

Developer Course

Web Services

I300 Data Retrieval with OData 2022 R2

Revision: 9/21/2022

Contents

Copyright.....	4
Introduction.....	5
How to Use This Course.....	6
Course Prerequisites.....	7
Deploying an Acumatica ERP Instance for the Training Course.....	7
OData Interface.....	8
Company Story.....	9
Part 1: Client Application Configuration.....	10
Lesson 1.1: Signing In to Acumatica ERP.....	10
Example: Using OData Version 3.....	10
Example: Using OData Version 4.....	12
Additional Information: OAuth 2.0 Authorization.....	14
Additional Information: CORS Support for OData.....	14
Lesson Summary.....	15
Part 2: Initial Data Retrieval.....	16
Lesson 2.1: Retrieving the List of Customers with Contacts.....	16
Example: Using a Custom Generic Inquiry (OData Version 3).....	17
Example: Using GET and the \$expand and \$select Parameters (OData Version 4).....	19
Additional Information: Retrieval of Custom and User-Defined Fields.....	20
Additional Information: Viewing of the Exposed Generic Inquiry Through an OData Client.....	20
Lesson Summary.....	21
Lesson 2.2: Retrieving the Quantities of Stock Items.....	21
Example: Using a Generic Inquiry (OData Version 3).....	22
Example: Retrieving Quantities from Tables (OData Version 4).....	23
Lesson Summary.....	24
Part 3: Retrieval of the Delta of Records.....	25
Lesson 3.1: Retrieving the List of Modified Stock Items.....	25
Prerequisites.....	26
Example: Filtering the Result of a Generic Inquiry (OData Version 3).....	28
Example: Filtering the Requested Data (OData Version 4).....	31
Lesson Summary.....	33
Lesson 3.2: Monitoring Item Availability with Push Notifications.....	33
Example: Configuring Push Notifications.....	33
Additional Information: Push Notifications.....	36

Lesson Summary..... 37

Part 4: Creation of a Customization Package.....38

Lesson 4.1: Configuring a Customization Project and Exporting It.....38

Example: Creating a Customization Package for the OData Version 3 Integration..... 38

Example: Creating a Customization Package for the OData Version 4 Integration..... 40

Lesson Summary..... 40

Appendix: Comparison of the Integration Interfaces.....41

Appendix: Web Integration Scenario Reference..... 44

Copyright

© 2022 Acumatica, Inc.

ALL RIGHTS RESERVED.

No part of this document may be reproduced, copied, or transmitted without the express prior consent of Acumatica, Inc.

3933 Lake Washington Blvd NE, # 350, Kirkland, WA 98033

Restricted Rights

The product is provided with restricted rights. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in the applicable License and Services Agreement and in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable.

Disclaimer

Acumatica, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Acumatica, Inc. reserves the right to revise this document and make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Trademarks

Acumatica is a registered trademark of Acumatica, Inc. HubSpot is a registered trademark of HubSpot, Inc. Microsoft Exchange and Microsoft Exchange Server are registered trademarks of Microsoft Corporation. All other product names and services herein are trademarks or service marks of their respective companies.

Software Version: 2022 R2

Last Updated: 09/21/2022

Introduction

External systems can use Acumatica ERP integration interfaces to access the business functionality and data of Acumatica ERP. This course introduces the Open Data (OData) interface.

This course is intended for developers who need to create applications that interact with Acumatica ERP.

The course is based on a set of examples of web integration scenarios that demonstrate the processes of developing a client application that uses the Acumatica ERP OData interface. The course demonstrates the main elements of the OData interface and their use in typical tasks that integrate Acumatica ERP with third-party applications. This course contains the same integration scenarios as *I310 Data Retrieval with REST API Basics*, which introduces the contract-based REST API.

After you complete all the lessons of the course, you will be familiar with the basic techniques of data retrieval through the Acumatica ERP OData interface.

How to Use This Course

To complete the course, you will complete the lessons from each part of the course in the order in which they are presented and pass the assessment test. More specifically, you will do the following:

1. Complete *Course Prerequisites*, and carefully read *Initial Steps*.
2. Complete the lessons in all parts of the training guide.
3. In Partner University, take *I300 Certification Test: OData*.

After you pass the certification test, you will be given the Partner University certificate of course completion.

What Is in a Part

The first part of the course explains how to configure a third-party application to use the Acumatica ERP OData integration interface.

The second and third parts of the course are dedicated to the implementation of particular web integration scenarios that you may need to implement in a third-party application that integrates an external system with Acumatica ERP.

The fourth part of the course describes how to include in a customization project the items that you have created in Acumatica ERP for integration with an external system.

Each part of the course consists of lessons you should complete.

What Is in a Lesson

Each lesson is dedicated to a particular web integration scenario that you can implement by using the OData interface. Each lesson consists of a brief description of the web integration scenario and examples of the implementation of this scenario.

The lesson may also include *Additional Information* topics, which are outside of the scope of this course but may be useful to some readers.

Each lesson ends with a *Lesson Summary* topic, which summarizes the possible options for the implementation of the web integration scenario with different versions of the OData integration interface.

What the Documentation Resources Are

The complete Acumatica ERP and Acumatica Framework documentation is available at <https://help.acumatica.com/> and is included in the Acumatica ERP instance. While viewing any form used in the course, you can click the **Open Help** button in the top pane of the Acumatica ERP screen to bring up a form-specific Help menu; you can use the links on this menu to quickly access form-related information and activities and to open a reference topic with detailed descriptions of the form elements.

Which License You Should Use

For the educational purposes of this course, you use Acumatica ERP under the trial license, which does not require activation and provides all available features. For the production use of the Acumatica ERP functionality, an administrator has to activate the license the organization has purchased. Each particular feature may be subject to additional licensing; please consult the Acumatica ERP sales policy for details.

Course Prerequisites

To complete this course, you should be familiar with the basic principles of Acumatica ERP and of the creation of generic inquiries in Acumatica ERP. We recommend that you complete the *S130 Reporting: Data Retrieval and Analysis* training course before you go through this course.

You need to perform the prerequisite actions described in this part before you start to complete the course.

1. Make sure the environment that you are going to use conforms to the [System Requirements for Acumatica ERP 2022 R2](#).
2. Make sure that the Web Server (IIS) features that are listed in [Configuring Web Server \(IIS\) Features](#) are turned on.
3. Deploy an instance of Acumatica ERP 2022 R2 with the name *MyStoreInstance* and a tenant that contains the *T100* data. If you have completed the *I310 Data Retrieval with REST API Basics* training course, you can use the instance that you have deployed for this course. For information on how to deploy this instance, see [Deploying an Acumatica ERP Instance for the Training Course](#).
4. Make sure the Postman application is installed on your computer. To download and install Postman, follow the instructions on <https://www.postman.com/downloads/>.
5. Make sure you have HTTP access from the computer where you work with the examples to the Acumatica ERP instance.

Deploying an Acumatica ERP Instance for the Training Course

You deploy an Acumatica ERP instance and configure it as follows:

1. Open the Acumatica ERP Configuration Wizard, and deploy a new application instance as follows:
 - a. On the **Database Configuration** page of the Acumatica ERP Configuration Wizard, type the name of the database: *MyStoreInstance*.
 - b. On the **Tenant Setup** page, set up one tenant with the *T100* data inserted by specifying the following settings:
 - **Login Tenant Name:** *MyStore*
 - **New:** Selected
 - **Insert Data:** *T100*
 - **Parent Tenant ID:** 1
 - **Visible:** Selected

The system creates a new Acumatica ERP instance, adds a new tenant, and loads the selected data.

2. Sign in to the new tenant by using the following credentials:

- Login: *admin*
- Password: *setup*

Change the password when the system prompts you to do so.

3. Click the user name in the top right corner of the Acumatica ERP window, and click **My Profile**. On the **General Info** tab of the [User Profile](#) (SM203010) form, which opens, select *MYSTORE* in the **Default Branch** box; then click **Save** on the form toolbar. In subsequent sign-ins to this account, you will be signed in to this branch.

OData Interface

In Acumatica ERP, by using Open Data Protocol (OData) Version 3.0, you can expose the results of generic inquiries to external systems. By using the OData interface based on OData Version 4.0, you can request data from Acumatica ERP by directly accessing data access classes (DACs) without configuring any generic inquiry. In this course, both approaches are shown.

Any generic inquiry that has been created on the [Generic Inquiry](#) (SM208000) form can be exposed via OData Version 3.0. To expose the generic inquiry through OData, you select the **Expose via OData** check box for the inquiry on the [Generic Inquiry](#) form.



For details about the creation of generic inquiries, see the *S130 Reporting: Data Retrieval and Analysis* training course.

You can use the OData interface to retrieve the data and analyze it in an external application, such as a business intelligence (BI) application. You cannot edit the data in Acumatica ERP through the OData interface.

Related Links

- [OData Version 3.0 Specification](#)
- [OData Version 4.0 Specification](#)
- [Generic Inquiries and OData: General Information](#)

Company Story

In this course, you will simulate the integration of Acumatica ERP with a business intelligence (BI) application that is used by a small retail company, MyStore. This company is a single business entity that has no branches or subsidiaries. MyStore uses Acumatica ERP for customer management and inventory management.

MyStore plans to extend its business and needs analytic data about its customers and the goods in the store. MyStore needs to investigate the options available in Acumatica ERP for integration with BI applications. The integration application, which MyStore is developing, should retrieve information about customers and stock items from Acumatica ERP.

For the integration of a BI application with Acumatica ERP, the MyStore company can use the OData interface.

