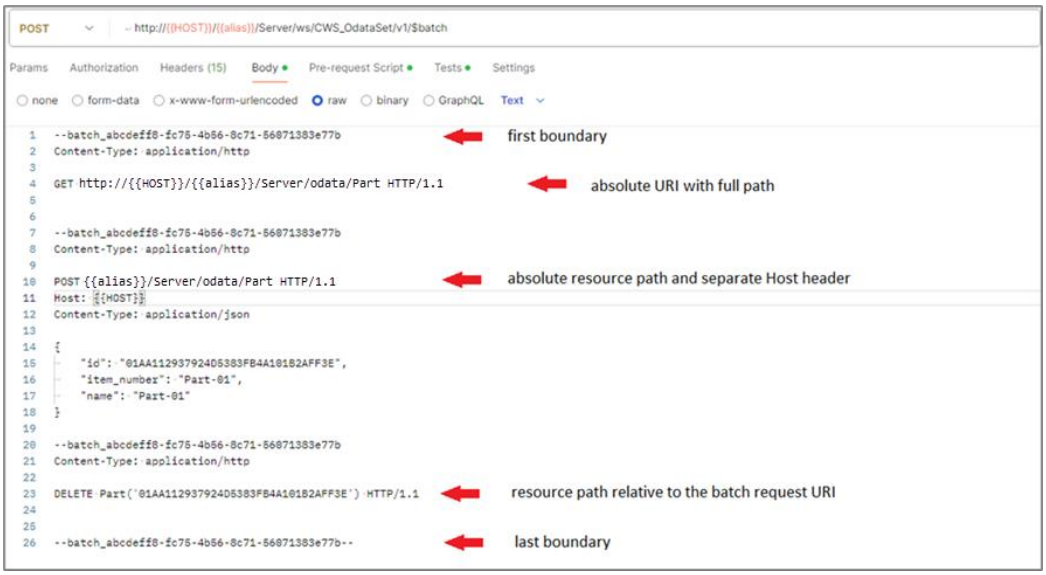
- Data request;
- Data Modification request;
- Action invocation request.

A MIME part representing an individual request must include a *Content-Type* header with the value *application/http* and should contain a *Content-Transfer-Encoding* header with the value *binary*.

CWS supports three Request-URI formats of HTTP requests serialized within MIME part bodies:

- Absolute URI with schema, host, port, and absolute resource path
  - The absolute URI of each request in a batch must be the same as the URI of the source batch request.
- Absolute resource path and separate Host header
- Resource path relative to the batch request URI

The following sample batch request demonstrates each of these formats.



Note: The formatting of the batch request body is essential. An empty line must separate each part. Add an extra empty line after requests without a body (see line 4 in the preceding example).

2.4.6.3 Change Sets

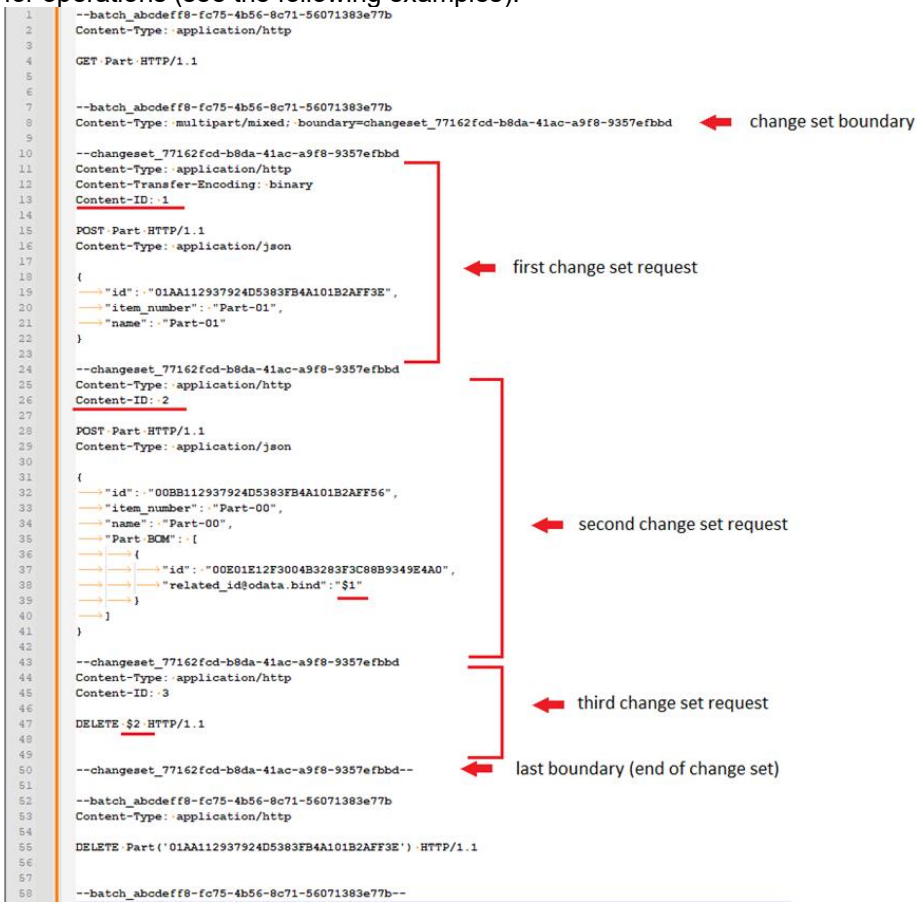
A **Change Set** is an atomic unit of work consisting of an unordered group of one or more *Data Modification* requests or *Action* invocation requests.

