



I320 Advanced Data Retrieval with REST API

Ruchika Sharma

Technical Account Manager

Timing and Agenda

May 18, 2022 -10 AM -11 AM

Day 1

Lesson 1.1: Registering the Application in Acumatica ERP

Lesson 1.2: Configuring the Application to Use OAuth 2.0

Lesson 1.3: Signing Out from Acumatica ERP

Lesson 2.1: Retrieving a List of Sales Orders with Details and Related Shipments

May 19, 2022 -10 AM -11 AM

Day 2

Lesson 2.2: Retrieving a List of Sales Orders in Batches

Lesson 2.3: Retrieving the List of Payments One by One

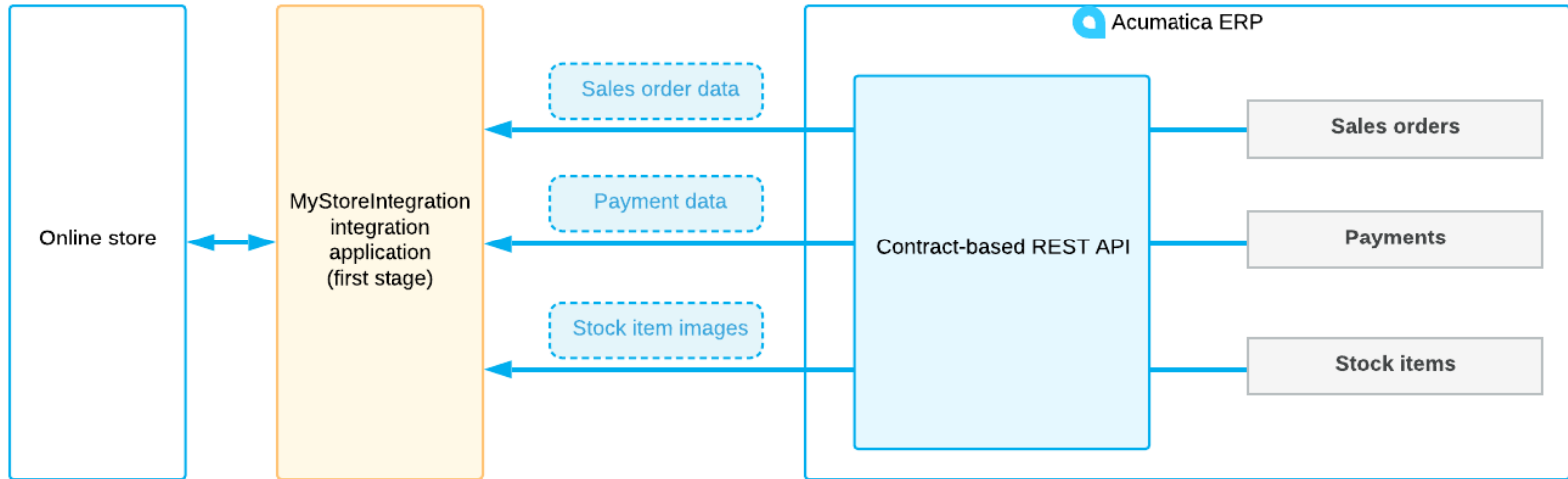
Lesson 3.1: Retrieving the Attachments of a Stock Item

Day 1



Part 1: Authorization of the Application to Work with the Web Service

Company Story and MyStoreIntegration Application



Lesson 1.1: Registering the Application in Acumatica ERP

Learning Objective

In this lesson, you will learn how to register a client application in Acumatica ERP.

Lesson Summary

In this lesson, you have learned how to register in Acumatica ERP a client application that uses the OAuth 2.0 authorization. During the registration, you have been signed in to the MyStore tenant, whose data the MyStoreIntegration application needs to access.

You have also reviewed the possible options of the OAuth 2.0 authorization.

The following table summarizes the availability of the OAuth 2.0 authorization method for each of the integration interfaces.

Integration Interface	OAuth 2.0 Authorization
OData Version 3.0 interface	Yes
OData Version 4.0 interface	Yes
REST API	Yes

Lesson 1.2: Configuring the Application to Use OAuth 2.0

Learning Objective

In this lesson, you will learn how to configure an integration application to use OAuth 2.0 for authorization in Acumatica ERP.

