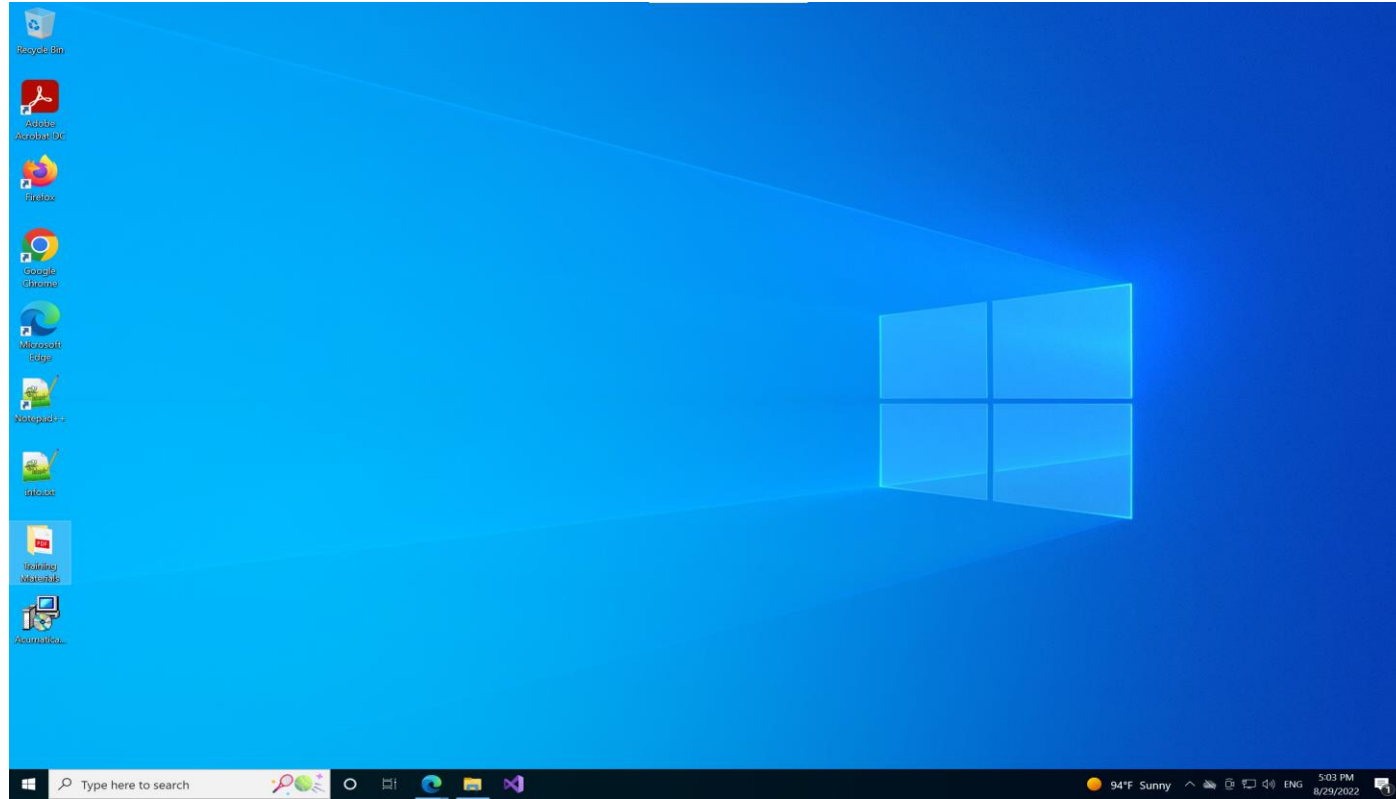


T230 Actions

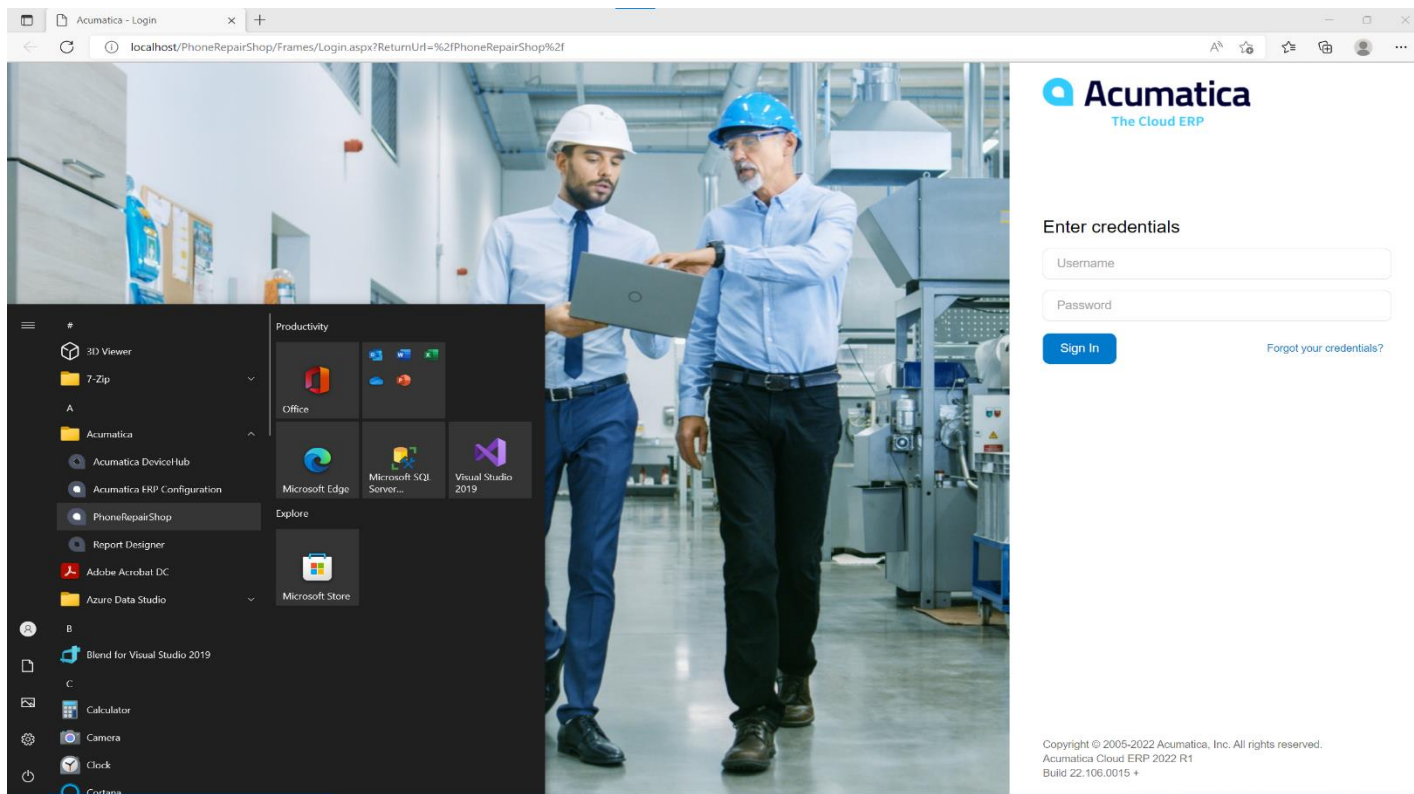
Artem Altukhov

Software Developer

Training Materials



Launching your Acumatica instance



Copyright © 2005-2022 Acumatica, Inc. All rights reserved.
Acumatica Cloud ERP 2022 R1
Build 22.106.0015 +

In The Previous Webinars

- **T200 Maintenance Forms training course.** You have created two simple maintenance forms: Repair Services (RS201000) and Serviced Devices (RS202000)
- **T210 Customized Forms and Master-Detail Relationship course.** You have created another maintenance form, Services and Prices (RS203000), and customized the Stock Items (IN202500) form of Acumatica ERP.
- **T220 Data Entry and Setup Forms course.** You have created the Repair Work Orders (RS301000) data entry form, which is used to create and manage work orders for repairs.