All Change Sets must meet the following requirements:

- Change Sets cannot contain any GET requests or other Change Sets.
- The order of requests within a Change Set is significant; the requests within a Change Set are processed sequentially.
- If one request within a Change Set fails, then all requests in the Change Set are rolled back.
- Each Change Set must be a multipart MIME document with one part for each operation that makes up the Change Set.
- Each part representing an operation in the Change Set must include the same headers (*Content-Type* and *Content-Transfer-Encoding*) and associated values previously described

©2026 Aras Corporation All Copyrights Reserved.

for operations (see the following examples).

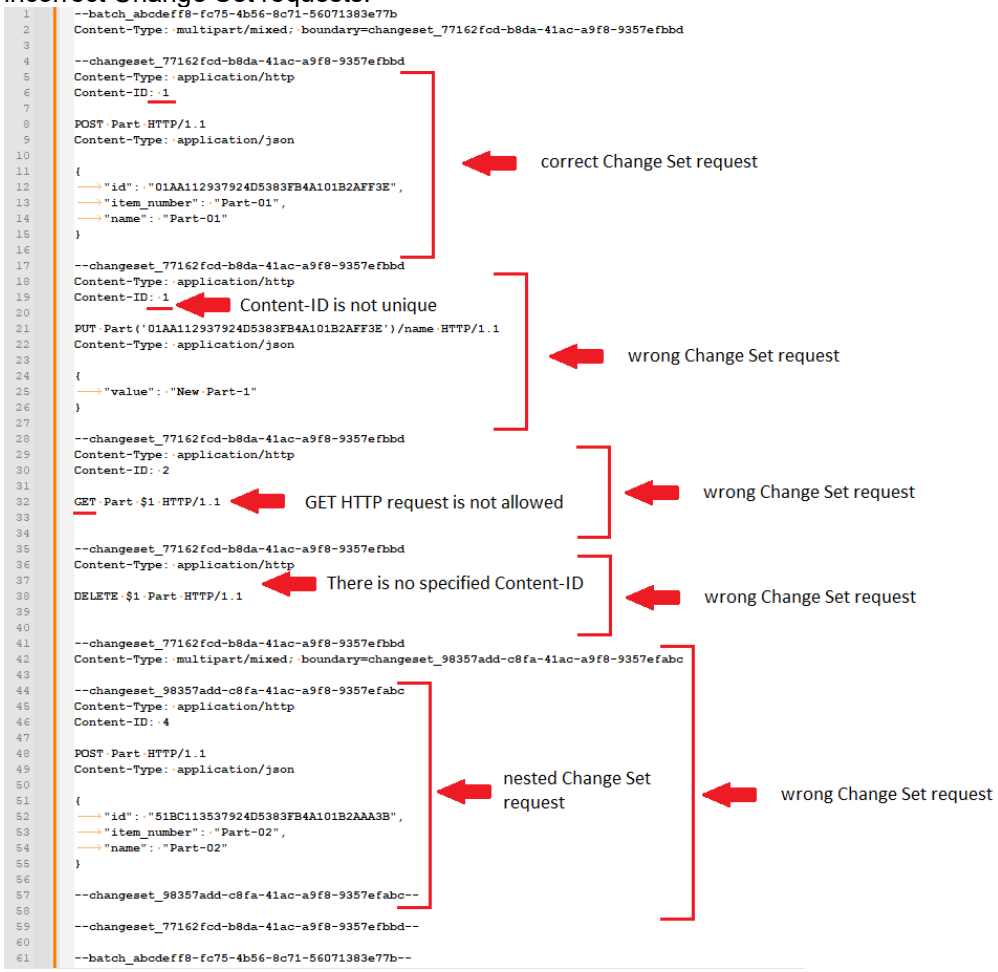



```
--batch_abcdeff8-fc75-4b56-8c71-56071383e77b
Content-Type: application/http

GET Part HTTP/1.1


--batch_abcdeff8-fc75-4b56-8c71-56071383e77b
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-a9f8-9357efbbd

--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-Transfer-Encoding: binary
Content-ID: 1

POST Part HTTP/1.1
Content-Type: application/json

{
    "id": "01AA112937924D5383FB4A101B2AFF3E",
    "item_number": "Part-01",
    "name": "Part-01"
}

--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 2

POST Part HTTP/1.1
Content-Type: application/json

{
    "id": "00BB112937924D5383FB4A101B2AFF56",
    "item_number": "Part-00",
    "name": "Part-00",
    "Part BOM": [
        {
            "id": "00E01E12F3004B3283F3C88B9349E4A0",
            "related_id@odata.bind":"$1"
        }
    ]
}

--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 3

DELETE $2 HTTP/1.1


--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd--

--batch_abcdeff8-fc75-4b56-8c71-56071383e77b
Content-Type: application/http

DELETE Part('01AA112937924D5383FB4A101B2AFF3E') HTTP/1.1


--batch_abcdeff8-fc75-4b56-8c71-56071383e77b--
```

- Each request within a Change Set must specify a *Content-ID* header with a value unique within the batch request.
- Entities created by an Insert request within a Change Set can be referenced by subsequent requests within the same Change Set in places where a resource path to an existing entity can be specified. The temporary resource path for a newly inserted entity is the value of the *Content-ID* header prefixed with a $ character. The following examples show correct and

©2026 Aras Corporation All Copyrights Reserved.