Lesson Summary

In this lesson, you have learned how to configure an integration application to use OAuth 2.0 for authorization in Acumatica ERP. You have connected to the token endpoint, passed the client ID and client secret in the authorization header, and requested access to the web service APIs. You have received the access token from Acumatica ERP. You will use this token in subsequent requests to Acumatica ERP.

Lesson 1.3: Signing Out from Acumatica ERP

Learning Objective

In this lesson, you will learn how to sign out from Acumatica ERP in an OAuth 2.0 application.

Signing Out from Acumatica ERP

Key	Value
Accept	<code>application/json</code>
Content-Type	<code>application/json</code>

Lesson Summary

In this lesson, you have learned how to sign out from Acumatica ERP in an OAuth 2.0 application. You have also reviewed whether the sign-out is necessary for the OAuth 2.0 applications.



Part 2: Performance Optimization

Lesson 2.1: Retrieving a List of Sales Orders with Details and Related Shipments

Learning Objective

In this lesson, you will learn how to retrieve from Acumatica ERP records with multiple kinds of details.

Figure: The Summary area and the Document Details tab

Sales Orders

SO 000001 - Jevy Computers

NOTES ACTIVITIES FILES



* Order Type:	<input type="text" value="SO"/>	Customer:	<input type="text" value="C000000003 - Jevy Computers"/>	Ordered Qty.:	<input type="text" value="1.00"/>
Order Nbr.:	<input type="text" value="000001"/>	Contact:		Discount Total:	<input type="text" value="0.00"/>
Status:	Shipping			Tax Total:	<input type="text" value="0.00"/>
Date:	<input type="text" value="10/28/2015"/>			Order Total:	<input type="text" value="2,200.00"/>
Requested On:	10/28/2015	Description:			
Customer Ord...:	<input type="text" value="SO180-009-01"/>				
External Refer...					

DETAILS TAXES COMMISSIONS FINANCIAL SHIPPING ADDRESSES SHIPMENTS PAYMENTS TOTALS

ADD ITEMS ADD INVOICE ADD BLANKET SO LINE ITEM AVAILABILITY

* Branch	* Inventory ID	Free Item	Warehouse	Line Description	* UOM	Quantity	Qty. On Shipments	Open Qty.	Unit Price
MYSTORE	<input type="text" value="AAMACHINE1"/>	<input type="checkbox"/>	MAIN	Injection molding machine	PIECE	<input type="text" value="1.00"/>	1.00	1.00	<input type="text" value="2,200.0000"/>

Figure: The Shipments tab

Sales Orders

SO 000001 - Jevy Computers

NOTES ACTIVITIES FILES



* Order Type:	SO	Customer:	C000000003 - Jevy Computers	Ordered Qty.:	1.00
Order Nbr.:	000001	Contact:		Discount Total:	0.00
Status:	Shipping			Tax Total:	0.00
Date:	10/28/2015			Order Total:	2,200.00
Requested On:	10/28/2015	Description:			
Customer Ord...:	SO180-009-01				
External Refer...:					

DETAILS TAXES COMMISSIONS FINANCIAL SHIPPING ADDRESSES **SHIPMENTS** PAYMENTS TOTALS



Shipment Type	Document Nbr.	Status	* Shipment Date	Shipped Qty.	Shipped Weight	Shipped Volume	Invoice Type	Invoice Nbr.	Inventory Doc.
Shipment	000001	On Hold	11/2/2015	1.00	0.000000	0.000000			

Retrieving the List of Sales Orders

- The following parameters of the request

Parameter	Value
\$filter	CustomerID eq 'C000000003'
\$expand	Details,Shipments
\$select	OrderNbr,OrderType,CustomerID,CustomerOrder,Details/InventoryID,Details/OrderQty,Details/UnitPrice,Date,OrderedQty,OrderTotal,Shipments/InvoiceNbr,Shipments/ShipmentNbr

- The headers shown below

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

Lesson Summary

In this lesson, you have added to the MyStoreIntegration REST application the requests that retrieve the list of sales orders with details and related shipments from Acumatica ERP. You have used one request to retrieve two kinds of details, which optimizes the performance of the application. You have also reviewed how this scenario can be implemented with the OData interface.

The following table summarizes the availability of the performance optimization options for different integration interfaces.

Integration Interface	Optimized Retrieval of Multiple Kinds of Detail Lines
OData Version 3.0 interface	Yes, if you create multiple custom generic inquiries (one inquiry for each kind of detail line)
OData Version 4.0 interface	Yes
REST API	Yes

Day 2

Lesson 2.2: Retrieving a List of Sales Orders in Batches

Learning Objective

In this lesson, you will learn how to retrieve records in batches from Acumatica ERP.