Timing and Agenda

August 30, 2022 -10:00 AM -11:30 AM

Day 1

Lesson 1: Define an Action and the Associated Button on the Form Toolbar

Lesson 2: Define Actions and the Associated Buttons on the Table Toolbar

August 31, 2022 -10:00 AM -11:30 AM

Day 2

Lesson 3: Implement an Asynchronous Operation

Lesson 4: Define a Link to an Acumatica ERP Entity

A close-up photograph of a hand pouring beer from a metal tap into a tall glass. The glass features a logo with a spade and the text "Devil's Peak". The background is a blurred bar or brewery setting.

Day 1

Lesson 1: Define an Action and the Associated Button on the Form Toolbar

Learning Objectives

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

- Create an action, the associated button on the form toolbar, and the equivalent command on the More menu
- Configure the availability and visibility of the button and command depending on field values on the form

Types of Actions

You typically use actions for the following purposes:

- To redirect a user to a specific form or report.
- To modify or validate data records and save changes to the database
- To start a background operation, which is executed on a separate thread

Figure: The Repair Work Orders form

Repair Work Orders

000003 - Battery Replacement

NOTES FILES CUSTOMIZATION TOOLS

← ↻ 📄 ↺ + 🗑️ 📋 ⏪ < > ⏩ REMOVE HOLD **ASSIGN TO ME** ...

Order Nbr.: 000003 * Customer ID: C000000001 - Jersey Central Office E Order Total: 35.00

Status: On Hold * Service: BATTERYREPLACE - Battery Replace Invoice Nbr.:

* Date Created: 10/29/2021 * Device: NOKIA3310 - Nokia 3310

Date Completed: Assignee:

Priority: Medium Description: Battery replacement, Nokia 3310

REPAIR ITEMS LABOR

↻ + ✎ × ⏪ ⏩

📄	🔍	📄	Repair Item Type	Inventory ID	Description	Price
>	🔍	📄	Battery	BAT3310	Battery for Nokia 3310	20.00
	🔍	📄	Back Cover	BCOV3310	Back cover for Nokia 3310	10.00

⏪ < > ⏩

Changing of Appearance of Buttons on the UI

To disable or enable an action button, you need to use the **SetEnabled** method:

```
MyAction.SetEnabled(false)
```

To configure the visibility of the corresponding command, you need to use the **SetVisible** method:

```
MyAction.SetVisible(false)
```

Changing of Category of Buttons

You can change the category of the command by setting the value of the Category property of the PXButton attribute:

