
# Push Notifications & Webhooks

## Synchronizing changes between Acumatica Instances

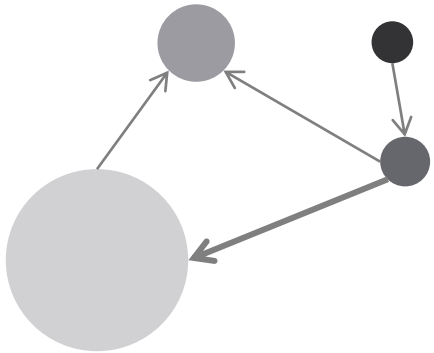
Marco Villaseñor
CTO
*interastar

# Agenda

- Importance of webhooks and notifications
- How to configure a webhook in Acumatica
- How to configure a push notification to a webhook in Acumatica
- Live demo with code!


# Why are webhooks awesome?

- Instant notifications
- Simple to implement
- Composable
- Are widely supported around the web

# Why are webhooks awesome?

- Instant notifications
- Simple to implement
- Composable
- Are widely supported around the web
- Available now in Acumatica 2020R1 !

# But what is a webhook?

It is an integration mechanism that…

…works like a (POST) REST web service endpoint

…lets external applications notify our app when something happened,

so no more polling!

...accepts the caller’s format

**Basically “just send the data to this URL”**

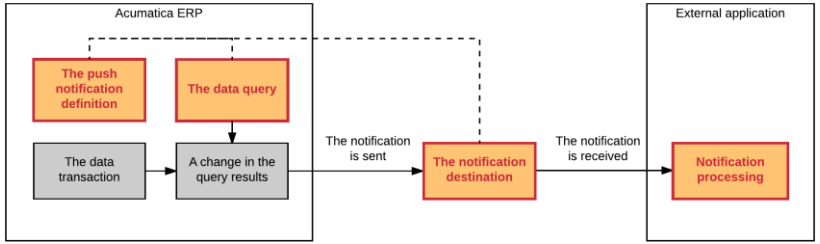
# About Acumatica push notifications

**Feature was first introduced in 2017R2**

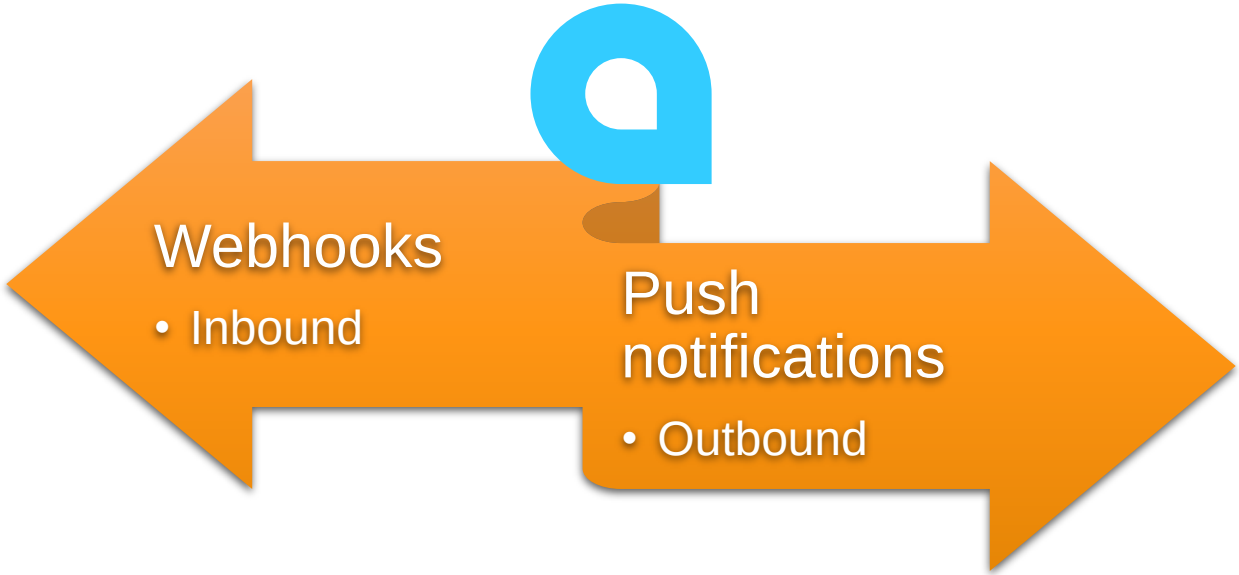
Push notifications are notifications in JSON format that are sent by Acumatica ERP to notification destinations when specific data changes occur in Acumatica ERP.