Retrieving the List of Sales Orders in Batches

- The following parameters of the request

Parameter	Value
<code>\$filter</code>	<code>CustomerID eq 'C000000003'</code>
<code>\$select</code>	<code>OrderNbr, OrderType, CustomerID, OrderTotal</code>

Parameter	Value
<code>\$top</code>	<code>5</code>

- The headers shown below

Key	Value
<code>Accept</code>	<code>application/json</code>
<code>Content-Type</code>	<code>application/json</code>

Lesson Summary

In this lesson, you have added to the MyStoreIntegration application the request that retrieves the list of sales orders from Acumatica ERP in batches of five records. You have implemented this scenario by using the contractbased REST API. You have used the \$top and \$skip parameters of the REST request. You have also reviewed how this scenario can be implemented with the OData interface.

Lesson 2.3: Retrieving the List of Payments One by One

Learning Objective

In this lesson, you will learn how to deal with the errors that can occur if the performance optimization fails.

Retrieving the List of Payments

- The following parameters of the request

Parameter	Value
<code>\$filter</code>	<code>Type eq 'Payment' and CustomerID eq 'C000000003'</code>
<code>\$expand</code>	<code>ApplicationHistory</code>
<code>\$select</code>	<code>ReferenceNbr,Type,Status,ApplicationHistory/DisplayDocType,ApplicationHistory/DisplayRefNbr,ApplicationDate</code>

- The headers shown below

Key	Value
<code>Accept</code>	<code>application/json</code>
<code>Content-Type</code>	<code>application/json</code>

Figure: The returned error

GET <https://localhost/MyStoreInstance/entity/Default/18.200.001/Payment?sselect=ReferenceNbr,Type,Status,ApplicationDate,ApplicationHistory/DisplayDocType,ApplicationHistory/DisplayR...> Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> \$select	ReferenceNbr,Type,Status,ApplicationDate,ApplicationHistory/DisplayDocTy...	
<input checked="" type="checkbox"/> \$filter	Type eq 'Payment' and CustomerID eq 'C000000003'	
<input checked="" type="checkbox"/> \$expand	ApplicationHistory	
Key	Value	Description

Body Cookies (6) Headers (8) Test Results Status: 500 Internal Server Error Time: 11.32s Size: 4.6 KB Save Response

Pretty Raw Preview Visualize BETA JSON

```
1 {
2   "message": "An error has occurred.",
3   "exceptionMessage": "Optimization cannot be performed.The following fields cause the error:\r\nApplicationHistory.DisplayDocType: View Adjustments_History has BQL
4     delegate\r\nApplicationHistory.DisplayRefNbr: View Adjustments_History has BQL delegate\r\n",
5   "exceptionType": "PX.Api.ContractBased.OptimizedExport.CannotOptimizeException",
  "stackTrace": " at PX.Api.ContractBased.OptimizedExport.NotWorkingOptimizedExportProvider.get_CanOptimize()\r\n at PX.Api.ContractBased.EntityService.GetList(ISystemContract systemContract,
String version, String name, EntityImpl entity, CbOperationContext operationContext, Boolean ignoreValueFields, PXGraph graph)\r\n at PX.Api.ContractBased.Soop.SoapFacadeBase.GetListImpl
(Entity entity)\r\n at PX.Api.ContractBased.SystemContracts.V2.RestController.GetList(String objectName, String select, String filter, String expand, String custom, Nullable`1 skip,
Nullable`1 top)\r\n at lambda_method(Closure , Object , Object[] )\r\n at System.Web.Http.Controllers.ReflectedHttpActionDescriptor.ActionExecutor.<>c__DisplayClass10.<GetExecutor>b__9
(Object instance, Object[] methodParameters)\r\n at System.Web.Http.Controllers.ReflectedHttpActionDescriptor.ExecuteAsync(HttpControllerContext controllerContext, IDictionary`2 arguments,
CancellationToken cancellationToken)\r\n--- End of stack trace from previous location where exception was thrown ---\r\n at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()
\r\n at System.Web.Http.Controllers.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)\r\n at
System.Web.Http.Controllers.ApiControllerActionInvoker.<InvokeActionAsyncCore>d__0.MoveNext()\r\n--- End of stack trace from previous location where exception was thrown ---\r\n at
System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n at System.Web.Http.Filters.ActionFilterAttribute.<CallOnActionExecutedAsync>d__5.MoveNext()\r\n--- End of stack trace from
previous location where exception was thrown ---\r\n at System.Web.Http.Filters.ActionFilterAttribute.<CallOnActionExecutedAsync>d__5.MoveNext()\r\n--- End of stack trace
from previous location where exception was thrown ---\r\n at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n at
System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)\r\n at
System.Web.Http.Filters.ActionFilterAttribute.<ExecuteActionFilterAsyncCore>d__0.MoveNext()\r\n--- End of stack trace from previous location where exception was thrown ---\r\n at
System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n at System.Web.Http.Controllers.ActionFilterResult.<ExecuteAsync>d__2.MoveNext()\r\n--- End of stack trace from previous location where exception was thrown ---\r\n at
System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)\r\n at
System.Web.Http.Filters.AuthorizationFilterAttribute.<ExecuteAuthorizationFilterAsyncCore>d__2.MoveNext()\r\n--- End of stack trace from previous location where exception was thrown ---\r\n at
System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)\r\n at
System.Web.Http.Controllers.ExceptionFilterResult.<ExecuteAsync>d__0.MoveNext()\r\n--- End of stack trace from previous location where exception was thrown ---\r\n at
System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n at System.Web.Http.Controllers.ExceptionFilterResult.<ExecuteAsync>d__0.MoveNext()\r\n--- End of stack trace from
previous location where exception was thrown ---\r\n at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw()\r\n at
System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification(Task task)\r\n at System.Web.Http.Dispatcher.HttpControllerDispatcher.<SendAsync>d__1.MoveNext()"
```