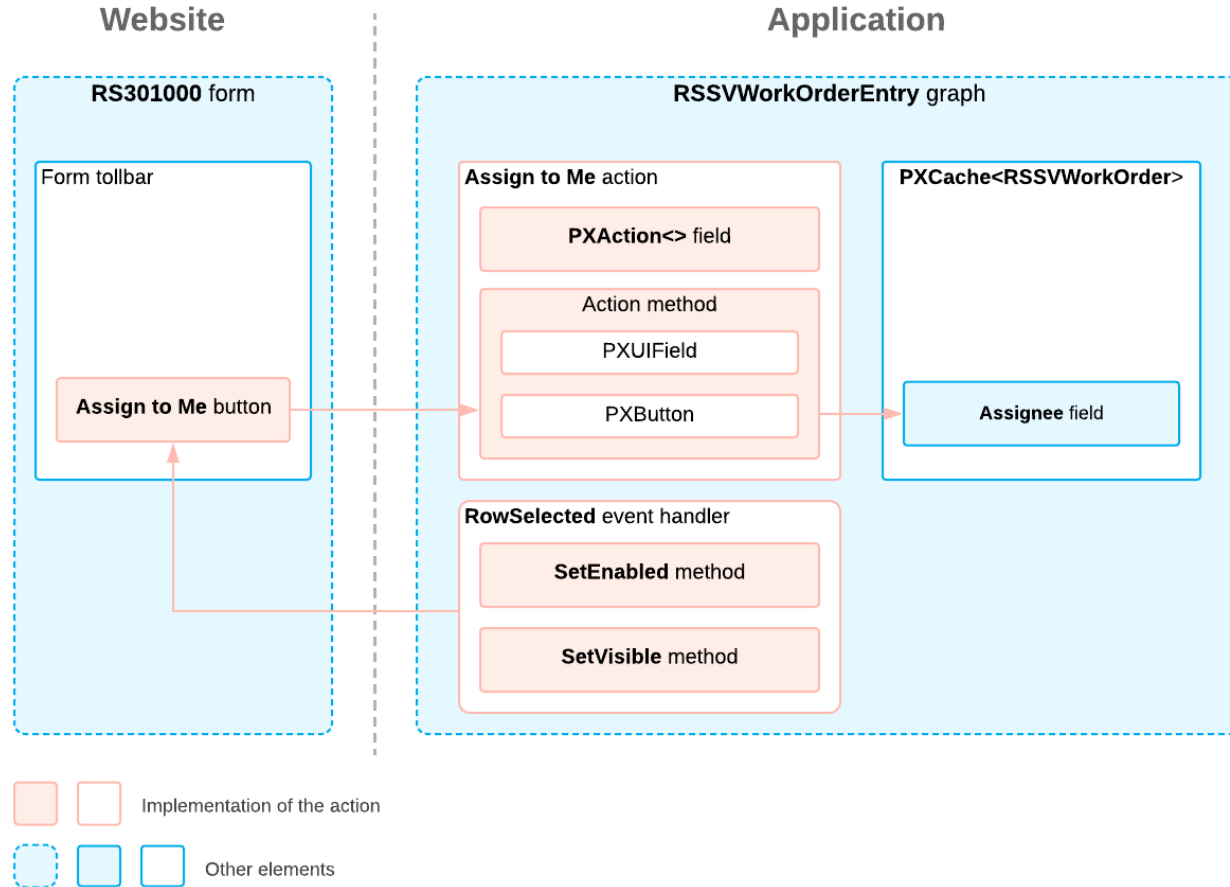
```
[PXButton(Category = "Test Category")]
```

Lesson Summary

In this lesson, you have learned how to define an action in Acumatica Framework and display the corresponding button on the form toolbar and command on the More menu. You have implemented the action, configured the availability of the button and command, and tested the action and button.

To define an action, you have added the field of the `PXAction<>` type and an action method with the `PXButton` and `PXUIField` attributes to the `RSSVWorkOrderEntry` graph. To configure the availability and visibility of the corresponding command, you have used the `SetEnabled` and `SetVisible` methods in the `RowSelected` event handler of the `RSSVWorkOrderEntry` graph.

Implementation of the Action



Lesson 2: Define Actions and the Associated Buttons on the Table Toolbar

Learning Objectives

In this lesson, you will learn how to specify the location of a button associated with an action on a form.

