

Developer Course

Web Services

I310 Data Retrieval with REST API Basics 2022 R1

Revision: 3/23/2022

Contents

Copyright	4
Introduction	5
How to Use This Course	6
Course Prerequisites	7
Deploying an Acumatica ERP Instance for the Training Course.....	7
Overview of the Acumatica ERP Integration Services	8
Contract-Based REST API.....	9
Company Story	10
Part 1: Client Application Configuration	11
Lesson 1.1: Signing In to and Signing Out from Acumatica ERP.....	11
Example: Signing In to Acumatica ERP.....	11
Example: Signing Out from Acumatica ERP.....	13
Additional Information: OAuth 2.0 Authorization.....	15
Additional Information: Custom Endpoints and Endpoint Extensions.....	15
Additional Information: Use of a REST Schema in a Visual Studio Project.....	15
Lesson Summary.....	15
Part 2: Initial Data Retrieval	16
Lesson 2.1: Retrieving the List of Customers with Contacts.....	16
Example: Using GET and the \$expand and \$select Parameters.....	17
Additional Information: Retrieval of Custom and User-Defined Fields.....	20
Lesson Summary.....	21
Lesson 2.2: Retrieving the Quantities of Stock Items.....	21
Example: Using PUT to Retrieve Data from an Inquiry.....	22
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: Using GET and the LastModified Field.....	27
Lesson Summary.....	29
Lesson 3.2: Monitoring Item Availability with Push Notifications.....	30
Example: Configuring Push Notifications.....	30
Additional Information: Push Notifications.....	33
Lesson Summary.....	34
Part 4: Creation of a Customization Package	35

Lesson 4.1: Configuring a Customization Project and Exporting It.....35

 Example: Creating a Customization Package for the Contract-Based REST API Integration.....35

 Lesson Summary..... 37

Appendix: Comparison of the Integration Interfaces.....38

Appendix: Web Integration Scenario Reference..... 41

Appendix: Troubleshooting.....42

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 R1

Last Updated: 03/23/2022

Introduction

External systems can use Acumatica ERP integration interfaces to access the business functionality and data of Acumatica ERP. This course introduces the contract-based REST API.

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 contract-based REST API. The course gives you ideas about how to develop your own applications by using the REST API. It demonstrates the main elements of the REST API and its use in typical tasks that integrate Acumatica ERP with third-party applications. This course contains the same integration scenarios as *1300 Data Retrieval with OData*, which introduces the OData interface.

After you complete all the lessons of the course, you will be familiar with the basic techniques of data retrieval through the Acumatica ERP contract-based REST API.

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 *I310 Certification Test: Data Retrieval Basics*.

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 contract-based REST API.

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 consists of a brief description of the web integration scenario and an example 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.

What Are the Documentation Resources?

The complete Acumatica ERP and Acumatica Framework documentation is available on <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.

Licensing Information

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 for the training course conforms to the [System Requirements for Acumatica ERP 2022 R1](#).
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 R1 with the name *MyStoreInstance* and a tenant that contains the *I100* data. If you have completed the *I300 Data Retrieval with OData* 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 *I100* data inserted by specifying the following settings:
 - **Login Tenant Name:** *MyStore*
 - **New:** Selected
 - **Insert Data:** *I100*
 - **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.

Overview of the Acumatica ERP Integration Services

Acumatica ERP provides integration with external data sources and third-party systems through integration services. These integration services include the import and export of data by means of integration scenarios and interaction with external systems through the OData protocol, SOAP APIs, REST API, and mobile API. External applications and systems that use the Acumatica ERP integration services can access the data managed by Acumatica ERP and the business functionality of Acumatica ERP. For example, you can integrate Acumatica ERP with a business intelligence (BI) system, an eCommerce system, or an online store system.

In the diagram below, the integration service that is covered in this training course is highlighted in blue.

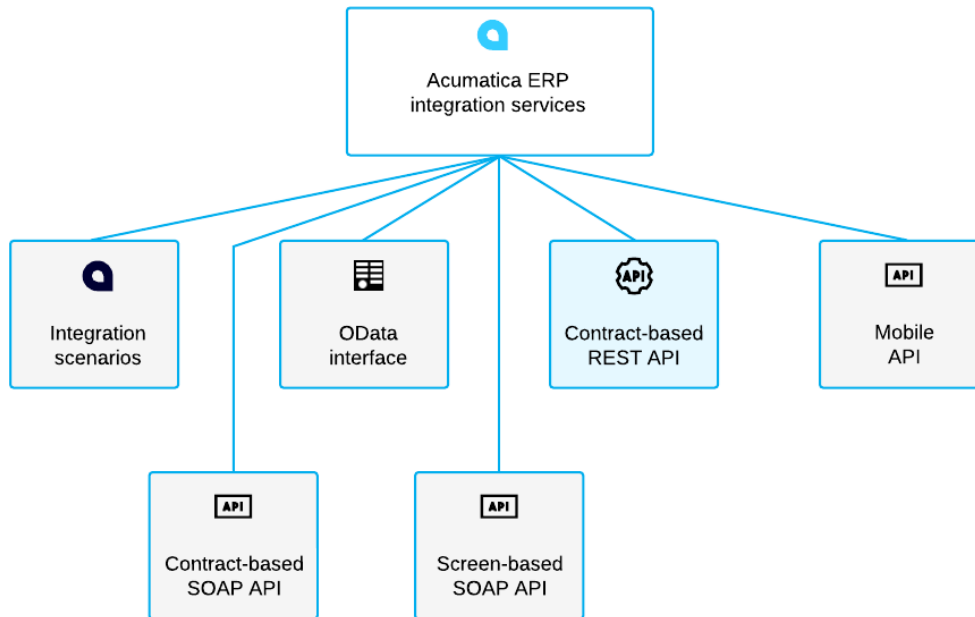


Figure: Acumatica ERP integration services



The integration scenarios, SOAP APIs, OData interface, and mobile API are outside of the scope of this training course. For details about the integration scenarios, see the *I100 Integration: Integration Scenarios* training course. For details about the OData interface, see the *I300 Data Retrieval with OData* training course. For information about the mobile API, see the *T400 Development: Basic | Customization of Mobile Application* and *T410 Development: Advanced | Customization of Mobile Application* training courses.