The following diagram shows the integration scheme of the MyStore BI application with Acumatica ERP.

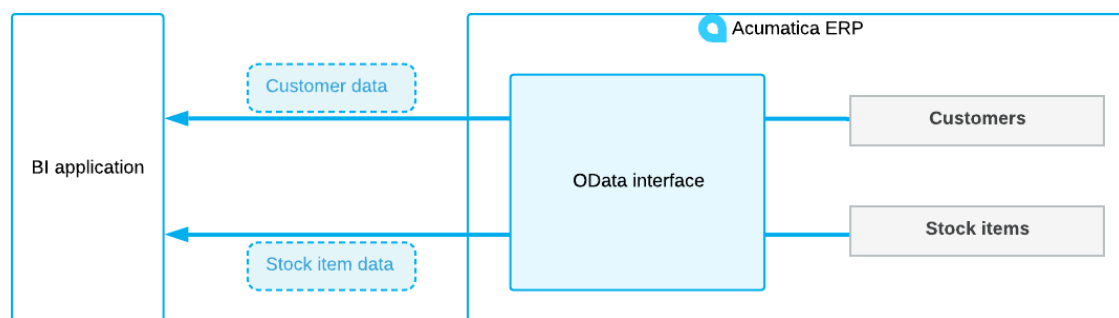


Figure: Integration of the MyStore BI application and Acumatica ERP

Integration Requirements

The BI application of the MyStore company will be used by the marketing managers and warehouse managers to analyze existing customers and review the item availability in warehouses.

This course examines the following usage scenarios in the BI application:

- Retrieval of the list of customers with contact and address details of each customer
- Retrieval of the information about availability of stock items in warehouses

In this course, you do not implement the BI application itself; instead, you implement the integration part between the BI application and Acumatica ERP, which provides the support for the listed scenarios in the BI application. (The implementation of the BI application is outside of the scope of this course.)

Part 1: Client Application Configuration

In this part of the guide, you will learn how to sign in to Acumatica ERP in the client application. You will also learn how to configure a client application to use the Acumatica ERP integration interfaces.

Lesson 1.1: Signing In to Acumatica ERP

Before you can retrieve data from Acumatica ERP by using the integration interfaces provided by Acumatica ERP, you need to sign in to Acumatica ERP. In this lesson, you will sign in by using basic authentication with a username and password.

You can also use the OAuth 2.0 authorization of applications, which is outside of the scope of this course. For a brief description, see [Additional Information: OAuth 2.0 Authorization](#).

Lesson Objectives

In this lesson, you will learn how to sign in to Acumatica ERP to use the OData interface.

Example: Using OData Version 3

The OData Version 3.0 protocol uses the basic authentication in Acumatica ERP. That is, you need to sign in with a username and password to Acumatica ERP before you request the data through the OData protocol.

In this example, you will configure a Postman collection to use the basic authentication and retrieve the list of generic inquiries available through OData to test the sign-in. If you want to use the browser to test the OData request, you can skip the configuration of a Postman collection, and enter your username and password in the browser when you are asked to do so. When the session expires, you have to sign in once again.

A sign-in through the OData Version 3.0 protocol is treated by the system as a sign-in of a conventional user (not an API user). The license restriction for conventional users is shown in the **Concurrent Users** box on the **License** tab of the [License Monitoring Console](#) (SM604000) form.

Testing the Sign-In to Acumatica ERP (OData Version 3)

To configure basic authorization in Postman and test it, do the following:



Instead of configuring the requests by yourself, you can import the Postman collection from the `OData.postman_collection.json` file provided with this training course and test the corresponding requests from this collection. (You can find the file in <https://github.com/Acumatica/Help-and-Training-Examples/tree/2022R2/IntegrationDevelopment/I300>.)

1. In Postman, create a new collection and, on the **Authorization** tab, configure its authorization settings as follows:
 - a. In the **Type** box, select the *Basic Auth* type.
 - b. In the **Username** and **Password** boxes, type the username and password that you are using to access the Acumatica ERP instance for the training course.
 - c. Click **Save**.
2. In the collection, add a **GET** request to the following URL to retrieve the list of generic inquiries exposed through OData.

```
http://localhost/MyStoreInstance/OData
```



The Acumatica ERP instance that you are using for the training course has only one tenant configured. Therefore, you should not specify the tenant name in the URL. If the instance that you are accessing through the OData protocol has multiple tenants configured, you have to specify the tenant name in the URL. For example, if you need to retrieve the list of generic inquiries exposed through OData in the *MyTenant* tenant in the *MyStoreInstance* instance, you would use the following URL.

```
http://localhost/MyStoreInstance/OData/MyTenant
```

3. Send the request. The response of the successful request contains the 200 OK status code. The following code example shows the response body.

```
<?xml version="1.0" encoding="utf-8"?>
<service xml:base="http://localhost/MyStoreInstance/odata"
  xmlns="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom">
  <workspace>
    <atom:title type="text">Default</atom:title>
    <collection href="BI-LeadConversion">
      <atom:title type="text">BI-LeadConversion</atom:title>
    </collection>
    <collection href="BI-Cases">
      <atom:title type="text">BI-Cases</atom:title>
    </collection>
    <collection href="DB-StorageDetails">
      <atom:title type="text">DB-StorageDetails</atom:title>
    </collection>
    <collection href="PM-Project%20Transactions%20Inquiry">
      <atom:title type="text">PM-Project Transactions Inquiry</atom:title>
    </collection>
    <collection href="BI-ARInvoices">
      <atom:title type="text">BI-ARInvoices</atom:title>
    </collection>
    <collection href="BI-GLBudgetActual">
      <atom:title type="text">BI-GLBudgetActual</atom:title>
    </collection>
    <collection href="BI-Employees">
      <atom:title type="text">BI-Employees</atom:title>
    </collection>
    <collection href="CR-CampaignSalesOrders">
      <atom:title type="text">CR-CampaignSalesOrders</atom:title>
    </collection>
    <collection href="PM-Project%20Summary">
      <atom:title type="text">PM-Project Summary</atom:title>
    </collection>
    <collection href="DB-StorageDetailsByItemWarehouseLocation">
      <atom:title type="text">DB-StorageDetailsByItemWarehouseLocation</
atom:title>
    </collection>
    <collection href="PM-Revised%20Contract%20vs%20Approved%20CO">
      <atom:title type="text">PM-Revised Contract vs Approved CO</atom:title>
    </collection>
    <collection href="BI-OppStages">
      <atom:title type="text">BI-OppStages</atom:title>
    </collection>
```

```

<collection href="BI-Dates">
  <atom:title type="text">BI-Dates</atom:title>
</collection>
<collection href="BC-PUSH-AvailabilityStockItem">
  <atom:title type="text">BC-PUSH-AvailabilityStockItem</atom:title>
</collection>
<collection href="BI-Opportunities">
  <atom:title type="text">BI-Opportunities</atom:title>
</collection>
<collection href="CR-CampaignInvoices">
  <atom:title type="text">CR-CampaignInvoices</atom:title>
</collection>
<collection href="BI-Customers">
  <atom:title type="text">BI-Customers</atom:title>
</collection>
<collection href="BI-Leads">
  <atom:title type="text">BI-Leads</atom:title>
</collection>
<collection href="BI-BranchCompany">
  <atom:title type="text">BI-BranchCompany</atom:title>
</collection>
<collection href="PM-Vendor%20Commitment%20Details">
  <atom:title type="text">PM-Vendor Commitment Details</atom:title>
</collection>
<collection href="BC-PUSH-AvailabilityTemplates">
  <atom:title type="text">BC-PUSH-AvailabilityTemplates</atom:title>
</collection>
<collection href="Item%20Availability%20Data">
  <atom:title type="text">Item Availability Data</atom:title>
</collection>
<collection href="Modified%20Stock%20Items">
  <atom:title type="text">Modified Stock Items</atom:title>
</collection>
<collection href="Stock%20Item%20Attachments">
  <atom:title type="text">Stock Item Attachments</atom:title>
</collection>
<collection href="Customer%20Contacts">
  <atom:title type="text">Customer Contacts</atom:title>
</collection>
</workspace>
</service>

```

4. Save the request.

Related Links

- [Generic Inquiries and OData: General Information](#)

Example: Using OData Version 4

The OData Version 4.0 protocol uses the basic authentication in Acumatica ERP in the same way as the OData Version 3.0 protocol does.

In this example, to test the sign-in, you will retrieve the full list of data access classes (DACs), their fields, the fields' types, and the navigation properties of DACs, which are relationships between the DACs. You can use the Postman collection that you created in [Example: Using OData Version 3](#). If you want to use the browser to test the OData

request, you can enter your username and password in the browser when you are asked to do so. When the session expires, you have to sign in once again.