Figure: The Update Prices button on the Repair Items tab

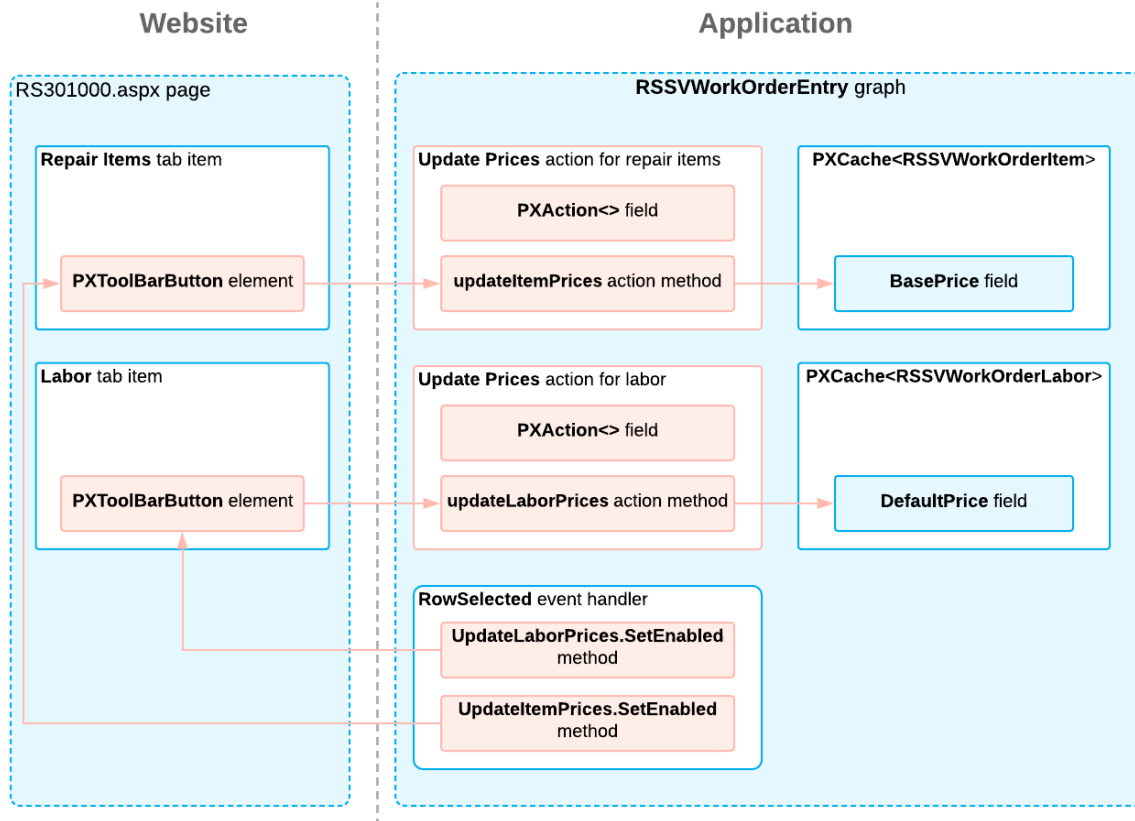
REPAIR ITEMS						LABOR	
						UPDATE PRICES	
		Repair Item Type	Inventory ID	Description		Price	
>		Battery	BAT3310	Battery for Nokia 3310		20.00	
		Back Cover	BCOV3310	Back cover for Nokia 3310		10.00	


Lesson Summary

In this lesson, you have defined the `UpdateItemPrices` and `UpdateLaborPrices` actions and learned how to place the associated buttons on the table toolbar of a tab. To configure the button's location to be on a table toolbar, you have done the following:

- Hidden the buttons from the form toolbar and the equivalent commands from the More menu by setting the `DisplayOnMainToolbar` property of the `PXButton` attribute to `false`
- In the ASPX file of the form, added the `ActionBar`, `CustomItem`, `PXToolBarButton`, and `AutoCallBack` elements to the `PXGrid` section of the `PXTabItem` element

Implementation of the Actions



A woman with blonde hair in a ponytail, wearing glasses and a grey t-shirt with an American flag patch, is working at a metalworking machine. She is using a pair of pliers to work on a piece of metal. The machine has a "DANGER" warning label that says "KEEP FINGERS CLEAR OF MACHINE WHILE IN MOTION". To the left of the machine is a rack filled with various small metal parts. The background shows a factory setting with rolls of material.

Day 2

Lesson 3: Implement an Asynchronous Operation

Learning Objectives

In this lesson, you will learn how to implement an asynchronous operation by using the `PXLongOperation` class.

Figure: Item Classes form

Item Classes

NOTES ACTIVITIES FILES CUSTOMIZATION TOOLS

Item Class Tree

- LABOR***** Labor
- NSTOCKITEM Non-stock item
- STOCKITEM* Stock item**

* Class ID: STOCKITEM - Stock item

Description: Stock item

GENERAL RESTRICTION GROUPS ATTRIBUTES

GENERAL SETTINGS

☒ Stock Item

☒ Allow Negative Quantity

☐ Accrue Cost

Item Type: Finished Good

Valuation Method: Average

Tax Category: EXEMPT - Exempt

Posting Class: STOCKITEM - Stock item

Price Class:

Default Warehouse:

* Availability Calculation ... STOCKITEM

Country Of Origin:

SHIPPING THRESHOLDS

Undershhip Threshold (%): 100.00

Overship Threshold (%): 100.00

UNIT OF MEASURE

* Base Unit: PIECE ☒ Divisible Unit

* Sales Unit: PIECE ☒ Divisible Unit

* Purchase Unit: PIECE ☒ Divisible Unit

* From Unit	Multiply/Divide	Conversion Factor	To Unit

PRICE MANAGEMENT

Price Workgro...