This training course covers the contract-based REST API, which operates through the REST service with business logic objects and their properties and methods, which do not depend on Acumatica ERP forms. (In this context, *contract-based* means based on the object model the web service API provides.)

Through the web services of Acumatica ERP, external systems can retrieve data records from Acumatica ERP. With the web services, external systems can also process these records and save the new or updated records to Acumatica ERP.

Contract-Based REST API

The contract-based web service APIs operate with business logic objects that do not depend on Acumatica ERP forms or their properties and methods. (In this context, *contract-based* means based on the object model the web service API provides.) Each web service contract is fixed and does not change based on system customization, localization, or any other changes made to Acumatica ERP.

You can work with the contract-based web services through the REST interface.

You access the contract-based REST API through endpoints, which you can configure on the [Web Service Endpoints](#) (SM207060) form. An *endpoint* is an entry point to the Acumatica ERP web services. For each endpoint that a web service API provides, a *contract* of the endpoint defines the entities, along with their actions and fields, that are available through the endpoint and the methods that you can use to work with these entities. For the information about the API entities, fields, and actions, see [API Entities, Fields, and Actions](#) in the documentation.

You can use two types of endpoints to access the web services: system and custom. The system endpoints are preconfigured in the system and have the *Default* name. Each of these endpoints has a predefined contract, which includes the API that is preconfigured in the system. You cannot change the contract of a system endpoint. By default, there are no custom endpoints in the system. If the API provided by the system endpoint is not sufficient for the requirements of your application, you can create a custom endpoint.

Related Links

- [Contract-Based Web Service API](#)
- [Endpoints and Contracts](#)
- [Comparison of Contract Versions](#)
- [Comparison of System Endpoints](#)
- [API Entities, Fields, and Actions](#)

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 contract-based REST API. The following diagram shows the integration scheme of the MyStore BI application with Acumatica ERP.

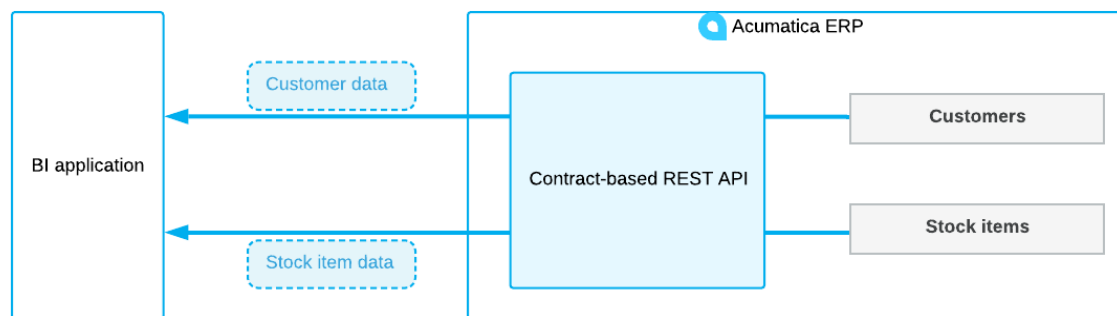


Figure: Integration of the MyStore BI application and Acumatica ERP

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 and sign out from 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 and Signing Out from Acumatica ERP

Before you can retrieve data from Acumatica ERP by using the contract-based REST API, you need to sign in to Acumatica ERP. In this lesson, you will add the requests for signing in to and signing out from Acumatica ERP.



If two-factor authentication is turned on for a user account that is used to sign in to Acumatica ERP through a web service API, the two-factor authentication is bypassed by the sign-in method. You can explicitly turn off two-factor authentication for the user accounts that are used only through the web service API by defining the user type of the user account. For details about the creation of a user type to be used by an external application, see [To Limit the Number of API Connections of Integrated Applications](#).

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 do the following:

- Sign in to Acumatica ERP through the REST API
- Sign out from Acumatica ERP through the REST API

Example: Signing In to Acumatica ERP

In this example, through the REST API, you will sign in to Acumatica ERP.

To sign in to Acumatica ERP, you will use the `POST` HTTP method and the URL for signing in. You will pass the authentication parameters (the username, password, tenant name, and branch ID) in the body of the request.

With each subsequent request to the service, the application has to pass the cookies that it has received during sign-in. The Postman application, which you use for testing the REST API requests in this training course, appends the cookies to subsequent requests automatically.

For each sign-in to Acumatica ERP, you must sign out after you finish your work with Acumatica ERP to close the session. If the session is not closed, you may have issues with subsequent sign-ins to Acumatica ERP through the web service API. The trial license, which you are using in the instance for this training course, limits the number of API users to two, as you can see in the **Maximum Number of Web Services API Users** box on the **License** tab of the License Monitoring Console (SM604000) form. Thus, if the session is not closed twice, you will not be able to sign in to Acumatica ERP through the web service API for ten minutes. For details on how you can deal with this issue, see [Appendix: Troubleshooting](#).

Example

To sign in to Acumatica ERP, do the following:



Instead of configuring the requests by yourself, you can import the Postman collection from the `REST.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/2022R1/IntegrationDevelopment/I310>.)

1. In Postman, configure the following settings of the contract-based REST API request:

- HTTP method: `POST`
- URL: `http://localhost/MyStoreInstance/entity/auth/login`
- The headers shown below

Key	Value
Accept	application/json
Content-Type	application/json

- The following body of the request

```
{
  "name": "admin",
  "password": "123",
  "tenant": "MyStore",
  "branch": "MYSTORE"
}
```

You specify the values in the body of a sign-in request as follows:

- In the `name` element, you specify the username that the application should use to sign in to Acumatica ERP.
 - In the `password` element, you specify the password for the username.
 - In the `tenant` element, you specify the name of the tenant to which the application should sign in. In this example, this is the name (*MyStore*) that you have specified during the creation of the Acumatica ERP instance for the training course. You can view the name that should be used for the tenant in the **Login Name** box of the [Tenants](#) (SM203520) form.
 - In the `branch` element, you specify the ID of the branch (in this example, *MYSTORE*) to which the application should sign in. You can view the ID of the branch in the **Branch ID** box of the [Branches](#) (CS102000) form.
2. Send the request. If the request is successful, the response contains the `204 No Content` status code and the cookies that should be sent with subsequent requests to Acumatica ERP. In Postman, you can view the cookies on the **Cookies** tab, which is shown in the following screenshot. The list of cookies contains six elements, one of which is `.ASPXAUTH`.