Testing the Sign-In to Acumatica ERP (OData Version 4)

To retrieve the list of DACs with details, do the following:

1. Create a GET request to the following URL.

```
http://localhost/MyStoreInstance/ODataV4/$metadata
```



The Acumatica ERP instance that you are using for the training course has only one tenant configured. Therefore, you should not specify the tenant name in the URL. If the instance that you are accessing through the OData protocol has multiple tenants configured, you have to specify the tenant name in the URL. For example, if you needed to retrieve the list of DACs in the *MyTenant* tenant in the *MyStoreInstance* instance, you would use the following URL.

```
http://localhost/MyStoreInstance/ODataV4/MyTenant/$metadata
```

2. Send the request. The response of the successful request contains the 200 OK status code. The following code example shows the response body.

```
<?xml version="1.0" encoding="utf-8"?>
<edmx:Edmx Version="4.0"
  xmlns:edmx="http://docs.oasis-open.org/odata/ns/edmx">
  <edmx:DataServices>
    <Schema Namespace="PX.Api"
      xmlns="http://docs.oasis-open.org/odata/ns/edm">
      <EntityType Name="SYMapping">
        <Key>
          <PropertyRef Name="Name" />
        </Key>
        <Property Name="IsSimpleMapping" Type="Edm.Boolean" />
        <Property Name="MappingID" Type="Edm.Guid" />
        <Property Name="IsActive" Type="Edm.Boolean" Nullable="false" />
        <Property Name="Name" Type="Edm.String" />
        ...
        <NavigationProperty Name="SYProviderObjectByProviderID"
          Type="PX.Api.SYProviderObject">
          <ReferentialConstraint Property="ProviderObject"
            ReferencedProperty="Name" />
          <ReferentialConstraint Property="ProviderID"
            ReferencedProperty="ProviderID" />
        </NavigationProperty>
        <NavigationProperty Name="SYMappingFieldCollection"
          Type="Collection(PX.Api.SYMappingField)" />
        ...
      </EntityType>
      ...
    </Schema>
    ...
    <Schema Namespace="PX.Objects.CA"
      xmlns="http://docs.oasis-open.org/odata/ns/edm">
      ...
      <ComplexType Name="CASetup">
        <Property Name="BatchNumberingID" Type="Edm.String" />
      </ComplexType>
    </Schema>
  </DataServices>
</Edmx>
```

```

...
<NavigationProperty Name="UsersByCreatedByID" Type="PX.SM.Users">
  <ReferentialConstraint Property="CreatedByID"
    ReferencedProperty="PKID" />
</NavigationProperty>
...
</ComplexType>
...
</Schema>
...
<Schema Namespace="Default"
  xmlns="http://docs.oasis-open.org/odata/ns/edm">
  <EntityContainer Name="Container">
    <EntitySet Name="PX_Api_SYMapping" EntityType="PX.Api.SYMapping">
      <NavigationPropertyBinding Path="HSEntitySetupCollection"
        Target="PX_DataSync_HubSpot_HSEntitySetup" />
      ...
    </EntitySet>
    ...
    <EntitySet
      Name="ReconciliationTools_ARGLDiscrepancyByDocumentEnqResult"
      EntityType="ReconciliationTools.ARGLDiscrepancyByDocumentEnqResult"
    />
    <EntitySet Name="ReconciliationTools_DiscrepancyByAccountEnqResult"
      EntityType="ReconciliationTools.DiscrepancyByAccountEnqResult" />
    </EntityContainer>
  </Schema>
</edmx:DataServices>
</edmx:Edmx>

```

Additional Information: OAuth 2.0 Authorization

Acumatica ERP supports the OAuth 2.0 mechanism of authorization for applications that are integrated with Acumatica ERP through an application programming interface (API) or OData interface. When a client application of Acumatica ERP uses the OAuth 2.0 mechanism of authorization, the client application does not operate with the Acumatica ERP credentials to sign in a user to Acumatica ERP; instead, the application obtains an access token from Acumatica ERP and uses this token when it requests data from Acumatica ERP.

For details about the OAuth 2.0 authorization mechanism, see [Authorizing Client Applications to Work with Acumatica ERP](#). The authorization of a client application to work with Acumatica ERP is also illustrated in the *I320 Advanced Data Retrieval with REST API* training course.

Additional Information: CORS Support for OData

Acumatica ERP supports cross-origin resource sharing (CORS), meaning that requests for resources can come from a different domain than that of the resource making the request. With CORS enabled, you can allow access to the OData interface of your Acumatica ERP instance for the client applications.

For more information about the CORS support, see *Configuring CORS* in [Exposing an Inquiry by Using OData](#).

For details about how to configure CORS for your instance of Acumatica ERP, see [Accessing the Exposed Inquiry Through OData: To Configure CORS](#).

Lesson Summary

In this lesson, you have learned how an application that uses the OData protocol can sign in to Acumatica ERP. For information on how to sign in to and sign out from Acumatica ERP by using the contract-based REST API, see *Lesson 1.1: Signing In to and Signing Out from Acumatica ERP* in the *I310 Data Retrieval with REST API Basics* course.

You have also reviewed OAuth 2.0 authorization in Acumatica ERP.

Part 2: Initial Data Retrieval

In the BI application of the MyStore company, a marketing manager should be able to view analytics for the distribution of existing customers by geographical state. To display to the marketing manager the information about the customers, you need to retrieve from Acumatica ERP the information on the customers, their contacts, and their addresses.

Also, in the BI application of the MyStore company, a warehouse manager should be able to view the statistics of the items available in the warehouse. To make it possible for the BI application to display these statistics, you need to retrieve from Acumatica ERP the information about the quantities of stock items available in the warehouses.

By completing the examples of this part, you will retrieve the full list of customer records with contacts and addresses. (The import of this list to the BI application is outside of the scope of this course.) You will also retrieve the quantities of stock items.

Lesson 2.1: Retrieving the List of Customers with Contacts

The BI application of the MyStore company should display to a user the distribution of existing customers by geographic state. The data of these customers has been entered on the [Customers](#) (AR303000) form in Acumatica ERP. To provide the list of customers to the BI application, you should retrieve the following values from Acumatica ERP (with the corresponding locations on the [Customers](#) form):

- Customer ID (the **Customer ID** box of the Summary area)
- Account name (the **Account Name** box of the **Account Info** section on the **General** tab)
- Customer class (the **Customer Class** box of the Summary area)
- Details of the main contact of the customer (the **Additional Account Info** section of the **General** tab):
 - Email address (the **Account Email** box)
 - Primary phone number (the **Business 1** box)
- Customer address (the **Account Address** section of the **General** tab):
 - City (the **City** box)
 - State (the **State** box)
 - Postal code (the **Postal Code** box)
 - Address line (the **Address Line 1** and **Address Line 2** boxes)

These elements are shown in the following screenshot.

Customers
C000000001 - Jersey Central Office Equip

VIEW ACCOUNT ...

* Customer ID: C000000001
* Customer Status: Active
* Customer Class: DEFAULT - Default

Balance: 730.00
Prepayment Balance: 0.00

GENERAL FINANCIAL BILLING SHIPPING PAYMENT METHODS CONTACTS SALESPERSONS ATTRIBUTES

ACCOUNT INFO
* Account Name: Jersey Central Office Equip

ACCOUNT ADDRESS
VIEW ON MAP
Address Line 1: 1 De Villiers & Harrison St, 11-th Flr.
Address Line 2:
City: Johannesburg
State:
Postal Code:
* Country: ZA - South Africa

ADDITIONAL ACCOUNT INFO
Business 1: +1 (777) 283-0414
Business 2:
Fax:
Account Email: jersey-equip@mail.con
Web:

PRIMARY CONTACT
Name: First Name Last Name
Job Title:
Email:
Business 1
Cell:

CONTACT PERSONAL DATA PRIVACY
☐ Consented to the Processing of Personal Data
Date of Consent:
Consent Expires:

Figure: Elements on the Customers form

This lesson shows how you can implement this scenario by using the OData interface. To export this data by using the OData Version 3.0 protocol, you need to create a generic inquiry. Through the OData Version 4.0 protocol, you can export this data from Acumatica ERP without any additional preparations.

Lesson Objective

In this lesson, you will learn how to retrieve the data from an Acumatica ERP data entry form.

Example: Using a Custom Generic Inquiry (OData Version 3)

In this example, through the OData protocol, you will export the list of customer records with contacts.

Through the OData Version 3.0 protocol, you cannot export records directly from a data entry form, such as the [Customers](#) (AR303000) form. Thus, before the export, you have to configure a generic inquiry that retrieves the needed data from Acumatica ERP.

In this example, you will use the Customer Contacts (ARGI0015) custom generic inquiry, which has been preconfigured for this training course. This generic inquiry has no parameters and is based on the `PX.Objects.AR.Customer`, `PX.Objects.CR.Address`, and `PX.Objects.CR.Contact` data access classes. You can view this generic inquiry on the [Generic Inquiry](#) (SM208000) form by selecting the inquiry with the title *Customer Contacts* in the Summary area and clicking **View Inquiry** on the form toolbar.



Creating a generic inquiry is outside of the scope of this course. You can find information on how to create generic inquiries in the S130 Reporting: Data Retrieval and Analysis training course, which is available in Partner University.

In the training data, the **Expose via OData** check box is selected for this generic inquiry in the Summary area of the Generic Inquiry form. This means that the generic inquiry is available through the OData interface.

You will obtain the results of the generic inquiry in JSON format by using the `$format` parameter of the request.

Prerequisites

Before you send the request in the example, you need to sign in to Acumatica ERP, as described in [Example: Using OData Version 3](#). If you have created a Postman collection with the basic authentication configured, add the request described below to the collection and configure the request to inherit the authorization type from the parent collection.