Price Manager:

Min. Markup %: 0.00

Instantiate graphs from code

To instantiate graphs from code, use the **PXGraph.CreateInstance()** method.

```
var invoiceGraph =  
PXGraph.CreateInstance<SOInvoiceEntry>();
```

Invoke a method on a separate thread

To make the system invoke a method on a separate thread, you can use the **PXLongOperation.StartOperation()** method

```
PXLongOperation.StartOperation(this, delegate ()  
{  
    MyMethod(...);  
});
```

Figure: Creation of an invoice

Repair Work Orders

000003 - Battery Replacement

← ↻ 📄 ↶ + 🗑️ 📄 ▾ ⏪ < > ⏩ CREATE INVOICE ...

NOT

Executing. Press to abort
00:00:17

CANCEL

Order Nbr.: 000003 Customer ID: C000000001 - Jersey Central Office Equip Order Total: 45.00

Status: Completed Service: BATTERYREPLACE - Battery Replacement Invoice Nbr.:

* Date Created: 10/29/2021 Device: NOKIA3310 - Nokia 3310

Date Completed: 11/10/2021 Assignee: Andrews, Michael

Priority: Medium Description: Battery replacement, Nokia 3310

REPAIR ITEMS LABOR

↻ + ✎ ✕ UPDATE PRICES ⏪ ⏩

📄	🔍	📄	Repair Item Type	Inventory ID	Description	Price
>	🔍	📄	Battery	BAT3310	Battery for Nokia 3310	25.00
	🔍	📄	Back Cover	BCOV3310	Back cover for Nokia 3310	10.00

Figure: Update of the Invoice Nbr box

Repair Work Orders

000003 - Battery Replacement

NOT

✓ The operation has completed.

Order Nbr.: 000003 Customer ID: C000000001 - Jersey Central Office Equip Order Total: 45.00

Status: Completed Service: BATTERYREPLACE - Battery Replacement Invoice Nbr.: INV000049