incorrect Change Set requests:




```
--batch_abcdeff8-fc75-4b56-8c71-56071383e77b
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-a9f8-9357efbbd

--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 1

POST Part HTTP/1.1
Content-Type: application/json

{
    "id": "01AA112937924D5383FB4A101B2AFF3E",
    "item_number": "Part-01",
    "name": "Part-01"
}

--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 1

PUT Part('01AA112937924D5383FB4A101B2AFF3E')/name HTTP/1.1
Content-Type: application/json

{
    "value": "New Part-1"
}

--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 2

GET Part $1 HTTP/1.1


--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http

DELETE $1 Part HTTP/1.1


--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: multipart/mixed; boundary=changeset_98357add-c8fa-41ac-a9f8-9357efabc

--changeset_98357add-c8fa-41ac-a9f8-9357efabc
Content-Type: application/http
Content-ID: 4

POST Part HTTP/1.1
Content-Type: application/json

{
    "id": "51BC113537924D5383FB4A101B2AAA3B",
    "item_number": "Part-02",
    "name": "Part-02"
}

--changeset_98357add-c8fa-41ac-a9f8-9357efabc--

--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd--

--batch_abcdeff8-fc75-4b56-8c71-56071383e77b--
```

2.4.6.4 *Responding to a Batch Request*

Requests within a batch are evaluated according to the same semantics used when processing individual requests. The preceding sample batch request generated the following response.

```
--batch_abcdeff8-fc75-4b56-8c71-56071383e77b
Content-Type: application/http

HTTP/1.1 200 OK
OData-Version:4.0
Content-
Type:application/json;odata.metadata=minimal;IEEE754Compatible=false

{
  "@odata.context":
"http://{{host}}/{{alias}}/Server/odata/$metadata#Part",
  "value": []
}
--batch_abcdeff8-fc75-4b56-8c71-56071383e77b
Content-Type: multipart/mixed; boundary=changeset_77162fcd-b8da-41ac-
a9f8-9357efbbd
```

©2026 Aras Corporation All Copyrights Reserved.

```
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 1

HTTP/1.1 201 Created
Preference-Applied:return=representation
Location:http://{{host}}/{{alias}}/Server/odata/Part('01AA112937924D538
3FB4A101B2AFF3E')
OData-Version:4.0
Content-
Type:application/json;odata.metadata=minimal;IEEE754Compatible=false

{
  "@odata.context":
"http://{{host}}/{{alias}}/Server/odata/$metadata#Part/$entity",
  "id": "01AA112937924D5383FB4A101B2AFF3E",
  "major_rev": "A",
  "name": "Part-01",
  "item_number": "Part-01"
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 2

HTTP/1.1 201 Created
Preference-Applied:return=representation
Location:http://{{host}}/{{alias}}/Server/odata/Part('00BB112937924D538
3FB4A101B2AFF56')
OData-Version:4.0
Content-
Type:application/json;odata.metadata=minimal;IEEE754Compatible=false

{
  "@odata.context":
"http://{{host}}/{{alias}}/Server/odata/$metadata#Part/$entity",
  "id": "00BB112937924D5383FB4A101B2AFF56",
  "major_rev": "A",
  "name": "Part-00",
  "item_number": "Part-00",
  "Part_BOM": [
    {
      "id": "00E01E12F3004B3283F3C88B9349E4A0",
      "sort_order": 128
    }
  ]
}
--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd
Content-Type: application/http
Content-ID: 3

HTTP/1.1 204 NoContent
OData-Version:4.0
Content-Type:text/plain


--changeset_77162fcd-b8da-41ac-a9f8-9357efbbd--
```

©2026 Aras Corporation All Copyrights Reserved.

```
--batch_abcdeff8-fc75-4b56-8c71-56071383e77b
Content-Type: application/http

HTTP/1.1 204 NoContent
OData-Version:4.0
Content-Type:text/plain


--batch_abcdeff8-fc75-4b56-8c71-56071383e77b--
```

©2026 Aras Corporation All Copyrights Reserved.

3 Aras Innovator specific extensions

This section describes the extensions that can be used in Aras Innovator to:

- Configure server-driven paging
- Retrieve the language associated with multi-lingual string types

3.1 Paging

Use the offset and fetch AML attributes to configure Server-driven paging. When the client specifies the odata.maxpagesize value in the Prefer request header, the OData Interface translates the value using a combination of the offset and fetch attributes while considering any $skip and $top options specified in the request. The @odata.nextLink annotation value includes a $skiptoken option, which determines the starting position of the next page within a requested data set. All query options from the original request appear in the @odata.nextLink URL.

The last page cannot contain the @odata.nextLink annotation. To determine whether @odata.nextLink must be present in the response, the translator produces AML that requests one additional item. If all requested items are received, @odata.nextLink must be included in the response. If the number of returned items is less than requested, @odata.nextLink must be omitted.

```
GET http://host/server/odata/Part?$skip=3&$top=30
Prefer: odata.maxpagesize=10

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "@odata.nextLink": "http://host/server/odata/Part?$skip=3&$top=30&$skiptoken=10",
  "value": [
    { "id": "…", "item_number": "P-01", … remaining Part item properties },
    … 9 remaining Part items
  ]
}
```

The request for the next page is translated as follows:

```
GET http://host/server/odata/Part?$skip=3&$top=30&$skiptoken=10
Prefer: odata.maxpagesize=10

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "@odata.nextLink": "http://host/server/odata/Part?$skip=3&$top=30&$skiptoken=20",
  "value": [
    { "id": "…", "item_number": "P-10", … remaining Part item properties },
    … 9 remaining Part items
  ]
}
```

©2026 Aras Corporation All Copyrights Reserved.