Retrieving the List of Customers

To retrieve the list of customers with contacts, do the following:

1. Configure the following settings of the request:
 - HTTP method: GET
 - URL: `http://localhost/MyStoreInstance/OData/Customer%20Contacts`
 - Parameter: `$format=json`
2. Send the request. If the request is successful, its response contains the 200 OK status code. The following code example shows a fragment of the response body.

```
{
  "odata.metadata":
    "http://localhost/MyStoreInstance/odata/$metadata#Customer%20Contacts",
  "value":
    [
      {
        "CustomerID": "C000000001",
        "CustomerName": "Jersey Central Office Equip",
        "CustomerClass": "DEFAULT",
        "Email": "jersey-equip@mail.con",
        "Phone1": "+1 (777) 283-0414",
        "City": "Johannesburg",
        "State": null,
        "PostalCode": null,
        "AddressLine1": "1 De Villiers & Harrison St, 11-th Flr.",
        "AddressLine2": null,
        "ContactID": 5764,
        "AddressID": 5759
      },
      ...
    ]
}
```

3. Save the request.

Related Links

- [Generic Inquiries and OData: General Information](#)

Example: Using GET and the \$expand and \$select Parameters (OData Version 4)

In this example, through the OData interface based on the OData Version 4.0 protocol, you will request the list of customer records with contacts.

Through the OData Version 4.0 protocol, you request records directly from data access classes (DACs), such as `PX.Objects.AR.Customer`. You do not need to create a generic inquiry.

Prerequisites

Before you send the request in the example, you need to sign in to Acumatica ERP, as described in [Example: Using OData Version 3](#). If you have created a Postman collection with the basic authentication configured, add the request described below to the collection and configure the request to inherit the authorization type from the parent collection.

The `PX.Objects.AR.Customer` entity contains common information about a customer. The `ContactByDefBillContactID` navigation property of the `PX.Objects.AR.Customer` entity refers to the contact information contained in the `PX.Objects.CR.Contact` entity. Also, the `AddressByDefAddressID` navigation property of the `PX.Objects.AR.Customer` entity refers to the address contained in the `PX.Objects.CR.Address` entity. In the request provided in this topic, you will select the following fields of the entities:

- The `AcctCD`, `AcctName`, and `CustomerClassID` fields of the `PX.Objects.AR.Customer` entity
- The `Email` and `Phone1` fields of the `PX.Objects.CR.Contact` entity
- The `AddressLine1`, `AddressLine2`, `City`, `State`, and `PostalCode` fields of the `PX.Objects.CR.Address` entity

In the `$select` parameter, you list every field you want to provide in the result. In the `$expand` parameter, you specify every nested entity you want to provide in the result. For nested entities, you specify their `$select` parameters (and other parameters used as well) in parentheses.

Retrieving the List of Customers

To retrieve the list of customers with contacts, do the following:

1. Specify the following settings in the request:
 - HTTP method: `GET`
 - URL: `http://localhost/MyStoreInstance/ODataV4/PX_Objects_AR_Customer`
 - The parameters in the following table

Parameter	Value
<code>\$select</code>	<code>AcctCD, AcctName, CustomerClassID</code>
<code>\$expand</code>	<code>ContactByDefBillContactID(\$select=Email, Phone1), AddressByDefAddressID(\$select=AddressLine1, AddressLine2, City, State, PostalCode)</code>

2. Send the request. If the request is successful, its response contains the 200 OK status code. The following code example shows a fragment of the response body.

```
{
```

```

"@odata.context": "http://localhost/MyStoreInstance/odatav4/
$metadata#PX_Objects_AR_Customer
(AcctCD,AcctName,CustomerClassID,ContactByDefBillContactID (EMail,Phone1),
AddressByDefAddressID(AddressLine1,AddressLine2,City,State,PostalCode)) ",
"value": [
  {
    "AcctCD": "C000000001",
    "AcctName": "Jersey Central Office Equip",
    "CustomerClassID": "DEFAULT",
    "ContactByDefBillContactID": {
      "EMail": "jersey-equip@mail.con",
      "Phone1": "+1 (777) 283-0414"
    },
    "AddressByDefAddressID": {
      "AddressLine1": "1 De Villiers & Harrison St, 11-th Flr.",
      "AddressLine2": null,
      "City": "Johannesburg",
      "State": null,
      "PostalCode": null
    }
  },
  ...
]
}

```

3. Save the request.

Additional Information: Retrieval of Custom and User-Defined Fields

In a customization project, you can add custom fields to Acumatica ERP forms. You can also add user-defined fields to Acumatica ERP forms and include them in a customization project. (For details about user-defined fields, see [User-Defined Fields](#).)

You can retrieve the values of custom and user-defined fields through the OData interface. The retrieval of custom and user-defined fields is outside of the scope of this course but may be useful to some readers.

Retrieval of Custom and User-Defined Fields Through OData Version 3

To retrieve custom and user-defined fields through OData Version 3.0, you need to add these fields to the results of a generic inquiry exposed via OData. The name of a user-defined field is `Attribute<AttributeID>`, where you replace `<AttributeID>` with the ID of the attribute that corresponds to the user-defined field.

Retrieval of Custom and User-Defined Fields Through OData Version 4

To retrieve custom and user-defined fields through OData Version 4.0, you do not need to perform special actions. You address custom and user-defined fields by their names.

Additional Information: Viewing of the Exposed Generic Inquiry Through an OData Client

You can view the data exposed through the OData interface by using external OData clients, such as Microsoft Excel and Microsoft Power BI.

For details about viewing the data with Microsoft Excel, see [Access to an Inquiry Through OData: To Access an Exposed Inquiry in Microsoft Excel](#).

For details about viewing the data with Microsoft Power BI, see [Access to Inquiry Through OData: Connecting to Acumatica ERP from Power BI](#).

Lesson Summary

In this lesson, you have learned how to retrieve the data that is exposed in Acumatica ERP on a data entry form by using different versions of the OData interface. For information on how to retrieve the same data by using the contract-based REST API, see *Lesson 2.1: Retrieving the List of Customers with Contacts* in the *I310 Data Retrieval with REST API Basics* course.

Lesson 2.2: Retrieving the Quantities of Stock Items

The BI application of the MyStore company should display to a user the statistics of the items available in the warehouse. In Acumatica ERP, you can view the item availability data on the [Inventory Summary](#) (IN401000) form. In this lesson, you will use the method that exports data from the Item Availability Data (INGI0002) generic inquiry, which has been preconfigured for this training course. This generic inquiry has been created based on the `PX.Objects.IN.INSiteStatus` data access class and has one parameter, which you can use to filter the list of stock items by inventory ID. You can view this generic inquiry on the [Generic Inquiry](#) (SM208000) form by selecting the inquiry with the title *Item Availability Data* and clicking **View Inquiry**. The Item Availability Data generic inquiry is shown in the following screenshot.

Item Availability Data ☆ CUSTOMIZATION ▾ TOOLS ▾

🔄 ↶ ⏮ ⏭

Inventory ID:

Drag column header here to configure filter 🔍

Warehouse	Inventory ID	Description	Qty. On Hand	Qty. Available
> MAIN	AACOMPUT01	Acer Laptop Computer	150.00	150.00
MAIN	AALEGO500	Lego 500 piece set	1,999.00	1,977.00
MAIN	AAMACHINE1	Injection molding machine	130.00	127.00
MAIN	CONGRILL	Char-Broil Classic 480	200.00	193.00
MAIN	CONTABLE1	Folding Picnic Table 6 Foot	200.00	195.00
MAIN	HEADSET	Headset	40.00	40.00
MAIN	KEYBOARD	Computer keyboard	40.00	40.00
MAIN	LAPTOP12	Laptop 12"	20.00	20.00
MAIN	LAPTOP14	Laptop 14"	20.00	20.00
MAIN	LEATHCASE	Leather case for laptops 12"-15"	40.00	40.00
MAIN	MONITOR26	Monitor 26"	40.00	40.00
MAIN	MOUSE	Computer mouse	40.00	40.00
YOGI	SIMCARD	SIM card & contract	1,000.00	1,000.00

1-13 of 13 records ⏪ ⏩ ⏴ ⏵

Figure: Item Availability Data generic inquiry

To create an OData Version 4.0 request, you will use data from the `PX.Objects.IN.INSiteStatus` DAC and its navigation properties. You will do this to demonstrate that this task can be done using OData Version 4.0.

Lesson Objective

In this lesson, you will learn how to retrieve the list of records from a generic inquiry form.

Example: Using a Generic Inquiry (OData Version 3)

In this example, through OData, you will export the quantities of stock items from the Item Availability Data (INGI0002) custom generic inquiry in Acumatica ERP.

In the training data, the **Expose via OData** check box is selected for this generic inquiry in the Summary area of the [Generic Inquiry](#) (SM208000) form. This means that the generic inquiry is available through the OData interface.

In this example, you need to retrieve the full list of records of the generic inquiry. Although this generic inquiry has a parameter (which is optional), you cannot specify the parameter value of the inquiry in the OData request. The OData request returns the records for all possible variants of the parameters.

You will configure the request to return the data in JSON format by using the `$format` parameter.

Prerequisites

Before you send the request in the example, you need to sign in to Acumatica ERP, as described in [Example: Using OData Version 3](#). If you have created a Postman collection with the basic authentication configured, add the request described below to the collection and configure the request to inherit the authorization type from the parent collection.

Instructions for Retrieving the Quantities of Items

To retrieve the quantities of items, do the following:

1. Configure the following settings of the request:
 - HTTP method: `GET`
 - URL: `http://localhost/MyStoreInstance/OData/Item%20Availability%20Data`
 - Parameter: `$format=json`
2. Send the request. If the request is successful, the response contains the 200 OK status code. A fragment of the result is shown in the following code.

```
{
  "odata.metadata":
    "http://localhost/MyStoreInstance/odata/$metadata#Item%20Availability%20Data",
  "value":
    [
      {
        "InventoryID": "SIMCARD  ",
        "Warehouse": "YOGI      ",
        "Description": "SIM card & contract",
        "QtyOnHand": "1000.000000",
        "QtyAvailable": "1000.000000",
        "Subitem": "0"
      },
      {
        "InventoryID": "AACOMPUT01",
        "Warehouse": "MAIN      ",
```

```

        "Description": "Acer Laptop Computer",
        "QtyOnHand": "150.000000",
        "QtyAvailable": "150.000000",
        "Subitem": "0"
    },
    ...
]
}

```

3. Save the request.

Related Links

- [Generic Inquiries and OData: General Information](#)

Example: Retrieving Quantities from Tables (OData Version 4)

In OData Version 4.0 requests, you can specify the data access classes (DACs) from which data is retrieved. You cannot specify generic inquiries in OData Version 4.0 requests. In particular, to get the data provided by the Item Availability Data (INGI0002) generic inquiry, in an OData Version 4.0 request, you specify DACs on which the generic inquiry is based.