* Date Created: 10/29/2021 Device: NOKIA3310 - Nokia 3310

Date Completed: 11/10/2021 Assignee: Andrews, Michael

Priority: Medium Description: Battery replacement, Nokia 3310

REPAIR ITEMS LABOR

UPDATE PRICES

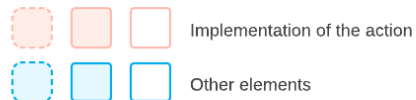
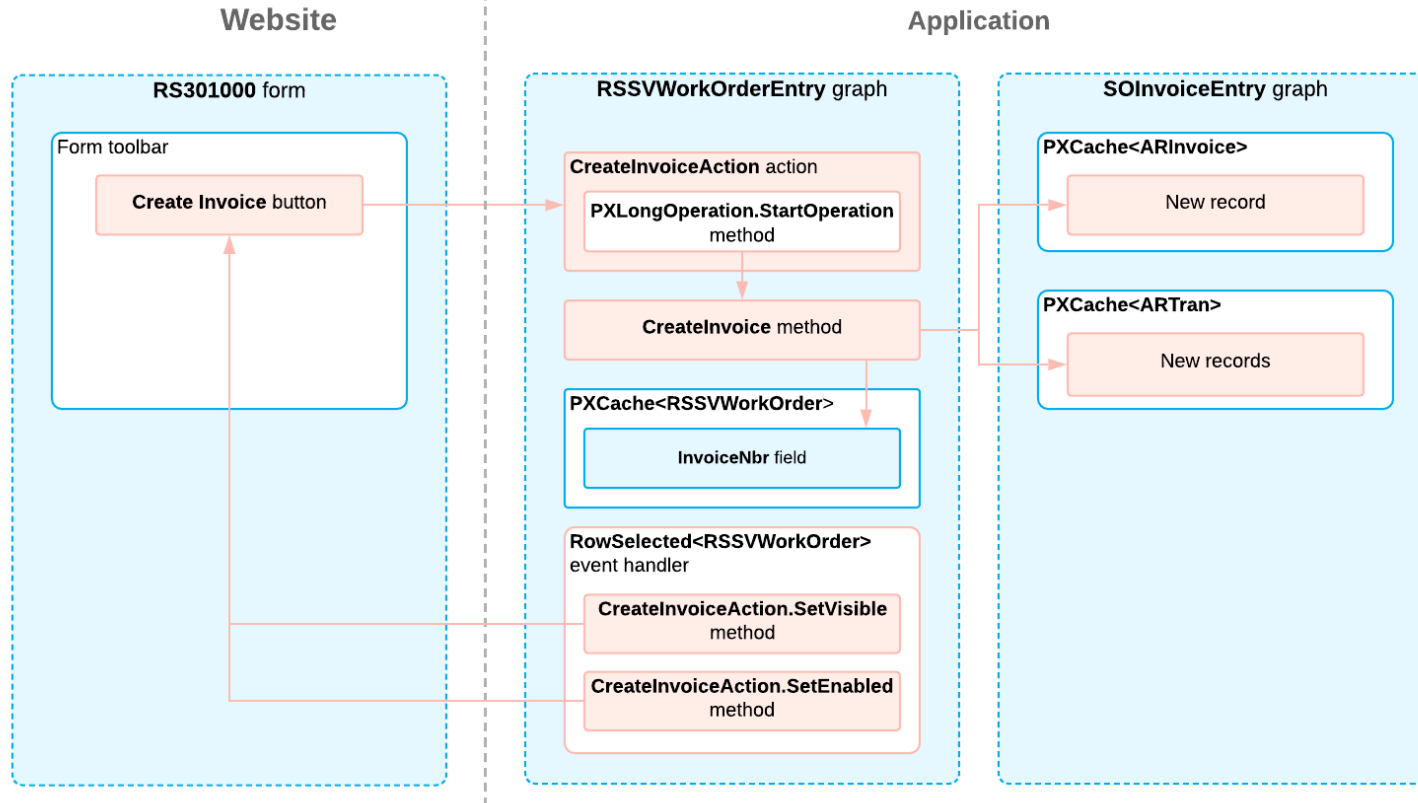
Repair Item Type	Inventory ID	Description	Price
Battery	BAT3310	Battery for Nokia 3310	25.00
Back Cover	BCOV3310	Back cover for Nokia 3310	10.00

Lesson Summary

In this lesson, you have learned how to initiate an asynchronous operation inside an action method by using the `PXLongOperation` class. Also, you have implemented the creation of an SO invoice based on a repair work order by doing the following in the `RSSVWorkOrderEntry` graph:

- Defining the static `CreateInvoice` method, which creates an instance of the `SOInvoiceEntry` graph
- Defining the **Create Invoice** button on the form toolbar and the command with the same name on the More menu; the underlying action initiates the asynchronous execution of the `CreateInvoice` method by using the `PXLongOperation` class
- Specifying the availability of the Create Invoice action in the `RowSelected` event handler so that only a single invoice can be created for a repair work order

Implementation of the Action



Lesson 4: Define a Link to an Acumatica ERP Entity

Learning Objectives

As you complete this lesson, you will learn how to define a link to an Acumatica ERP entity on a form.

Figure: The link to open the related invoice on the Repair Work Orders form

Repair Work Orders

000003 - Battery Replacement

NOTES FILES CUSTOMIZATION TOOLS ▾

← ↺ ↻ + 🗑️ 📄 < > >| ...