3.2 Multilingual strings

The properties of the multilingual string type are returned using the *lang* attribute to specify the language used by the string. The *lang* attribute is translated to *@aras.lang* annotation for this property:

```
GET http://host/server/odata/ItemType('4F1A…8A88')?$select=label

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#ItemType(label)/$entity",
  "id": "4F1A…8A88",
  "label": "Part",
  "label@aras.lang": "en"
}
```

The client can also ask for the property value specifying a language other than the language used in the session or all the languages defined in the database. In AML, the *language* attribute serves this purpose. It may contain a comma-separated list of languages or star "*" specifying all the languages in the database. The OData service introduces the *lang* query option with the same meaning and *#aras.lang.** annotation qualifier that is used to return the property or property attribute value in the requested languages:

```
GET http://host/server/odata/ItemType('4F1A…8A88')?$select=label&lang=*

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#ItemType(label)/$entity",
  "id": "4F1A…8A88",

  "label": "Part",
  "label@aras.lang": "en",
  "label@aras.custom_attr": "PPP",

  "label@#aras.lang.ru": "Деталь"
  "label@aras.custom_attr#aras.lang.ru": "ДДД"
}
```

3.3 Unsupported Features

Currently, the OData interface does not support the following terms. It responds with a "501 Not Implemented" HTTP code if you use one of them:

- *$all* resource path component
- *$crossjoin* resource path component
- POST request to reference ($ref) of collection navigation property
- Asynchronous requests
- Using the number of related items as sorting criteria:
  http://host/server/odata/Part?$orderby=Part_CAD/$count
- Asynchronous operations
- Delta responses

©2026 Aras Corporation All Copyrights Reserved.

4 Response translation

This section describes the response types supported by the OData server and how they translate to AML.

4.1 Error response

An Error response reports OData service-specific errors and unsupported functionality, as well as AML error responses:

```
Response:
HTTP/1.1 400 Bad Request
{
  "code": "400",
  "message": "Bad Request",
  "target": "AML",
  "details":
  {
    "code": "SOAP-ENV:Server",
    "message": "addRelshipPermInfo is not specified or its sourceID is empty",
    "target": "System.ArgumentException"
  }
}
```

©2026 Aras Corporation All Copyrights Reserved.

5 Vault OData interface

The content Vault servers store files. When there is more than one Vault server, each server may be in a different location. All Vault servers have an assigned priority based on the user's location. For example, if a user in China tries to access a Vault Server also located in China, then that Vault server has priority 1 because of its proximity to the user.

The content Vault server has its own OData interface for uploading file content. This interface exposes the three methods described in the following sections.

5.1 Begin file upload transaction

The *vault.BeginTransaction* Method starts the file upload transaction and returns a transaction ID:

```
POST http://vault/odata/vault.BeginTransaction
VAULTID: {{vault_id}}

Response:
HTTP/1.1 200 OK
Content-Type: application/json
{
  "transactionId": "8970e933322d1328537f645d3683214c"
}
```

Note: You must add the `VAULTID` header to the vault.BeginTransaction request. To receive the `{{vault_id}}` value, send the following request:

```
GET http://host/server/odata/User('{{current_user_id}}')?$select=default_vault

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#User(default_vault)/$entity",
  "@odata.id": "User({{current_user_id}})",
  "default_vault@odata.associationLink": "User('{{current_user_id}}')/default_vault/$ref",
  "default_vault@odata.navigationLink": "User('{{current_user_id}}')/default_vault",
  "default_vault@aras.keyed_name": "Default",
  "default vault@aras.id": "{{vault id}}"
}
```

5.2 Upload file chunk

The *vault.UploadFile* Method uploads the file chunk within a given transaction. The File ID is GUID generated on the client and passed to the vault server using the *fileId* query option. The upload must use Transfer-encoding: chunked and containing a Content-range header where the range is the range of bytes within the chunk of the file being transferred and the size is the file size. A Content-disposition header must be included containing the name of the file being uploaded.

©2026 Aras Corporation All Copyrights Reserved.

```
POST http://vault/odata/vault.UploadFile?fileId=2778F0A17FDF4095A497A9A27B594376
VAULTID: {{vault_id}}
Content-Disposition:attachment; filename*=utf-8''file 0.txt
Content-Length:53
Content-Range:bytes 0-52/53
Content-Type:application/octet-stream
Aras-Content-Range-Checksum:2535782373
Aras-Content-Range-Checksum-Type:xxHashAsUInt32AsDecimalString
transactionid:{{transaction_id}}

… file chunk content

Response:
HTTP/1.1 200 OK
```

Note: You must add a `VAULTID` header to the vault.UploadFile request. To receive the `{{vault_id}}` value, send the request as shown in the following example.

```
GET http://host/server/odata/User('{{current_user_id}}')?$select=default_vault

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#User(default_vault)/$entity",
  "@odata.id": "User({{current_user_id}})",
  "default_vault@odata.associationLink": "User('{{current_user_id}}')/default_vault/$ref",
  "default_vault@odata.navigationLink": "User('{{current_user_id}}')/default_vault",
  "default_vault@aras.keyed_name": "Default",
  "default_vault@aras.id": "{{vault_id}}"
}
```

5.3 Commit file upload transaction

The *vault.CommitTransaction* method finalizes a file upload and submits the OData representation of the item using file(s) uploaded by the *vault.UploadFile* method. This includes the file ID, filename, file size, and the Located item that points to the current user's vault. For example:

```
{
  'id': [id passed to the vault.UploadFile method],
  'filename': [the name of the file being uploaded],
  'file_size': [the size of the file in bytes],
  'located':
    [
      {
        'file_version': [the version of the file],
        'related_id': [the ID of the user's vault]
      }
    ]
}
```

Once Aras Innovator responds that the transaction is successful, the Vault server commits the transaction.