The following request retrieves the quantities of items:

- HTTP method: GET
- URL: *http://localhost/MyStoreInstance/ODataV4/PX_Objects_IN_INSiteStatus*
- Parameters: The parameters shown in the following table

Key	Value
\$select	QtyOnHand, QtyAvail
\$expand	InventoryItemByInventoryID(\$select=InventoryCD, Descr) , INSiteBySiteID(\$select=SiteCD)

Following is an example of the response.

```

{
  "@odata.context":
    "http://localhost/MyStoreInstance/odatav4/$metadata#PX_Objects_IN_INSiteStatus
    (QtyOnHand,QtyAvail,InventoryItemByInventoryID(InventoryCD,Descr),
    INSiteBySiteID(SiteCD))",
  "value": [
    {
      "QtyOnHand": 1000,
      "QtyAvail": 1000,
      "InventoryItemByInventoryID": {
        "InventoryCD": "SIMCARD  ",
        "Descr": "SIM card & contract"
      },
      "INSiteBySiteID": {
        "SiteCD": "YOGI      "
      }
    }
  ],
}

```

```
    ...  
  ]  
}
```

Lesson Summary

In this lesson, you have learned how to retrieve a list of records that is exposed in Acumatica ERP on a generic inquiry form by using different versions of the OData interface. For information on how to retrieve the same data by using the contract-based REST API, see *Lesson 2.2: Retrieving the Quantities of Stock Items* in the *I310 Data Retrieval with REST API Basics* course.

Part 3: Retrieval of the Delta of Records

The BI application of the MyStore company displays information about the stock items registered in Acumatica ERP. To provide actual data about the stock items to the BI system, you need to retrieve this information once a day. To optimize the performance, you retrieve only the data about the items that have been modified within the past day.

The BI application also needs to display information about item availability in Acumatica ERP in real time. To display actual data about the stock item availability, you use push notifications configured in Acumatica ERP.

By completing the examples of this part, you will retrieve only the modified records from Acumatica ERP. You can use both OData Version 3.0 and OData Version 4.0 for the export. You will also configure push notifications to be sent by Acumatica ERP when item availability data has been changed.

Lesson 3.1: Retrieving the List of Modified Stock Items

The BI application of the MyStore company should display to each user the information about the items that are sold in the store. These items are entered and updated on the [Stock Items](#) (IN202500) form in Acumatica ERP. When you already have a list of items with the necessary information, you need to retrieve only the changes to the items that have been made during the past day. To display the list of modified items to a user, you should export the list of stock items that satisfy the specified conditions from Acumatica ERP. You will export the stock item records that have the *Active* status and that were modified within the past day.

For a stock item record entered and maintained on the [Stock Items](#) form, you will export the following values:

- The inventory ID (the **Inventory ID** box in the Summary area)
- The description of the item (**Description** of the Summary area)
- The item class assigned to the item in Acumatica ERP (the **Item Class** box of the **Item Defaults** section of the **General** tab)
- The base unit of measurement (the **Base Unit** box of the **Unit of Measure** section of the **General** tab)
- The date and time the record was last modified (for which there are no corresponding elements on the form)
- Information about the availability of the item in particular warehouses (the **Warehouse Details** tab):
 - The warehouse ID (the **Warehouse** column)
 - The quantity of the item available in the warehouse (the **Qty. On Hand** column)

The elements that are available on the form are shown in the following screenshots. The value of the last modified date and time will be exported from an internal field, which is not available among the elements of the form.

Stock Items

AALEGO500 - Lego 500 pieces set

←

📄

📁

↶

+

🗑️

📋

⏪

⏩

⏴

⏵

⋮

* Inventory ID:

AALEGO500 - Lego 500 pieces set

Product Workgroup:

Item Status:

Active

Product Manager:

Description:

Lego 500 pieces set

GENERAL

PRICE/COST

WAREHOUSES

VENDORS

ATTRIBUTES

PACKAGING

CROSS-REFERENCE

ITEM DEFAULTS

UNIT OF MEASURE

* Item Class:

STOCKITEM - Stock item

Type:

Finished Good

Valuation Method:

Average

* Tax Category:

EXEMPT - Exempt

* Posting Class:

STOCKITEM - Stock item

Auto-Incremental Value:

Country Of Origin:

Base Unit:

PIECE

* Sales Unit:

PIECE

* Purchase Unit:

PIECE

☐ Weight Item

WAREHOUSE DEFAULTS

Default Warehouse:

MAIN

⏻

+

×

* From Unit	Multiply/Divide	Conversion Factor	To Unit
-------------	-----------------	-------------------	---------

Figure: Elements whose values will be exported from the Summary area and the General tab

Stock Items

AALEGO500 - Lego 500 pieces set

Inventory ID: AALEGO500 - Lego 500 pieces set Product Workgroup:

Item Status: Active Product Manager:

Description: Lego 500 pieces set

GENERAL PRICE/COST **WAREHOUSES** VENDORS ATTRIBUTES PACKAGING CROSS-REFERENCE GL ACCOUNTS

ADD WAREHOUSE DETAIL

	Default	Warehouse	Status	Inventory Account	Inventory Sub.	Override Product Manager	Product Workgroup	Product Manager	Overri Std. Cost	Price Overri	Qty. On Hand
>	<input checked="" type="checkbox"/>	MAIN	Active	130000	000-00-00						1,999.00

Figure: Elements whose values will be exported from the Warehouse Details tab

This lesson shows how you can implement this scenario by using the OData interface.

Lesson Objective

In this lesson, you will learn how to retrieve the list of records that were modified within a period.

Prerequisites

To have at least one stock item record modified within the past day, you need to perform the actions described in this topic before you perform the examples of this lesson.

On the [Stock Items](#) (IN202500) form, modify the *KEYBOARD* and *AALEGO500* inventory items as follows:

- Change the status of the *KEYBOARD* inventory item to *Inactive* and save the record.
- Change the description of the *AALEGO500* inventory item to *Lego, 500 pieces*, and save the record.

Now you have at least two inventory items that were modified within the past month, and one of them has *Active* status.

The system tracks the last modified date for every record, but this date is not displayed on the [Stock Items](#) form. In the system, there is a preconfigured generic inquiry where you can check the dates when stock items were last modified. To view this generic inquiry, on the [Generic Inquiry](#) (SM208000) form, select the inquiry with the title *Stock Items: Last Modified Date*, and click **View Inquiry** on the form toolbar. You can sort items by the last modified date on the Last Modified Date (INGI0001) generic inquiry, as shown in the following screenshot.

Last Modified Date



Drag column header here to configure filter						
			Inventory ID	Stock Item	Item Status	Last Modified On
>			<N/A>	<input type="checkbox"/>		1/1/1900
			AACOMPUT01	<input checked="" type="checkbox"/>	Inactive	9/21/2021
			AALEGO500	<input checked="" type="checkbox"/>	Active	9/21/2021
			AAMACHINE1	<input checked="" type="checkbox"/>	Active	1/1/1900
			ADVERT	<input type="checkbox"/>	Active	1/1/1900
			CALLIN	<input type="checkbox"/>	Active	1/1/1900
			CALLOUT	<input type="checkbox"/>	Active	1/1/1900
			CONGRILL	<input checked="" type="checkbox"/>	Active	1/1/1900
			CONSULT	<input type="checkbox"/>	Active	1/1/1900
			CONSULTJR	<input type="checkbox"/>	Active	1/1/1900
			CONTABLE1	<input checked="" type="checkbox"/>	Active	1/1/1900
			CSTMZATION	<input type="checkbox"/>	Active	1/1/1900
			HEADSET	<input checked="" type="checkbox"/>	Active	1/1/1900
			KEYBOARD	<input checked="" type="checkbox"/>	Inactive	9/21/2021
			LAPTOP12	<input checked="" type="checkbox"/>	Active	1/1/1900
1-15 of 22 records						

Figure: Last Modified Date generic inquiry

Example: Filtering the Result of a Generic Inquiry (OData Version 3)

In this example, through the OData Version 3.0 protocol, you will export the list of active stock item records that were modified within the past day.

Through the OData protocol, you cannot export records directly from a data entry form, such as the [Stock Items](#) (IN202500) form. Thus, before the export, you have to configure a generic inquiry that retrieves the needed data from Acumatica ERP.

In this example, you will use the Modified Stock Items (INGI0016) custom generic inquiry, which has been preconfigured for this training course. This generic inquiry has no parameters and is based on the `PX.Objects.IN.INSiteStatus` and `PX.Objects.IN.InventoryItem` data access classes. You can view this generic inquiry on the [Generic Inquiry](#) (SM208000) form by selecting the inquiry with the title *Modified Stock Items* in the Summary area and clicking **View Inquiry** on the form toolbar. This generic inquiry contains all the fields that should be retrieved. (Creation of generic inquiries is outside of the scope of this course.)