Overview | POST Lesson 1.1: Signin... X | POST Lesson 1.1: Signin... | + ...

MyBlIntegr... / Part 1: Client Applicat... / Lesson 1.1: Signing in ...

POST http://localhost/MyStoreInstance/entity/auth/login Send

Params Auth Headers (12) Body ● Pre-req. Tests Settings Cookies

raw JSON Beautify

```

1 {
2   "name": "admin",
3   "password": "123",
4   "tenant": "MyStore",
5   "branch": "MYSTORE"
6 }

```

Cookies 204 No Content 1 m 24.12 s 1.21 KB Save Response

ASP.NET_Ses...	jypocrukcvmj...	localhost	/	Session	true	false
UserBranch	82	localhost	/	Session	false	false
Locale	Culture%3De...	localhost	/	Mon, 20 Sep ...	false	false
.ASPXAUTH	A37D42B8E49 C5B2F4E89A4	localhost	/	Session	true	false
requestid	AD91BD067B8 47042205707	localhost	/	Session	false	false
requeststat	8821A110D41 DC4097226EA BD3D51B3CB	localhost	/	Session	false	false

Figure: Cookies after signing in

3. Save the request.

Related Links

- [Sign In to the Service](#)
- [License Restrictions for API Users](#)
- [Companies and Branches](#)

Example: Signing Out from Acumatica ERP

In this example, through the REST API, you will sign out from Acumatica ERP.

To sign out from Acumatica ERP, do the following:

1. In Postman, configure the contract-based REST API request as follows:

- HTTP method: POST
- URL: `http://localhost/MyStoreInstance/entity/auth/logout`
- The headers shown below

Key	Value
Accept	application/json
Content-Type	application/json

2. Send the request. The response of the successful request contains the 204 No Content status code and the list of cookies, which now contains four elements (the .ASPXAUTH element is missing), as shown in the following screenshot.

Overview POST Lesson 1.1: Signin... X POST Lesson 1.1: Signin... + ...

MyBlInteg... / Part 1: Client Applica... / Lesson 1.1: Signing out ... Save

POST ... `http://localhost/MyStoreInstance/entity/auth/logout` Send

Params Auth Headers (11) Body Pre-req. Tests Settings Cookies

Headers 9 hidden

	KEY	VALUE	DESC	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Accept	application/json				
<input checked="" type="checkbox"/>	Content-Type	application/json				
	Key	Value	Description			

Cookies ... 204 No Content 149 ms 757 B Save Response

Name	Value	Domain	Path	Expires	HttpOnly	Secure
Locale	TimeZone%3...	localhost	/	Session	false	false
UserBranch	82	localhost	/	Session	false	false
requestid	C34018A628 ...	localhost	/	Session	false	false
requeststat	%2Bst%3A12!...	localhost	/	Session	false	false

Figure: Cookies after signing out

3. Save the request.

Related Links

- [Sign Out from the Service](#)

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 *1320 Advanced Data Retrieval with REST API* training course.

Additional Information: Custom Endpoints and Endpoint Extensions

The API provided by the system endpoints of Acumatica ERP may not be sufficient for the requirements of your application. For example, you may need to retrieve data from a custom generic inquiry, or you may have added new data access classes and business logic to Acumatica ERP in a customization project. If the entity from which you want to retrieve data is not available in any of the system endpoints of Acumatica ERP, to retrieve the data from this entity, you should create a custom endpoint either from scratch or by extending an existing endpoint.

For details about custom endpoints and endpoint extensions, see [Custom Endpoints and Endpoint Extensions](#) in the documentation.

Additional Information: Use of a REST Schema in a Visual Studio Project

You can create a Visual Studio project and use the REST schema contained in the Acumatica ERP `swagger.json` files for interaction with Acumatica ERP. You import the REST schema to a Visual Studio solution in two stages:

1. You generate Visual Studio packages from the REST schema.
2. You include the generated Visual Studio packages in your Visual Studio solution.

For details about how to import a REST schema to a Visual Studio solution, see [To Import a REST Schema to a Visual Studio Solution](#) in the documentation.

Lesson Summary

In this lesson, you have learned how to sign in to and sign out from Acumatica ERP through the REST API. For information on how to sign in to Acumatica ERP by using the OData interface, see *Lesson 1.1: Signing In to Acumatica ERP* in the *1300 Data Retrieval with OData* 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: 🔍 Balance: 730.00
 * Customer Status: Active ▾ Prepayment Balance: 0.00
 * Customer Class: 🔍 ✎

🚨 GENERAL FINANCIAL BILLING SHIPPING PAYMENT METHODS CONTACTS SALESPERSONS ATTRIBUTES

ACCOUNT INFO
 ACCOUNT ADDRESS
 Address Line 1:
 Address Line 2:
 City:
 State: 🔍
 Postal Code:
 * Country: 🔍

ADDITIONAL ACCOUNT INFO
 Business 1 ▾
 Business 2 ▾
 Fax ▾
 Account Email: ✉️
 Web: 🌐

PRIMARY CONTACT
 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

Lesson Objective

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

Example: Using GET and the \$expand and \$select Parameters

In this example, through the REST API, you will configure an HTTP request that exports the list of customer records with contacts and addresses.

You will use the GET HTTP method and the `Customer` entity of the `Default/20.200.001` endpoint to list customer records. The `Customer` entity is mapped to the `Customers` (AR303000) form.

Web Service Endpoints
Default 20.200.001

CUSTOMIZATION TOOLS ▾

← ↻ 📄 ↶ + DELETE EXTEND ENDPOINT ...