©2026 Aras Corporation All Copyrights Reserved.

```
POST http://vault/odata/vault.CommitTransaction
VAULTID: {{vault_id}}
Prefer: return=minimal
OData-Version: 4.0
transactionid: 8970e933322d1328537f645d3683214c
Content-Type: multipart/mixed; boundary=batch_36522ad7-fc75-4b56-8c71-56071383e77b
Content-Length: ###

--batch_36522ad7-fc75-4b56-8c71-56071383e77b
Content-Type: application/http

POST /odata/Document HTTP/1.1
Host: host
Content-Type: application/json

{
  "@odata.context":"http://host/server/odata/$metadata#Document/$entity"
  "@odata.type":"#Document",
  "id":"FBCC…74FC",
  "item_number": "DOC-01",
  "Document File": [{
    "id":"40F1…6369",
    "related_id":
    {
      "id":"2778F0A17FDF4095A497A9A27B594376",
      "filename": "6d039e86857376a761b87c4d559953e2.jpg"
    }
  }]
}

--batch_36522ad7-fc75-4b56-8c71-56071383e77b--


Response:
HTTP/1.1 204 No Content
Location: http://host/server/odata/Document('FBCC…74FC')
```

Note: You must add the `VAULTID` header to the vault.CommitTransaction request. To receive the `{{vault_id}}` value, send the request shown here.

```
GET http://host/server/odata/User('{{current_user_id}}')?$select=default_vault

Response:
{
  "@odata.context": "http://host/server/odata/$metadata#User(default_vault)/$entity",
  "@odata.id": "User({{current_user_id}})",
  "default_vault@odata.associationLink": "User('{{current_user_id}}')/default_vault/$ref",
  "default_vault@odata.navigationLink": "User('{{current_user_id}}')/default_vault",
  "default_vault@aras.keyed_name": "Default",
  "default_vault@aras.id": "{{vault_id}}"
}
```

©2026 Aras Corporation All Copyrights Reserved.

6 AML Enhancements in OData

This section describes enhancements to AML in the OData interface that enable you to retrieve information about extended properties and classes. You can also filter items by extended class or extended property definition. The Extended Classification feature introduces these enhancements.

6.1 Retrieving extended property values and attributes

To request the value of a specific extended property, include the xProperty name in a *$select* request:

```
GET http://host/server/odata/Part('048…3BC')?$select=xp-cost

AML:
<Item type="Part" action="get" id="048…3BC" select="xp-cost($value)" />

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part(xp-cost)/$entity",
  "@odata.id": "Part('0486…23BC')",
  "xp-cost": "{defined xp-cost value}",
  "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"
}
```

Use OData annotations to request an attribute associated with an extended property, as shown in the following example:

```
GET http://host/server/odata/Part('048…3BC')?$select=xp-cost@aras.permission_id

AML:
<Item type="Part" action="get" id="048…3BC" select="xp-cost(@permission_id)" />

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part(xp-*@aras.permission_id)/$entity",
  "@odata.id": "Part('0486…23BC')",
  "xp-text@aras.permission_id": "{Permission_PropertyValue ID}",
  "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"
}
```

If you request both a value and/or more than one attribute associated with the same property, you must combine them into one selection list element in the resulting AML:

©2026 Aras Corporation All Copyrights Reserved.

```
GET http://host/server/odata/Part('048…3BC')?$select=xp-cost,xp-cost@aras.permission_id

AML:
<Item type="Part" action="get" id="048…3BC" select="xp-cost($value,@permission_id)" />

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part(xp-*@aras.permission_id)/$entity",
  "@odata.id": "Part('0486…23BC')",
  "xp-cost": "{defined xp-cost value}",
  "xp-text@aras.permission_id": "{Permission_PropertyValue ID}",
  "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"
}
```

Use `xp-` as the property name to request all of the extended properties that are defined for an item:

```
GET http://host/server/odata/Part('048…3BC')?$select=xp-*

AML:
<Item type="Part" action="get" id="048…3BC" select="xp-*($value)" />

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part(xp-cost)/$entity",
  "@odata.id": "Part('0486…23BC')",
  "xp-cost": "{defined xp-cost value}",
  "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"

  … other Part item extended properties
}
```

6.2 Filtering by Extended Property Definition

Use `is_defined` the filter function to filter request results using the extended property definition and how the property is defined (either explicit or through extended class):

```
GET http://host/server/odata/Part?$filter=is_defined(xp-cost)

AML:
<Item type="Part" action="get">
  <xp-cost condition="is defined"/>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    {
      "id": "048…3BC ",
      "item_number": "Part_with_Defined_xp_cost",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"

      … other Part item properties
```

©2026 Aras Corporation All Copyrights Reserved.

```
    }
    … other Part items with defined xp-cost
  ]
}
```