In the training data, the **Expose via OData** check box is selected for this generic inquiry in the Summary area of the [Generic Inquiry](#) form. This means that the generic inquiry is available through the OData interface.

Because you need to filter the results of the inquiry to obtain only the active records that were modified within the past day, you need to use the `$filter` parameter. Before you configure the filter, you will find out the names of the fields that are available in the generic inquiry by using a specific request to the OData interface.

You will use the `$filter` parameter to specify the search conditions for the fields. You use [OData URI conventions](#) to specify the conditions.

You will obtain the data in JSON format by using the `$format` parameter of the request.

Prerequisites

Before you send the request in the example, you need to sign in to Acumatica ERP, as described in [Example: Using OData Version 3](#). If you have created a Postman collection with the basic authentication configured, add the request described below to the collection and configure the request to inherit the authorization type from the parent collection.

Instructions for Retrieving the Names of the Fields of the Generic Inquiry

To obtain the list of fields of the generic inquiry, do the following:

1. Configure the following settings of the request:
 - HTTP method: GET
 - URL: `http://localhost/MyStoreInstance/OData/$metadata`
2. Send the request. If the request is successful, its response contains the 200 OK status code. In the response body, find the `EntityType` tag with `Name="Modified Stock Items"`, learn the `EntityType` value, and find the properties with the names `ItemStatus` and `LastModifiedOn`, which are shown in the following screenshots.

```
<EntitySet Name="Item Availability Data" EntityType="PX.Data.ItemAvailabilityData" />
<EntitySet Name="Modified Stock Items" EntityType="PX.Data.ModifiedStockItems" />
<EntitySet Name="Stock Item Attachments" EntityType="PX.Data.StockItemAttachments" />
<EntitySet Name="Customer Contacts" EntityType="PX.Data.CustomerContacts" />
```

Figure: System name of the generic inquiry

```

<EntityType Name="ModifiedStockItems">
  <Key>
    <PropertyRef Name="InventoryID" />
    <PropertyRef Name="Warehouse" />
    <PropertyRef Name="InventoryID_2" />
    <PropertyRef Name="Subitem" />
  </Key>
  <Property Name="InventoryID" Type="Edm.String" />
  <Property Name="Warehouse" Type="Edm.String" />
  <Property Name="Description" Type="Edm.String" />
  <Property Name="ItemStatus" Type="Edm.String" />
  <Property Name="LastModifiedOn" Type="Edm.DateTime" />

```

Figure: Fields of the generic inquiry

You will use these field names to configure a filter for the results of the generic inquiry through the OData protocol.

3. Save the request.

Instructions for Retrieving the List of Modified Stock Items

To retrieve the list of modified stock items, do the following:

1. Configure the following settings of the request:
 - HTTP method: GET
 - URL: `http://localhost/MyStoreInstance/OData/Modified%20Stock%20Items`
 - Parameters:

Parameter	Value
\$filter	<pre>ItemStatus eq 'Active' and LastModifiedOn gt datetime'2019-11-14T00:00:00.000'</pre> <p>Specify today's date instead of 2019-11-14.</p>
\$format	json

2. Send the request. The response of the successful request contains the 200 OK status code. The following code shows an example of the response body.

```

{
  "odata.metadata":
    "http://localhost/MyStoreInstance/odata/$metadata#Modified%20Stock%20Items",
  "value": [
    {
      "InventoryID": "AALEGO500 ",
      "Warehouse": "MAIN      ",
      "Description": "Lego, 500 piece set",
      "ItemStatus": "Active",
      "LastModifiedOn": "2019-11-14T06:13:09.57",
      "ItemClass": "STOCKITEM "
    }
  ]
}

```

```

    "BaseUnit": "PIECE",
    "QtyOnHand": "1999.000000",
    "InventoryID_2": "AALEGO500 ",
    "Subitem": "0"
  }
]
}

```

3. Save the request.

Related Links

- [Generic Inquiries and OData: General Information](#)

Example: Filtering the Requested Data (OData Version 4)

In this example, through the OData interface based on the OData Version 4.0 protocol, you will retrieve the list of active stock item records that were modified within the past day.

An active stock item is a `PX.Objects.IN.InventoryItem` record whose `StkItem` field contains `true` and `ItemStatus` field contains `AC`. Unlike the use of OData Version 3.0, you will not configure a generic inquiry to retrieve data from Acumatica ERP. Instead, you will use the `$filter` parameter to select those data records that meet the stated conditions.

The [Stock Items](#) (IN202500) form displays aggregated values among others. The OData Version 4.0 standard extension provides aggregation functions (see [OData Extension for Data Aggregation Version 4.0](#)), but Acumatica ERP 2022 R2 does not support them. You will display every single value instead of the aggregated values.

You use [OData URL conventions](#) to specify the conditions in the `$filter` parameter. In this example, you will specify the following conditions:

- The `StkItem` value is `true`
- The `ItemStatus` value is `AC`
- The `LastModifiedOn` value is greater than today's midnight

By using the request provided in this topic, you will retrieve the following information about a stock item:

- The stock item identifier (`InventoryCD`), description (`Descr`), status (`ItemStatus`), date and time of the last modification (`LastModifiedDateTime`), base unit (`BaseUnit`)
- The default warehouse (the `INSiteByDfltSiteID` navigation property)
- The item class (the `INItemClassByItemClassID` navigation property)
- The quantity on hand in each warehouse (the `INSiteStatusCollection` navigation property, the `QtyOnHand` field)

Prerequisites


Before you send the request in the example, you need to sign in to Acumatica ERP, as described in [Example: Using OData Version 3](#). If you have created a Postman collection with the basic authentication configured, add the request described below to the collection and configure the request to inherit the authorization type from the parent collection.

Instructions for Retrieving the List of Modified Stock Items

To retrieve the list of modified stock items, do the following:

1. Configure the following settings of the request:

- HTTP method: GET
- URL: *http://localhost/MyStoreInstance/ODataV4/PX_Objects_IN_InventoryItem*
- Parameters:

Parameter	Value
\$select	InventoryCD, Descr, ItemStatus, LastModifiedDateTime, BaseUnit
\$expand	<pre> INSiteByDfltSiteID(\$select=SiteCD), INItemClassByItemClassID(\$select=ItemClassCD), INSiteStatusCollection(\$select=QtyOnHand) </pre>
\$filter	<pre> StkItem eq true and ItemStatus eq 'AC' and LastModifiedDateTime gt 2020-12-01T00:00:00-04:00 </pre> <p>Specify today's date instead of 2020-12-01 and your timezone instead of -04:00.</p> <div>  <p>To specify a positive timezone, use %2b instead of + (for example, %2b03:00).</p> </div>

2. Send the request. The response of the successful request contains the 200 OK status code. The following code shows an example of the response body.

```

{
  "@odata.context":
    "http://localhost/MyStoreInstance/odatav4/$metadata#PX_Objects_IN_InventoryItem
    (InventoryCD,Descr,ItemStatus,LastModifiedDateTime,BaseUnit,
    INSiteByDfltSiteID(SiteCD),INItemClassByItemClassID(ItemClassCD),
    INSiteStatusCollection(QtyOnHand))",
  "value": [
    {
      "InventoryCD": "AALEGO500 ",
      "Descr": "Lego 500 piece set",
      "ItemStatus": "AC",
      "LastModifiedDateTime": "2020-12-01T12:34:56+03:00",
      "BaseUnit": "PIECE",
      "INSiteByDfltSiteID": {
        "SiteCD": "MAIN"
      },
      "INItemClassByItemClassID": {
        "ItemClassCD": "STOCKITEM "
      },
      "INSiteStatusCollection": [
        {
          "QtyOnHand": 1999.000000
        }
      ]
    }
  ]
}

```


3. Save the request.

Related Links

- [OData Extension for Data Aggregation Version 4.0](#)

Lesson Summary

In this lesson, you have learned how to export from Acumatica ERP the list of stock item records that were modified within a particular period by using different integration approaches. For information on how to retrieve the same data by using the contract-based REST API, see *Lesson 3.1: Retrieving the List of Modified Stock Items* in the *I310 Data Retrieval with REST API Basics* course.

Lesson 3.2: Monitoring Item Availability with Push Notifications

The BI application of the MyStore company needs to display up-to-date information about the item availability in warehouses. You will configure Acumatica ERP to track the changes in item availability and to send notifications to an HTTP address when the availability of any item changes. (The processing of these notifications in the BI application is outside of the scope of this course.) With push notifications, the BI application does not need to continually poll for the data to find out whether there are any changes to this data, which helps improve the performance of the application.

You may skip this lesson, if you have already completed *Lesson 3.2: Monitoring Item Availability with Push Notifications* in the *I310 Data Retrieval with REST API Basics* course.

Lesson Objective

In this lesson, you will learn how to configure Acumatica ERP to send push notifications to an HTTP address.

Example: Configuring Push Notifications

In this example, you will configure Acumatica ERP to monitor changes in the Item Availability Data (INGI0002) custom generic inquiry. Acumatica ERP will send push notifications to an HTTP address in JSON format about the changes.

Prerequisites

To test the push notifications that you will set up Acumatica ERP to send to an HTTP address, you will use the <http://webhookinbox.com/> website, which is an open-source service. To obtain an HTTP address that you will use for testing, do the following:

1. Open <http://webhookinbox.com/>.
2. Click **create an inbox**.
3. Copy the HTTP address that the site has generated. An example of the address is shown in the following screenshot. Do not close the webpage; you will use it for testing.