Retrieving the List of Payments

3. Modify the parameters of the request to return only the key fields of the payments as follows.

Parameter	Value
<code>\$filter</code>	<code>Type eq 'Payment' and CustomerID eq 'C000000003'</code>
<code>\$select</code>	<code>ReferenceNbr,Type</code>

6. Configure the request that retrieves the details of the payments by the values of the key fields (ReferenceNbr and Type) as follows:

- HTTP method: GET
- URL: `https://localhost/MyStoreInstance/entity/Default/20.200.001/Payment/Payment/000001`
- The following parameters of the request

Parameter	Value
<code>\$select</code>	<code>ReferenceNbr,Type,Status,ApplicationHistory/DisplayDocType,ApplicationHistory/DisplayRefNbr,ApplicationDate</code>
<code>\$expand</code>	<code>ApplicationHistory</code>

Lesson Summary

In this lesson, you have added to the MyStoreIntegration REST application the methods that retrieve the payments of a customer one by one. You have used this approach because the retrieval of the payments with the specified fields could not be optimized for performance.



Part 3: Retrieval of Attachments

Lesson 3.1: Retrieving the Attachments of a Stock Item

Learning Objective

In this lesson, you will learn how to retrieve the files that are attached to a stock item by using the contract-based REST API.

Figure: Attached file

The screenshot displays the 'AAMACHINE1 - Injection molding machine' record in an ERP system. The 'FILES (1)' tab is active, showing a modal window with the following details:

- Inventory ID: AAMACHINE1 - Injection molding mac
- Product Workgroup: [Empty]
- Item Status: Active
- Description: Injection molding machine

The modal window contains a table of attached files:

File name	Comment	Last Date	EditUrl
T2MCRQ.jpg		1/1/1900	Edit

Additional modal controls include: 'Select the file.' input, 'Browse' and 'Upload' buttons, and actions: 'ADD LINK', 'REMOVE LINK', and 'UPLOAD USING MOBILE APP'.

Retrieving the Attachments of a Stock Item

- The headers shown below

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

Lesson Summary

In this lesson, you have learned how to export the files that are attached to a record.

The following table summarizes whether the attachment can be retrieved from Acumatica ERP through the use of the integration interfaces.

Integration Interface	Retrieval of Attachments
OData Version 3.0 interface	No
OData Version 4.0 interface	No
REST API	Yes

Your feedback is appreciated

<https://www.surveymonkey.com/r/OnlineSessions2022>



No Reliance

This document is subject to change without notice. Acumatica cannot guarantee completion of any future products or program features/enhancements described in this document, and no reliance should be placed on their availability.

Confidentiality: This document, including any files contained herein, is confidential information of Acumatica and should not be disclosed to third parties.



Thank you

Ruchika Sharma