You can request a property that is defined in a specific way:

```
GET http://host/server/odata/Part?$filter=is_defined(xp-cost,explicit)

AML:
<Item type="Part" action="get">
  <xp-cost condition="is defined" defined_as="explicit"/>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    {
      "id": "048…3BC ",
      "item number": "Part with Defined xp cost",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"

      … other Part item properties
    }
    … other Part items with explicitly defined xp-cost
  ]
}
```

Use the Negate operation to request items that do not have a specific extended property defined:

```
GET http://host/server/odata/Part?$filter=not is_defined(xp-cost,class)

AML:
<Item type="Part" action="get">
  <xp-cost condition="is not defined" defined_as="class"/>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    {
      "id": "048…3BC ",
      "item_number": "Part_with_xp_cost_not_Defined_in_Class",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"

      … other Part item properties
    }
    … other Part items with xp-cost not defined via xClass
  ]
}
```

©2026 Aras Corporation All Copyrights Reserved.

6.3 Filtering Items Extended condition *in*

To implement the `in` extended condition, the OData Interface has introduced the `in()` filter function. This function enables you to request specific information related to an item. In the following example, the first argument is the property name, the second argument is the collection to look in, and the third argument is the property name of the collection element:

```
GET
http://host/server/odata/xPropertyContainerItem?$select=id,itemtype&$filter=in(itemtype,$root/Ite
mType($select=id; $filter=name eq 'Part'),'id')

AML:
<Item type="xPropertyContainerItem" action="get" select=”name,itemtype”>
  <itemtype condition="in" by=“id">
    <Item type=“ItemType" select=“id">
      <Name condition=“eq">Part</Name>
    </Item>
  </itemtype>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#xPropertyContainerItem(id,itemtype)",
  "value": [
    {
      "@odata.type": "#Part",
      "id": "048…3BC ",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"
    }
    … other xPropertyContainerItem items with Part data_source
  ]
}
```

You can use the `in()` function with other item types as well:

```
GET http://host/server/odata/Part?$filter=in(created_by_id,$root/User($select=id;
$filter=login name eq 'jsmith'),'id')

AML:
<Item type="Part" action="get">
  <created_by_id condition="in" by="id">
    <Item type="User" select="id">
      <logon>jsmith</logon>
    </Item>
  </created_by_id>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    {
      "id": "048…3BC ",
      "item_number": "Part-1",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"

      … other Part item properties
    }
    … other Part items created by User ‘jsmith’
  ]
}
```

©2026 Aras Corporation All Copyrights Reserved.

6.4 Filtering Items by an Extended Class or its Descendants

Use the `isof_xclass('<class-name-or-id>')` function to filter an item collection by its assigned `xClass`:

```
GET http://host/server/odata/Part?$filter=isof_xclass('Bolt')

AML:
<Item type="Part" action="get">
  <Relationships>
    <Item type="Part_xClass" action="get">
      <related_id>
        <Item type="xClass" action="get">
          <name>Bolt</name>
        </Item>
      </related_id>
    </Item>
  </Relationships>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    {
      "id": "048…3BC ",
      "item_number": "Bolt",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"

      … other Part item properties
    }
    … other Part items that are classified by class 'Bolt'
  ]
}
```

If you pass the id string as a function argument, the class id is returned instead of the class name:

```
GET http://host/server/odata/Part?$filter=isof_xclass('F48…3BC')

AML:
<Item type="Part" action="get">
  <Relationships>
    <Item type="Part_xClass" action="get">
      <related_id>
        <Item type="xClass" action="get">
          <id>048…3BC</id>
        </Item>
      </related_id>
    </Item>
  </Relationships>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    {
      "id": "048…3BC ",
      "item_number": "Bolt",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"
```

©2026 Aras Corporation All Copyrights Reserved.

```
      … other Part item properties
    }
    … other Part items that classified by class (with ID='F48…3BC')
  ]
}
```

Use the `isof_xclassordescendants('<class-name-or-id>')` function to filter an item collection by its assigned `xClass` or descendants:

```
GET http://host/server/odata/Part?$filter=isof_xclassordescendants('Bolt')

AML:
<Item type="Part" action="get">
  <Relationships>
    <Item type="Part_xClass" action="get">
      <related_id>
        <Item type="xClass" action="getxClassAndAllDescendants">
          <name>Bolt</name>
        </Item>
      </related_id>
    </Item>
  </Relationships>
</Item>

Response:
HTTP/1.1 200 OK

{
  "@odata.context": "http://host/server/odata/$metadata#Part",
  "value": [
    {
      "id": "048…3BC ",
      "item number": "Bolt",
      "itemtype": "4F1AC04A2B484F3ABA4E20DB63808A88"

      … other Part item properties
    }
    … other Part items that classified by class class='Bolt' or its subclasses
  ]
}
```

©2026 Aras Corporation All Copyrights Reserved.

7 Using OAuth 2.0 Tokens from the Authentication Server

The Authentication Server enables requesting an OAuth 2.0 token from the server to use as basic authentication. The information in this chapter has been adapted from the Authenticating in OAuth 2.0 with Aras RESTful API blog.

Note: Aras Innovator version 12 and earlier requires different authentication steps. Refer to Token Authentication using the REST API blog for more information.

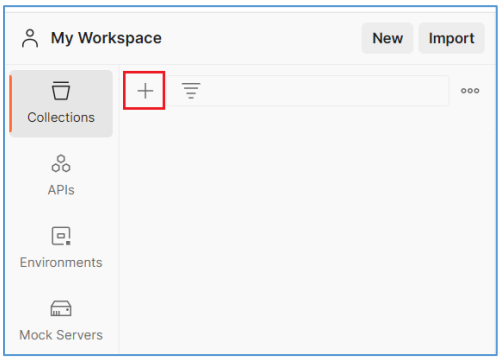
From Aras Innovator 14+ versions, 0Auth 2.0 tokens are used for token authentication.