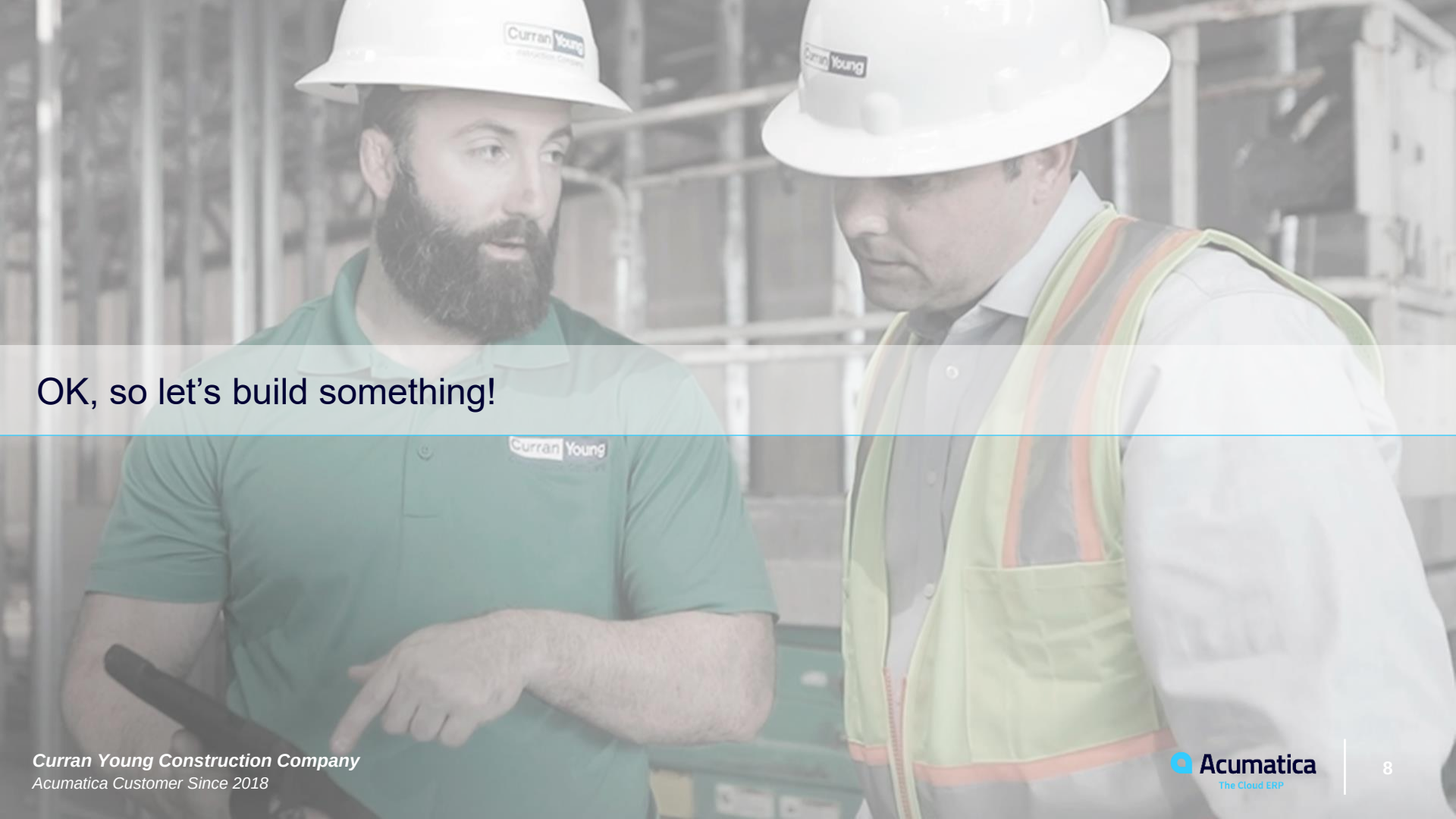
Support **webhooks** as a destination!