* Endpoint Name: * Endpoint Version:

+ INSERT 🗑️

Customer

- Actions
- Attributes: AttributeValue[]
- BillingContact: Contact
- Contacts: CustomerContact
- CreditVerificationRules: Cr...
- MainContact: Contact
- PaymentInstructions: Busin...
- PrimaryContact: Contact
- Salespersons: CustomerSe...
- ShippingContact: Contact
- CustomerClass
- CustomerLocation
- CustomerPaymentMethod
- CustomerPriceClass

ENTITY PROPERTIES **FIELDS**

🔄 + ✕ POPULATE EXTEND ENTITY VALIDATE ENTITY |↔|

Field Name	Mapped Object	Mapped Field	Field Type
> AccountRef	General -> Accoun...	AcctReferenceNbr	StringValue
ApplyOverdueCharges	General -> Additio...	FinChargeApply	BooleanValue
AutoApplyPayments	General -> Additio...	AutoApplyPayments	BooleanValue
BillingAddressOverride	Financial -> Retain...	OverrideBillAddress	BooleanValue
BillingContactOverride	Financial -> Finan...	OverrideBillContact	BooleanValue
CreatedDateTime	General -> Accoun...	CreatedDateTime	DateTimeValue
CurrencyID	General -> Additio...	CuryID	StringValue
CurrencyRateType	General -> Additio...	CuryRateTypeID	StringValue
CustomerClass	Customer Summary	Customer Class	StringValue
CustomerID	Customer Summary	Customer ID	StringValue

Figure: The Customer entity

Because the database can contain thousands of customer records and each record includes dozens of fields, to achieve the best performance of the application, you need to specify the fields of the customer records that should be returned. You will use the `$select` parameter to specify the fields whose values should be retrieved from Acumatica ERP for each customer record. For a customer record, you will export the values of the following fields of the `Customer` entity:

- CustomerID
- CustomerName
- CustomerClass
- MainContact (nested Contact entity):
 - Email
 - Phone1
- Address (nested Address entity):
 - AddressLine1
 - AddressLine2
 - City
 - State
 - PostalCode



By using the approach described in this example, you can export the values of the fields that are available in the endpoint. To export any other fields (such as the fields added to an Acumatica ERP form in a customization project), you should use the approach described in [Additional Information: Retrieval of Custom and User-Defined Fields](#).

You will use the `$expand` parameter to specify the nested entities to be returned. In this parameter, you have to specify all the nested entities whose fields you need to retrieve from Acumatica ERP. That is, in this example, you have to specify the following entities: `MainContact` and `MainContact/Address`.

Prerequisites

Before you can test the example, you need to sign in to Acumatica ERP, as described in [Example: Signing In to Acumatica ERP](#). For the request in this example, you have to pass the cookies that you have received during the sign-in.

Instructions for Retrieving the List of Customers

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

- In Postman, configure the following settings of the contract-based REST API request:
 - HTTP method: GET
 - URL: `http://localhost/MyStoreInstance/entity/Default/20.200.001/Customer`
 - The following parameters of the request

Parameter	Value
\$expand	MainContact,MainContact/Address
\$select	CustomerID, CustomerName, CustomerClass, MainContact/Email, MainContact/Phone1, MainContact/Address/AddressLine1, MainContact/Address/AddressLine2, MainContact/Address/City, MainContact/Address/State, MainContact/Address/PostalCode

- The headers shown below

Key	Value
Accept	application/json
Content-Type	application/json

- Send the request. If the request is successful, the response contains the 200 OK status code and includes the list of requested fields of all customer records in JSON format. The following code example shows a fragment of the response body.

```
[
  {
    "id": "f475b40c-8f59-4750-b361-84a9fd34000d",
    "rowNumber": 1,
    "note": {
      "value": ""
    },
    "CustomerClass": {
      "value": "DEFAULT"
    },
    "CustomerID": {
      "value": "C000000001"
    },
    "CustomerName": {
      "value": "Jersey Central Office Equip"
    },
    "MainContact": {
```

```

    "id": "d85c888a-d1c9-448a-a297-d3be2d6c9df9",
    "rowNumber": 1,
    "note": {
      "value": null
    },
    "Address": {
      "id": "6da0645e-63f1-4abc-85f3-a91f5308d817",
      "rowNumber": 1,
      "note": {
        "value": null
      },
      "AddressLine1": {
        "value": "1 De Villiers & Harrison St, 11-th Flr."
      },
      "AddressLine2": {},
      "City": {
        "value": "Johannesburg"
      },
      "PostalCode": {},
      "State": {},
      "custom": {},
      "files": []
    },
    "Email": {
      "value": "jersey-equip@mail.com"
    },
    "Phone1": {
      "value": "+1 (777) 283-0414"
    },
    "custom": {},
    "files": []
  },
  "custom": {},
  "files": []
},
...
]

```

3. Save the request.

Related Links

- [Retrieve Records by Conditions](#)
- [Parameters for Retrieving Records](#)

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 contract-based REST API. The retrieval of custom and user-defined fields is outside of the scope of this course but may be useful to some readers.

To retrieve custom and user-defined fields through the contract-based REST API, you can use the `$custom` parameter. You can use the same approach to retrieve the values of any fields that are not included in the entity

definition. For details about the work with the fields that are not included in the entity definition, see [Custom Fields](#) in the documentation. For details about the `$custom` parameter, see [\\$custom Parameter](#) in the documentation.

You can also add the custom fields to a custom endpoint or endpoint extension and work with them by using the same approach as was described in this lesson for the fields of the system endpoint. User-defined fields cannot be added to custom endpoints and endpoint extensions. For details about custom endpoints and endpoint extensions, see [Custom Endpoints and Endpoint Extensions](#) in the documentation.

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 the contract-based REST API. For information on how to retrieve the same data by using the OData interface, see [Lesson 2.1: Retrieving the List of Customers with Contacts](#) in the *1300 Data Retrieval with OData* 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. To view the on-hand and available quantities of stock items on this form, you select each item one by one on the form. You have to do the same if you retrieve the quantities of items through the contract-based REST API. However, this way of data export has a drawback: It performs many requests to the Acumatica ERP database; therefore, it has slow system performance.

In this lesson, you will use another way of exporting data from Acumatica ERP. 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

Lesson Objective

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

Example: Using PUT to Retrieve Data from an Inquiry

In this example, through the REST API, you will configure an HTTP request that exports the quantities of stock items from the Item Availability Data (INGI0002) custom generic inquiry in Acumatica ERP.