Order Nbr.: 000003 Customer ID: C000000001 - Jersey Central Office Equip Order Total: 45.00
Status: Completed Service: BATTERYREPLACE - Battery Replaceme Invoice Nbr.: [INV000049](#)
* Date Created: 3/10/2022 Device: NOKIA3310 - Nokia 3310
Date Completed: 3/11/2022 Assignee: Andrews, Michael
Priority: Medium Description: Battery replacement, Nokia 3310

REPAIR ITEMS LABOR

↺ + ✎ ✕ UPDATE PRICES |⇄| ☒

📄	🔗	📄	Repair Item Type	Inventory ID	Description	Price
>	🔗	📄	Battery	BAT3310	Battery for Nokia 3310	25.00
	🔗	📄	Back Cover	BCOV3310	Back cover for Nokia 3310	10.00

⇐ < > >| natica
The Cloud ERP

Define a Link to an Acumatica ERP Entity

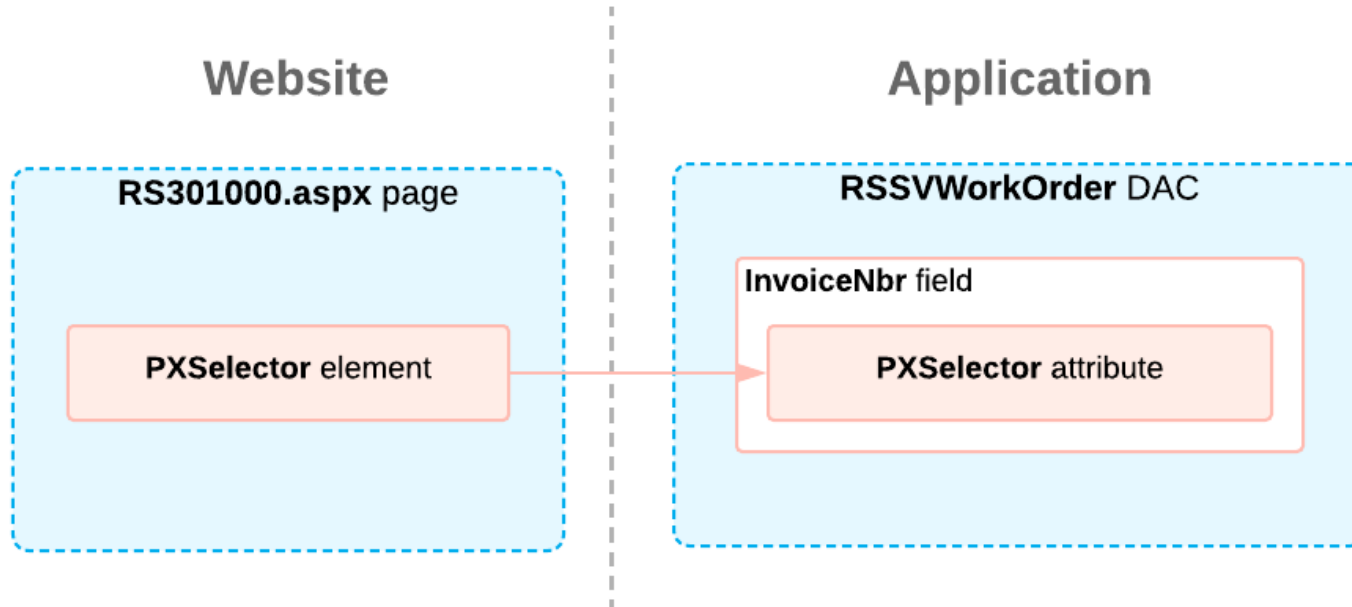
To define a hyperlink that leads to another form:

1. Decorate a DAC field with the **PXSelector** attribute
2. Update ASPX


Lesson Summary

In this lesson, you have configured the **Invoice Nbr.** box to contain a hyperlink that leads to the Invoices (SO303000) form. For this purpose, you have added the PXSelector attribute to the InvoiceNbr field of the RSSVWorkOrder DAC and configured the PXSelector element in the RS301000.aspx file.

Implementation of the Link



  Implementation of the link

 Other elements

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

Artem Altukhov