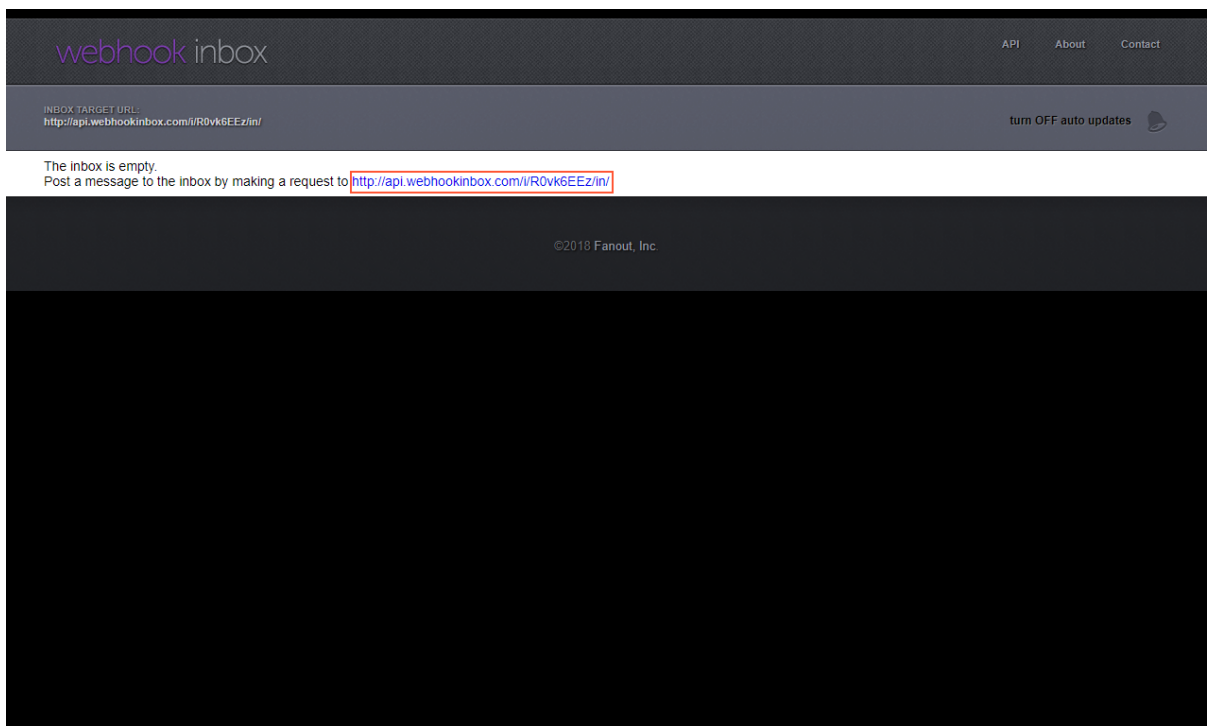


Figure: An HTTP address

Instructions for Configuring Push Notifications

To configure Acumatica ERP to send push notifications to an HTTP address, do the following:

1. In the Summary area of the *Push Notifications* (SM302000) form, specify the following settings:
 - **Destination Name:** *MyBIIntegration*
You can specify any name for the target notification destination.
 - **Destination Type:** *Webhook*
With *Webhook* selected, Acumatica ERP will send HTTP `POST` requests with notification information to an HTTP address.
 - **Address:** The address that you have created in the prerequisites of this example
2. On the **Generic Inquiries** tab, click **Add Row** on the toolbar of the **Inquiries** table.
3. For the added row, do the following:
 - a. In the **Inquiry Title** column of the added row, select *Item Availability Data*.
 - b. Select the **Track All Fields** check box.



If all fields in the results of the data query are tracked, the system produces push notifications for changes of any field in the results of the data query, which can cause overflow of the push notification queue. If you need to track only particular fields in the results of the data query, push notifications for changes in other fields are useless for you but consume system resources. Therefore, in this case, we recommend that you specify particular fields to be tracked in the **Fields** table.

4. On the form toolbar, click **Save**. The following screenshot shows the final configuration.

Push Notifications CUSTOMIZATION TOOLS ▾

* Destination Name:
 Header Name: ☒ Active

* Destination Type:
 Header Value:

* Address:

☒ GENERIC INQUIRIES BUILT-IN DEFINITIONS

Inquiries			Fields	
<input type="button" value="Refresh"/> <input type="button" value="+"/> <input type="button" value="X"/> <input type="button" value="VIEW INQUIRY"/> <input type="button" value="Filter"/> <input type="button" value="Reset"/>			<input type="button" value="Refresh"/> <input type="button" value="+"/> <input type="button" value="X"/> <input type="button" value="Filter"/> <input type="button" value="Reset"/>	
Active	* Inquiry Title	Track All Fields	Table Name	Field Name
> <input checked="" type="checkbox"/>	Item Availability Data	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>		

Figure: Push notification configuration

Instructions for Testing Push Notifications

To test the configured notifications, do the following:

1. On the [Sales Orders](#) (SO301000) form, create a sales order as follows:
 - a. In the **Customer** box, select *C000000001*.
 - b. Click **Remove Hold**.
 - c. On the table toolbar of the **Details** tab, click **Add Items**. In the dialog box that opens, select the unlabeled check box for the *AALEGO500* inventory item, and click **Add & Close**.
 - d. On the form toolbar, click **Save**.
2. On the webhook inbox webpage, which you have generated in the prerequisites, review the push notification that Acumatica ERP sent for the change in the available quantity for the *AALEGO500* inventory item. When you created a sales order, the available quantity of the *AALEGO500* inventory item was changed. The notification contains the information about the new value of the `QtyAvailable` field of the changed record in the `Inserted` element and the information about the previous value of this field in the `Deleted` element. An example of the notification is shown in the following code.

```
{
  "Inserted":
  [
    {
      "Warehouse": "MAIN ",
      "InventoryID": "AALEGO500",
      "Description": "Lego, 500 piece set",
      "QtyOnHand": 1999.000000,
      "QtyAvailable": 1974.000000
    }
  ],
  "Deleted":
  [
    {
      "Warehouse": "MAIN",
      "InventoryID": "AALEGO500 ",
      "Description": "Lego, 500 piece set",

```

```

        "QtyOnHand":1999.000000,
        "QtyAvailable":1975.000000
    }
],
"Query":"Item Availability Data",
"CompanyId":"MyStore",
"Id":"06284bdf-2952-4a9a-bf60-2ad0cd33924e",
"TimeStamp":636771954942887487,
"AdditionalInfo":
{
    "PXPerformanceInfoStartTime":"11/14/2019 13:51:34"
}
}

```

Related Links

- [Push Notifications](#)
- [To Configure Push Notifications](#)
- [Push Notification Format](#)

Additional Information: Push Notifications

The scenarios described in this topic are outside of the scope of this course but may be useful to some readers.

Sending Push Notifications to a Message Queue or the SignalR Hub

Acumatica ERP can send notifications to the following predefined destination types: an HTTP address (webhook), a Microsoft message queue, a SignalR hub (the destination type implemented in Acumatica ERP by using the ASP.NET SignalR library), and a commerce push destination.

For details about these destination types, see [Push Notification Destinations](#). For information about how to configure Acumatica ERP to send notifications to a message queue or the SignalR hub, see [To Configure Push Notifications](#).

Processing Failed Notifications

If Acumatica ERP is configured to send push notifications to an HTTP address or to a message queue and cannot send these notifications for some reason, Acumatica ERP saves the information about these notifications and displays these notifications on the [Process Push Notifications](#) (SM502000) form. You can resend the failed notifications for two days, after which the notifications are removed from the Acumatica ERP database. For details about how to resend the failed notifications, see [To Process Failed Notifications](#).

If Acumatica ERP is configured to send push notifications to the SignalR hub, the failed notifications cannot be resent. For details, see [Push Notification Destinations](#).

Defining the Data Query for Push Notifications in Code

You can define the data query that specifies the data changes for which Acumatica ERP should send notifications in one of the following ways:

- By using a generic inquiry, as has been described in this lesson
- By using a built-in definition, which is a data query defined in code

For the information about how to create a built-in definitions, see [To Create a Built-In Definition](#).

Adding Custom Information to Push Notifications

Push notifications in JSON format that an external application receives include the `AdditionalInfo` element. In this element, you can include custom information, such as the name of the user that performed the data change. For details about how to include custom information in push notifications, see [To Add Additional Information to Push Notifications](#).

Creating a Custom Destination for Push Notifications

Acumatica ERP can send notifications to the following predefined destination types: an HTTP address, a Microsoft message queue, a SignalR hub, and a commerce push destination. If these destination types do not satisfy the requirements of your application, you can create a custom destination type in code. For details about how to develop a custom destination type, see [To Create a Custom Destination Type](#).

Lesson Summary

In this lesson, you have configured Acumatica ERP to send push notifications to an HTTP address. You have also reviewed the possible options for the configuration of push notifications.

Part 4: Creation of a Customization Package

To distribute the changes that you made to your Acumatica ERP instance to the Acumatica ERP instances of the company, you need to include all the custom items that you have created in Acumatica ERP in a customization package.

By completing the examples of this part, you will configure customization packages that include all the items that have been created in Acumatica ERP.

Lesson 4.1: Configuring a Customization Project and Exporting It

In this lesson, for the OData Version 3.0 and Version 4.0 interface, you will create a customization project, include all necessary data in this project, and export the project as a ZIP file. You will need to include the following items in the customization project:

- All generic inquiries that are used in the integration application
- The push notification definition

Lesson Objectives

In this lesson, you will learn how to do the following:

- Include in a customization project all necessary items that were created in Acumatica ERP for an integration application
- Export the customization project as a ZIP file

Example: Creating a Customization Package for the OData Version 3 Integration

In this example, you will create a customization package for the OData Version 3.0 integration that you have implemented during this course.