Because no entity is mapped to the custom generic inquiry in the system endpoints, which are provided in Acumatica ERP by default, to export data from this generic inquiry by using the contract-based API, you need to use a custom endpoint or an extension of an existing endpoint. You will use the *ItemAvailabilityData/0001* custom endpoint, which has been preconfigured for this course on the [Web Service Endpoints \(SM207060\)](#) form. (The creation of custom endpoints and endpoint extensions is outside of the scope of this course. For more details, see [Additional Information: Custom Endpoints and Endpoint Extensions.](#))

You will use the PUT HTTP method and the *ItemAvailabilityDataInquiry* entity of the custom *ItemAvailabilityData/0001* endpoint. The *ItemAvailabilityDataInquiry* entity is mapped to the Item Availability Data custom generic inquiry. The PUT method is always used to retrieve data from an entity that is mapped to a generic inquiry because generic inquiries can contain parameters whose values you may need to specify by passing the values of the parameters in the body of the request.

You will use the `$expand` parameter to specify the nested detail entity (*Results*) to be returned. To retrieve the data from a generic inquiry, you have to expand the nested detail entity with the results of the generic inquiry. This entity is usually called *Results*.

Although this generic inquiry has a parameter (which is optional), you will not specify the parameter value of the inquiry in this example, because you need to retrieve the full list of records of the generic inquiry. Therefore, you will pass no values in the body of the request to specify the parameter.

Prerequisites

Before you can test the example, you need to sign in to Acumatica ERP, as described in [Example: Signing In to Acumatica ERP](#). For the request in this example, you have to pass the cookies that you have received during the sign-in.

Instructions for Retrieving the Quantities of Items

To retrieve the quantities of items, do the following:

- In Postman, configure the following settings of the contract-based REST API request:
 - HTTP method: PUT
 - URL: `http://localhost/MyStoreInstance/entity/ItemAvailabilityData/0001/ItemAvailabilityDataInquiry`
 - Parameter: `$expand=Results`
 - The headers shown below

Key	Value
Accept	application/json
Content-Type	application/json

- Body: {}
- Send the request. If the request is successful, the response contains the 200 OK status code and includes the result of the generic inquiry in JSON format. The following code example shows a fragment of the response body.

```
{
  "id": "51c779ee-a54a-4623-9cac-64ee87b4589e",
  "rowNumber": 1,
  "note": {
    "value": null
  },
  "InventoryID": {},
  "Results": [
    {
      "id": "88e44fc9-96f8-43b4-b93d-225ef1e614ed",
      "rowNumber": 1,
      "note": {
        "value": null
      },
      "Description": {
        "value": "SIM card & contract"
      },
      "InventoryID": {
        "value": "SIMCARD"
      },
      "QtyAvailable": {
        "value": 1000.000000
      },
      "QtyOnHand": {
```

```
        "value": 1000.000000
      },
      "Selected": {
        "value": false
      },
      "Warehouse": {
        "value": "YOGI"
      },
      "custom": {},
      "files": []
    },
    ...
  ]
}
```

Related Links

- [Retrieve Data from an Inquiry Form](#)
- [Parameters for Retrieving Records](#)

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 the contract-based REST API. For information on how to retrieve the same data by using the OData interface, see *Lesson 2.2: Retrieving the Quantities of Stock Items* in the *I300 Data Retrieval with OData* 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 the contract-based REST API 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.

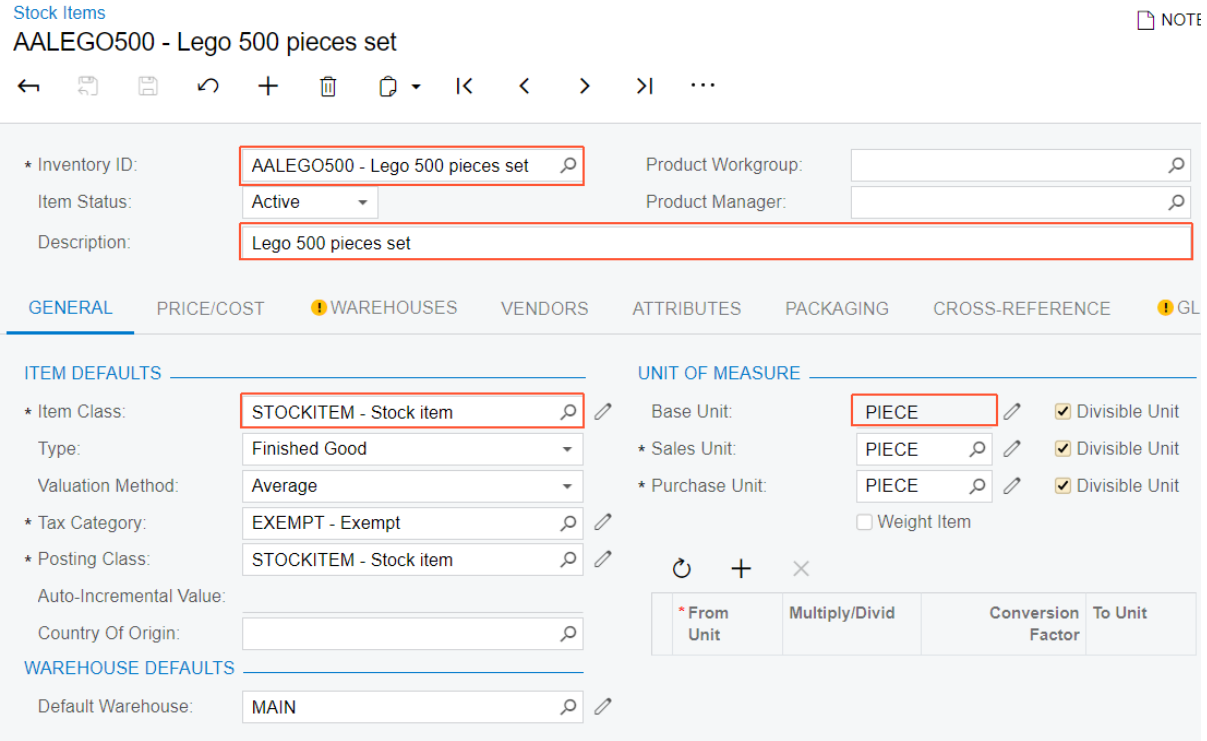


Figure: Elements whose values will be exported from the Summary area and the General tab