This document uses Postman, although you can send the same request using any API platform.

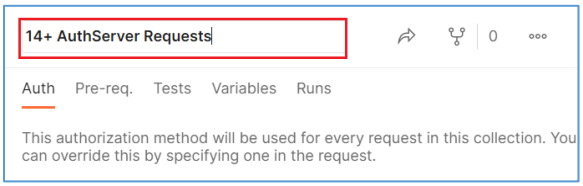
7.1 Getting the Access Token

To get the access token:

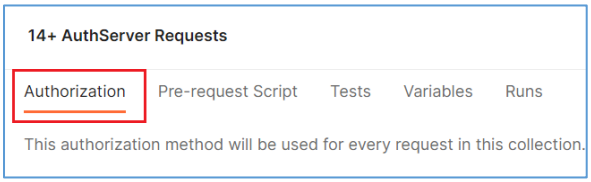
1. Download the Postman application on the local machine.
2. Under **My Workspace**, select **Collections** and click **Create New Collection**.



3. Provide a name for the new Workspace, for example, 14+ AuthServer Requests.

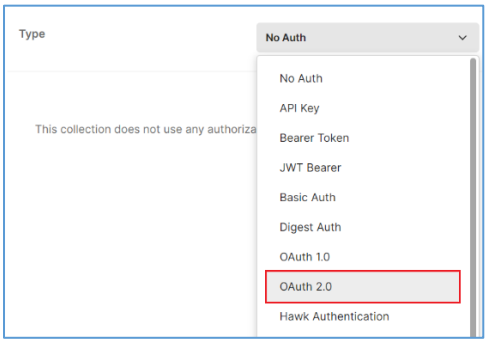


4. Select the **Authorization** tab.

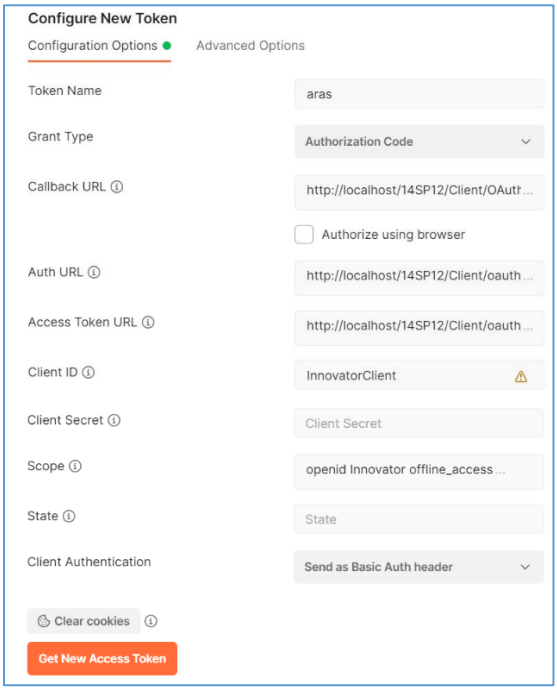


©2026 Aras Corporation All Copyrights Reserved.

5. For the **Type** field, select **OAuth 2.0** from the dropdown.



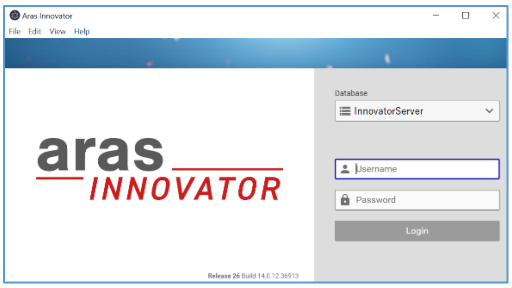
6. Scroll down to **the Configure New Token** section and set the configuration options as follows:



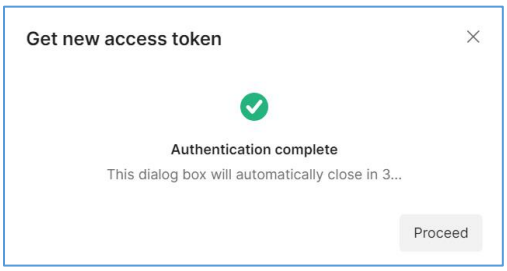
Field	Value
Token Name	Name of the Token
Grant Type	Authorization Code (With PKCE)
Callback URL	http://<servername>/<web alias>/Client/OAuth/PopupCallback
Auth URL	http://<servername>/<web alias>/oauthserver/connect/authorize
Access Token URL	http://<servername>/<web alias>/oauthserver/connect/token
Client ID	InnovatorClient
Scope	openid Innovator offline_access

©2026 Aras Corporation All Copyrights Reserved.

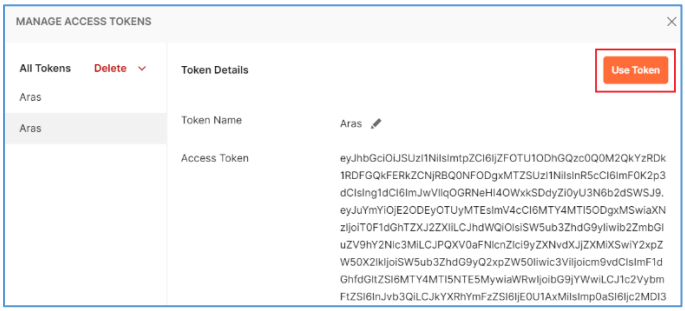
7. Click **Get New Access Token**.
8. Sign in to the popup Aras Innovator window with your credentials.