You will include in the customization package the following items:

- Generic inquiries:
 - Customer Contacts (ARGI0015), which was used in [Example: Using a Custom Generic Inquiry \(OData Version 3\)](#)
 - Item Availability Data (INGI0002), which was used in [Example: Using a Generic Inquiry \(OData Version 3\)](#) and [Example: Configuring Push Notifications](#)
 - Modified Stock Items (INGI0016), which was used in [Example: Filtering the Result of a Generic Inquiry \(OData Version 3\)](#)
- Push notification definition:
 - MyBIIntegration, which was created in [Example: Configuring Push Notifications](#)

You can include any other items in the customization project if they are necessary for the integration application.

Creating a Customization Package

To create a customization project and export it to a ZIP file, do the following:

1. On the [Customization Projects](#) (SM204505) form, do the following:

- a. On the form toolbar, click **Add Row**.
 - b. In the **Project Name** column of the added row, type the name of the project:
`MyBIIntegrationOData`.
 - c. On the form toolbar, click **Save**
2. In the **Project Name** column, click the `MyBIIntegrationOData` link, which opens the Customization Project Editor for the `MyBIIntegrationOData` customization project.
 3. Include the generic inquiries in the customization project as follows:
 - a. In the navigation pane of the Customization Project Editor, click **Generic Inquiries** to open the Generic Inquiries page.
 - b. On the toolbar of the Generic Inquiries page, click **Add New Record**.
 - c. In the **Add Generic Inquiries** dialog box, which opens, select the check boxes in the rows with the following inquiry titles (as shown in the screenshot below):
 - *Customer Contacts*
 - *Item Availability Data*
 - *Modified Stock Items*

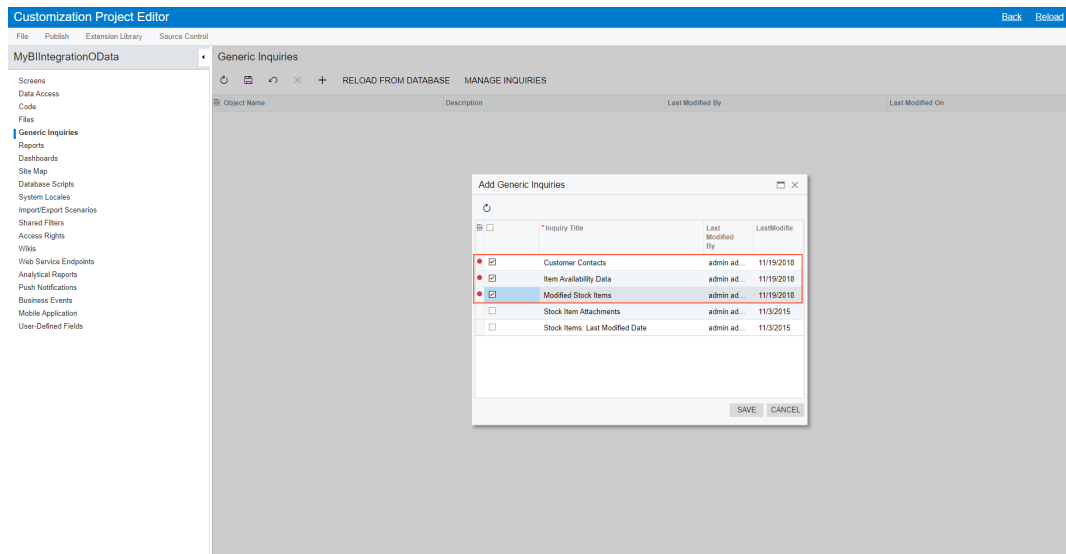


Figure: Selecting the generic inquiries

- d. Click **Save**.
Three inquiries have been added to the Generic Inquiries page.
4. Include the definition of push notifications in the customization project as follows:
 - a. In the navigation pane of the Customization Project Editor, click **Push Notifications**.
 - b. On the toolbar of the Push Notifications page, click **Add New Record**.
 - c. In the **Add Push Notifications** dialog box, select the check box in the row for the `MyBIIntegration` destination name.
 - d. Click **Save**.
The `MyBIIntegration` definition has been added to the Push Notifications page.
5. Close the Customization Project Editor.
6. On the [Customization Projects](#) form, click the row for the `MyBIIntegrationOData` customization project, and on the form toolbar, click **Export**; save the ZIP file with the customization project to your computer.

You can import this ZIP file to another Acumatica ERP instance and publish the customization project to make the items from the customization project available in this instance.

Related Links

- [Managing Items in a Project](#)

Example: Creating a Customization Package for the OData Version 4 Integration

In this example, you will create a customization package for the OData Version 4.0 integration that you have implemented during this course.

You will include in the customization package the following items:

- Generic inquiry: Item Availability Data (INGI0002), which was used in [Example: Configuring Push Notifications](#)
- Push notification definition: MyBIIntegration, which was created in [Example: Configuring Push Notifications](#)

You can include any other items in the customization project if they are necessary for the integration application.

The procedure of creating a customization package is similar to that described in [Example: Creating a Customization Package for the OData Version 3 Integration](#).

Related Links

- [Managing Items in a Project](#)

Lesson Summary

In this lesson, you have configured a customization project to include all necessary items that have been created in Acumatica ERP for the integration application. You have exported this customization project as a ZIP file.

Appendix: Comparison of the Integration Interfaces

This topic summarizes the differences between the integration approaches described in this training guide and in *I310 Data Retrieval with REST API Basics*.

Initial Configuration

The following table summarizes the configuration of Acumatica ERP that should be performed for each of the integration interfaces.

Integration Interface	Configuration of Acumatica ERP
OData Version 3.0 interface	Optional: Configuration of CORS
OData Version 4.0 interface	Optional: Configuration of CORS
REST API	Optional: Configuration of a custom endpoint or an endpoint extension

Signing In to and Signing Out from Acumatica ERP

The following table summarizes the sign-in and sign-out options and limitations for each of the integration interfaces.

Integration Interface	Authentication and Authorization	Limit for the Number of API Users
OData Version 3.0 interface	<ul style="list-style-type: none"> Basic authentication OAuth 2.0 authorization 	No
OData Version 4.0 interface	<ul style="list-style-type: none"> Basic authentication OAuth 2.0 authorization 	No
REST API	<ul style="list-style-type: none"> POST requests to the sign-in and sign-out endpoints OAuth 2.0 authorization 	Yes

Data Retrieval from a Data Entry Form

The following table summarizes the options of data retrieval from a data entry form for each of the integration interfaces.

Integration Interface	Ability to Retrieve Data from a Data Entry Form	Ability to Retrieve Custom Fields	Ability to Retrieve User-Defined Fields
OData Version 3.0 interface	Yes, if a custom generic inquiry is created that retrieves the needed data.	Yes	Yes
OData Version 4.0 interface	Yes, through the data access classes that correspond to the data entry form.	Yes	Yes

Integration Interface	Ability to Retrieve Data from a Data Entry Form	Ability to Retrieve Custom Fields	Ability to Retrieve User-Defined Fields
REST API	Yes, if the form is mapped in an endpoint.	Yes	Yes

Data Retrieval from a Generic Inquiry

The following table summarizes the options of data retrieval from a generic inquiry form for each of the integration interfaces.

Integration Interface	Ability to Retrieve Data from a Generic Inquiry Without Parameters	Ability to Specify the Values of Parameters of Generic Inquiries
OData Version 3.0 interface	Yes	No
OData Version 4.0 interface	Yes, by retrieving data from DACs on which the generic inquiry is based	No
REST API	Yes, if the generic inquiry is mapped in an endpoint	Yes, if the generic inquiry and its parameters are mapped in an endpoint

Retrieval of the Delta of Records

The following table summarizes the options for the retrieval of the delta of records for each of the integration interfaces.

Integration Interface	Ability to Filter the Data by Conditions
OData Version 3.0 interface	Yes
OData Version 4.0 interface	Yes
REST API	Yes

Creation of the Customization Package

The following table summarizes which items should be included in the customization projects for each of the integration interfaces.

Integration Interface	Generic Inquiries	Web Service End-points	Push Notifications	Other Customization Project Items
OData Version 3.0 interface	Yes	No	Yes, if push notifications have been used for the integration	Yes, if these items have been used for the integration

Integration Interface	Generic Inquiries	Web Service End-points	Push Notifications	Other Customization Project Items
OData Version 4.0 interface	Yes, if generic inquiries have been used for the integration	No	Yes, if push notifications have been used for the integration	No
REST API	Yes, if generic inquiries have been used in a custom endpoint or an endpoint extension	Yes, if a custom endpoint or an endpoint extension is used	Yes, if push notifications have been used for the integration	Yes, if these items have been used for the integration

Appendix: Web Integration Scenario Reference

In this topic, you can find reference links to the topics that describe how to implement the following web integration scenarios:

- **Retrieving the list of customers:** [Lesson 2.1: Retrieving the List of Customers with Contacts.](#)
- **Retrieving the list of modified stock items:** [Lesson 3.1: Retrieving the List of Modified Stock Items.](#)
- **Retrieving the quantities of stock items:** [Lesson 2.2: Retrieving the Quantities of Stock Items.](#)
- **Signing in:** [Lesson 1.1: Signing In to Acumatica ERP.](#)