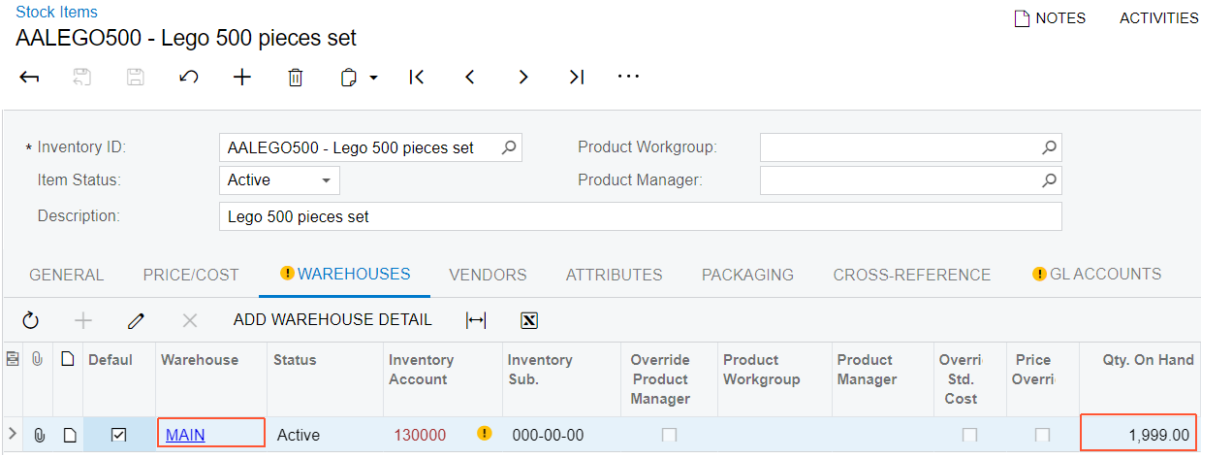


Figure: Elements whose values will be exported from the Warehouse Details tab

This lesson shows how you can implement this scenario by using the contract-based REST API.

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 *AACOMPUT01* and *AALEGO500* inventory items as follows:

- Change the status of the *AACOMPUT01* inventory item to *Inactive* and save the record.
- Change the description of the *AALEGO500* inventory item to *Lego, 500 piece set*, 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 ★ CUSTOMIZATION ▾ TOOLS ▾

⌂ ⏪ ⏩ ⌂

Drag column header here to configure filter

Inventory ID	Stock Item	Item Status	Last Modified On
AALEGO500	<input checked="" type="checkbox"/>	Active	12/25/2018
AACOMPUT01	<input checked="" type="checkbox"/>	Inactive	12/25/2018
<N/A>	<input type="checkbox"/>		9/6/2017
PRINTCOPY	<input type="checkbox"/>	Active	10/24/2016
CONTABLE1	<input checked="" type="checkbox"/>	Active	2/10/2016
AAMACHINE1	<input checked="" type="checkbox"/>	Active	2/9/2016
CONGRILL	<input checked="" type="checkbox"/>	Active	2/9/2016
CALLOUT	<input type="checkbox"/>	Active	8/26/2015
CALLIN	<input type="checkbox"/>	Active	8/26/2015
LAPTOP14	<input checked="" type="checkbox"/>	Active	8/25/2015
LAPTOP12	<input checked="" type="checkbox"/>	Active	8/25/2015
KEYBOARD	<input checked="" type="checkbox"/>	Active	8/25/2015
HEADSET	<input checked="" type="checkbox"/>	Active	8/25/2015
MOUSE	<input checked="" type="checkbox"/>	Active	8/25/2015
MONITOR26	<input checked="" type="checkbox"/>	Active	8/25/2015
LEATHCASE	<input checked="" type="checkbox"/>	Active	8/25/2015

1-16 of 21 records 1 of 2 pages

Figure: Last Modified Date generic inquiry

Example: Using GET and the LastModified Field

In this example, through the REST API, you will configure an HTTP request that exports the list of active stock item records that were modified within the past day.

You will use the GET HTTP method and the `StockItem` entity of the `Default/20.200.001` endpoint to list the stock items. The `StockItem` entity is mapped to the [Stock Items](#) (IN202500) form.

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

Because the database can contain thousands of stock item records, to achieve the best performance of the application, you need to specify the fields of the stock item records that should be returned. You will use the `$select` parameter to specify the fields whose values should be retrieved from Acumatica ERP for each stock item record. For a stock item record, you will export the values of the following fields of the `StockItem` entity:

- `InventoryID`
- `Description`

- ItemClass
- BaseUOM
- WarehouseDetails (nested StockItemWarehouseDetail entity):
 - WarehouseID
 - QtyOnHand

The LastModified and ItemStatus fields will be returned because they are specified in the \$filter parameter.

You will use the \$expand parameter to specify the WarehouseDetails nested detail entity to be returned.

Prerequisites

Before you can test the example, you need to sign in to Acumatica ERP, as described in [Example: Signing In to Acumatica ERP](#). For the request in this example, you have to pass the cookies that you have received during the sign-in.