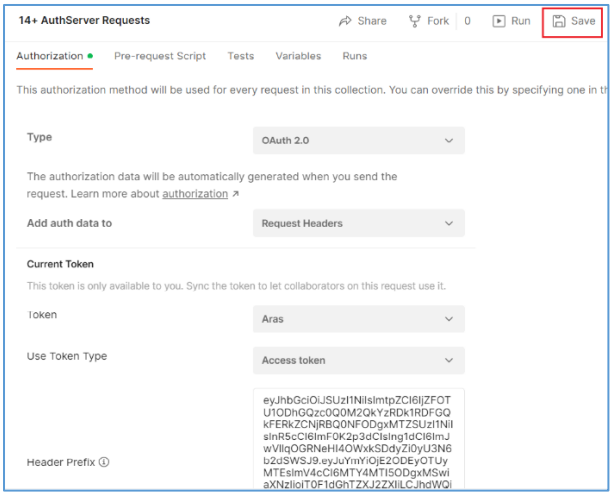
9. In the **Get new access token** dialog, click **Proceed**.



10. In the **Manage Access Token** dialog, click **Use Token**.



11. Click **Save** to save the changes.

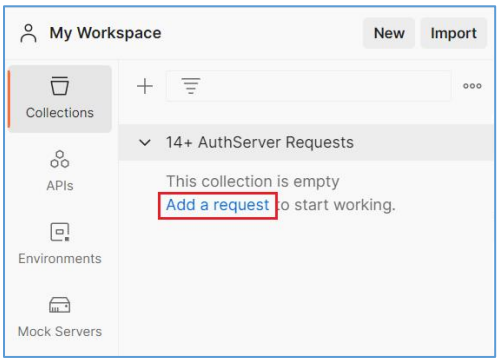


©2026 Aras Corporation All Copyrights Reserved.

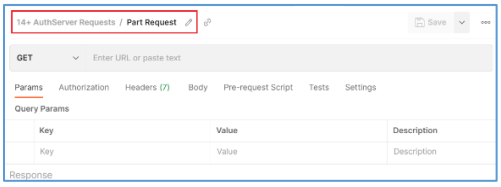
7.2 Confirming Token Authentication for Aras Innovator

To confirm that token authentication works:

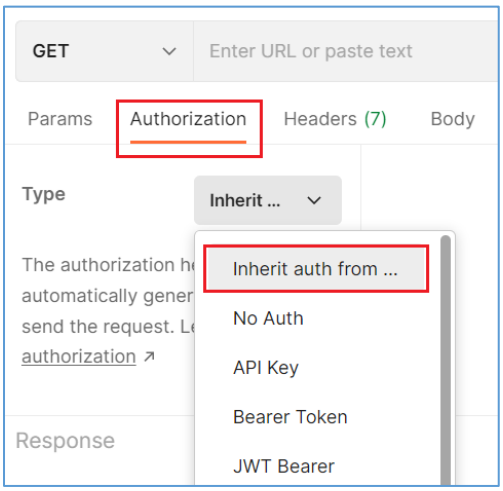
1. On the newly created workspace, click **Add a request**.



2. Enter the name of the request. For example, Part Request.



3. Select the **Authorization** tab and select Inherit auth from parent in the Type drop-down**.**

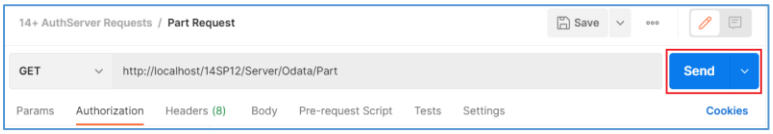


To confirm token authentication, run a query to get all the Part data from Aras Innovator.

4. Select the **GET** request and enter the following request URL.

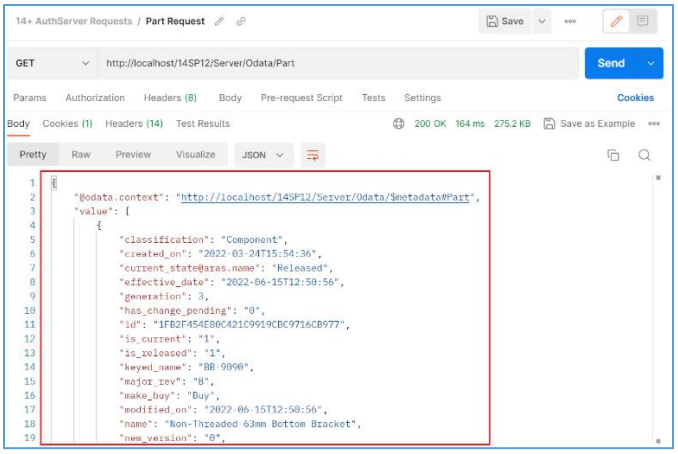
URL: http://<servername>/<web alias>/Server/Odata/Part

5. Click **Send** to run the request.



The request returns the JSON with all the parts within the database as follows:

©2026 Aras Corporation All Copyrights Reserved.



This confirms that the token authentication and Innovator server connection were successful.

©2026 Aras Corporation All Copyrights Reserved.