Instructions for Retrieving the List of Modified Stock Items

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

1. In Postman, configure the following settings of the contract-based REST API request:
 - HTTP method: GET
 - URL: `http://localhost/MyStoreInstance/entity/Default/20.200.001/StockItem`
 - The following parameters of the request

Parameter	Value
\$filter	ItemStatus eq 'Active' and LastModified gt datetimeoffset'2019-11-14T00:00:00.000' Specify today's date instead of 2019-11-14.
\$expand	WarehouseDetails
\$select	InventoryID,Description, WarehouseDetails/WarehouseID, WarehouseDetails/QtyOnHand, ItemClass, BaseUOM

- The headers shown below

Key	Value
Accept	application/json
Content-Type	application/json

2. Send the request. If the request is successful, its response contains the 200 OK status code and includes the list of requested fields of the active stock item records modified within the past day in JSON format. The following code example shows an example of the response body.

```
[
  {
    "id": "cae53ce0-1614-e511-9b82-c86000dddf0b",
    "rowNumber": 1,
  }
]
```

```

    "note": {
      "value": ""
    },
    "BaseUOM": {
      "value": "PIECE"
    },
    "Description": {
      "value": "Lego, 500 piece set"
    },
    "InventoryID": {
      "value": "AALEGO500"
    },
    "ItemClass": {
      "value": "STOCKITEM"
    },
    "ItemStatus": {
      "value": "Active"
    },
    "LastModified": {
      "value": "2019-11-14T06:13:09.57+03:00"
    },
    "WarehouseDetails": [
      {
        "id": "80f56719-6e81-e511-80c0-00155d012302",
        "rowNumber": 1,
        "note": {
          "value": ""
        },
        "QtyOnHand": {
          "value": 1999.000000
        },
        "WarehouseID": {
          "value": "MAIN"
        },
        "custom": {},
        "files": []
      }
    ],
    "custom": {},
    "files": []
  }
]

```

3. Save the request.

Related Links

- [Retrieve Records by Conditions](#)
- [Parameters for Retrieving Records](#)

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 the contract-based REST API. For information on how to retrieve the same data

by using the OData interface, see *Lesson 3.1: Retrieving the List of Modified Stock Items* in the *I300 Data Retrieval with OData* 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.

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.

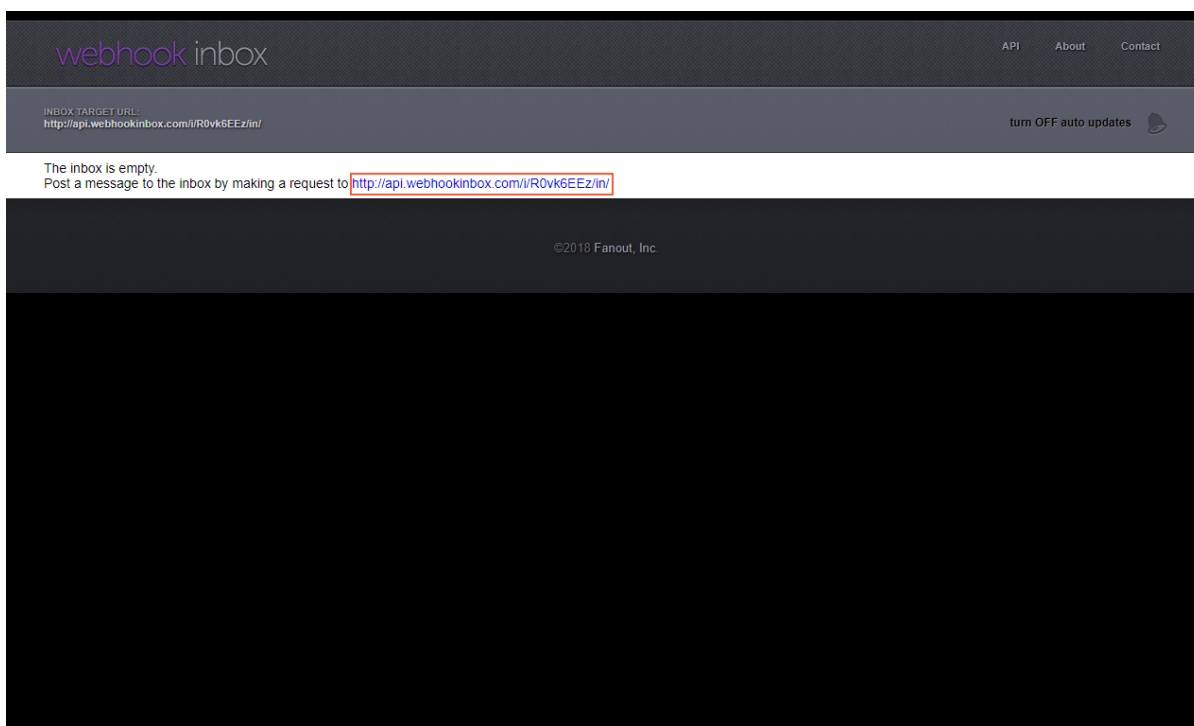


Figure: An HTTP address

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="Left Arrow"/> <input type="button" value="Right Arrow"/>			<input type="button" value="Refresh"/> <input type="button" value="+"/> <input type="button" value="X"/> <input type="button" value="Left Arrow"/> <input type="button" value="Right Arrow"/>	
Active	Inquiry Title	Track All Fields	Table Name	Field Name
> <input checked="" type="checkbox"/>	Item Availability Data	<input checked="" type="checkbox"/> <input checked="" type="button" value="!"/>		

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, the SignalR Hub, a Commerce Push Destination

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](#). For details about how to define push notifications for commerce push destination, see [To Implement Push Notifications for a Commerce Connector](#).

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 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 contract-based REST API, 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:

- The generic inquiry that is used in the integration application
- The custom endpoint
- 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 Contract-Based REST API Integration

In this example, you will create a customization package for the contract-based REST API integration that you have implemented during this course.

You will include in the customization package the following items:

- The *ItemAvailabilityData/0001* custom endpoint, which was used in [Lesson 2.2: Retrieving the Quantities of Stock Items](#) to retrieve the data from the Item Availability Data (INGI0002) generic inquiry
- The Item Availability Data (INGI0002) generic inquiry, which was used in [Lesson 2.2: Retrieving the Quantities of Stock Items](#) and [Example: Configuring Push Notifications](#)
- The MyBIIntegration push notification definition, 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:
MyBIIntegrationCBAPI.
 - c. On the form toolbar, click **Save**.
2. In the **Project Name** column, click the *MyBIIntegrationCBAPI* link, which opens the Customization Project Editor for the *MyBIIntegrationCBAPI* customization project.
3. Include the custom endpoint in the customization project as follows:
 - a. In the navigation pane of the Customization Project Editor, click **Web Service Endpoints** to open the Web Service Endpoints page.
 - b. On the toolbar of the Web Service Endpoints page, click **Add New Record**.
 - c. In the **Add Entity Endpoint** dialog box, which opens, select the check box in the row for the *ItemAvailabilityData* endpoint with Version *0001*, as shown in the following screenshot.

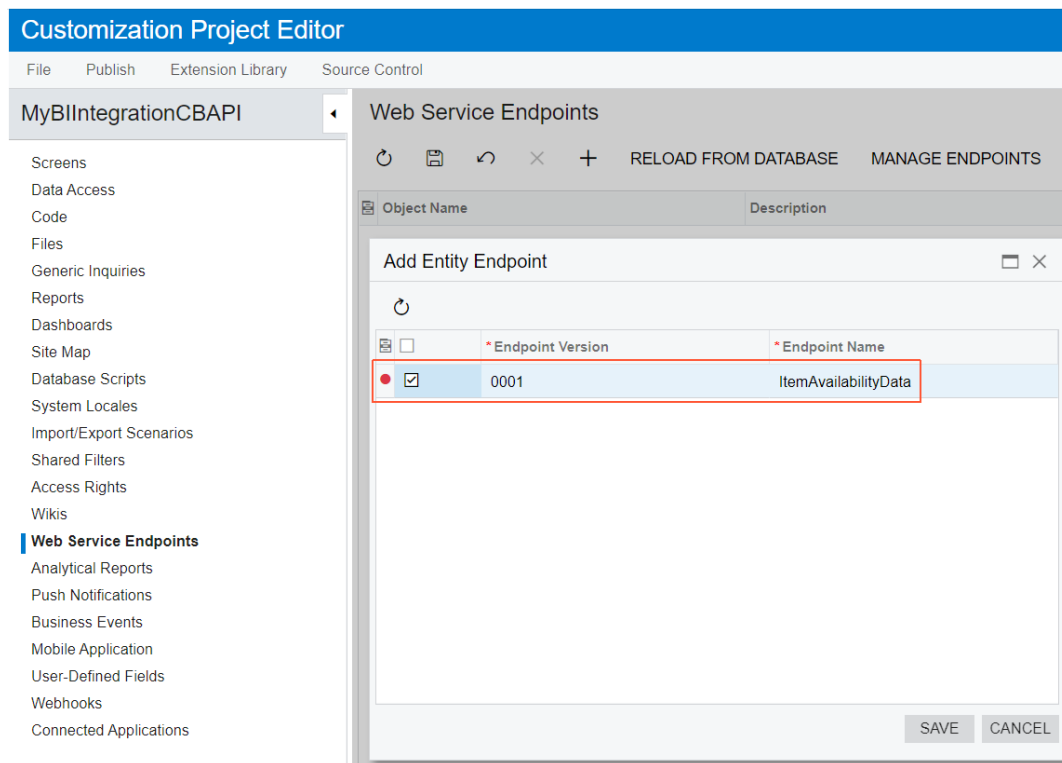


Figure: Selecting the endpoint

- d. Click **Save**.

The endpoint has been added to the Web Service Endpoints page.
4. Include the generic inquiry in the customization project as follows:
 - a. In the navigation pane of the Customization Project Editor, click **Generic Inquiries**.
 - b. On the toolbar of the Generic Inquiries page, click **Add New Record**.
 - c. In the **Add Generic Inquiries** dialog box, select the check box in the row for the *Item Availability Data* generic inquiry.
 - d. Click **Save**.

The inquiry has been added to the Generic Inquiries page.
5. 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.

6. Close the Customization Project Editor.
7. On the Customization Projects form, click the row for the *MyBIIntegrationCBAPI* 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](#)

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 *1300 Data Retrieval with OData*.

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:

- **Monitoring item availability in warehouses:** [Lesson 3.2: Monitoring Item Availability with Push Notifications.](#)
- **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 and Signing Out from Acumatica ERP.](#)
- **Signing out:** [Lesson 1.1: Signing In to and Signing Out from Acumatica ERP.](#)

Appendix: Troubleshooting

I get the error *API Login Limit* when I try to sign in to Acumatica ERP through the REST API. What should I do?

For an application that uses the REST API methods for the sign-in to Acumatica ERP (that is, uses cookies), this error appears if all of the following are true:

- The API login limit is specified in the Acumatica ERP license. The license restriction for the API users is shown in the **Maximum Number of Web Services API Users** box on the **License** tab of the [License Monitoring Console](#) (SM604000) form.
- The number of unclosed sessions (that is, the sessions in which you have signed in to Acumatica ERP through the REST API or obtained access to the Acumatica ERP REST API through OAuth 2.0 and have not signed out from Acumatica ERP) equals the API login limit in the license.
- You try to sign in to Acumatica ERP once more in another session.

You can deal with this error as follows:

1. Modify the code of your application so that it closes the session (signs out from Acumatica ERP) each time the work with Acumatica ERP is finished. For details, see [Lesson 1.1: Signing In to and Signing Out from Acumatica ERP](#).
2. If the integration application has not closed the session for some reason, you can do one of the following:
 - Sign out from Acumatica ERP by using the cookies that the application used during the signing in.
 - Wait for ten minutes until the expiration of the session that is managed by cookies.
 - Restart the site in the Internet Information Services (IIS) Manager or by clicking the **Restart Application** button on the toolbar of the [Apply Updates](#) (SM203510) form.

I get the error *HTTP 404 Not Found* when I click **View Endpoint Service** on the **Web Service Endpoints (SM207060)** form. What should I do?

This error can appear if the Acumatica ERP website is running on a 64-bit operating system, is using the *Classic* managed pipeline mode for the application pool, and is using settings for this application pool that are not recommended. Do the following to make sure you have the recommended IIS settings:

1. In the Control Panel, open **Administrative Tools > Internet Information Services (IIS) Manager**.
2. In **Application Pools**, select the application pool that your Acumatica ERP application uses, and click **Advanced Settings** in the **Actions** pane.
3. In the **Advanced Settings** dialog box, which opens, make sure the **Enable 32-Bit Applications** setting is *False*.

I get the error *The given key was not present in the dictionary* when I send a **GET** request through the REST API. What can be the reason for the error?

This error appears if an incorrect name is specified for the entity or field in the parameters of the request. Make sure that you have specified the correct names for the entities and fields in the parameters of the request.