push notifications



# Webhooks and Push Notifications in Acumatica



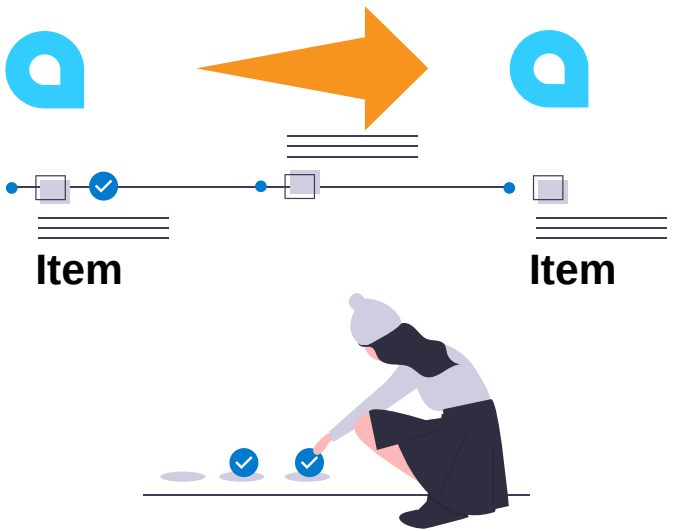


# OK, so let’s build something!

***Curran Young Construction Company***
*Acumatica Customer Since 2018*

Acumatica
The Cloud ERP

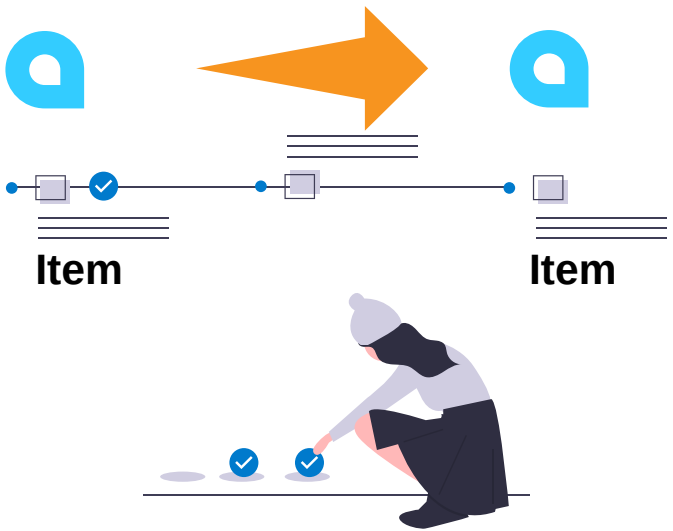
# A simple example…




## Sync non-stock items between 2 Acumatica instances

1. When a new item is created
2. Tell the other instance
3. Create the item

# A simple example…

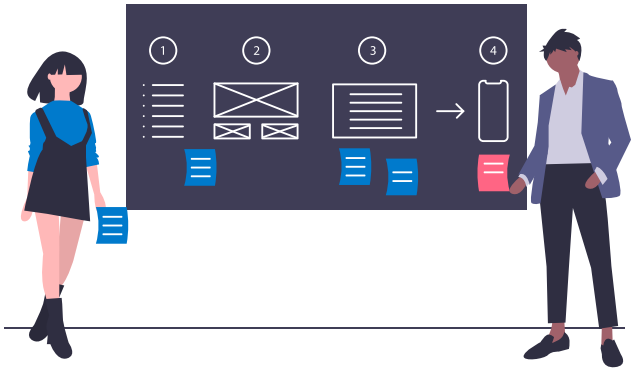



## Sync non-stock items between 2 Acumatica instances

1. Push notifications trigger when there are changes
2. Subscribe to a webhook to change item data
3. Implementation class creates the item

# Demo

# Creating webhooks in Acumatica




1. Get the structure of the notification we will handle
2. Create an implementation class using the **`IWebhookHandler`** interface
3. Register class as a webhook in Acumatica
4. Save the generated URL to 'subscribe' the external system to that webhook

(Screen SM304000)

# Example notification


```
{
  "Inserted": [{
    "InventoryID": "ITEMCD ",
    "Description": “My item’s description",
    "Type": "Non-Stock Item",
    "PostingClass": "FDI",
    "ItemClass": null,
    "TaxCategory": "EXEMPT",
    "RequireReceipt": true,
    "RequireShipment": true,
    "BaseUnit": "ITEM",
    "PurchaseUnit": "ITEM",
    "SalesUnit": "ITEM" }],
  "Query": "IN-PushItem",
  "CompanyId": "Company",
  "Id": "5ec9ca1b-7348-4c98-89c2-fb4743d54d9b",
  "TimeStamp": 637276282861245200,
  "AdditionalInfo": { "PXPerformanceInfoStartTime": "06/13/2020 06:58:05"}
}
```

# Push Notification fields

push notifications

**Inserted** – new rows in query. New updated values.

**Deleted** - rows that were present in a query but were removed. Old modified rows.
*Comparing deleted and inserted sets will show updated fields.*

**Query** - name of source definition (class name for Built-In definitions or Generic Inquiry name for GI definition).

**CompanyId** - login name of company.

**Id** - transaction identifier generated on DB level. Guarantees at least one delivery, so ‘Id’ and ‘Query’ fields can be used for deduplication.

**TimeStamp** - value that is guaranteed to increase with every transaction. Can be used to define order.

# Webhook handler implementation

```
public class TestWebhookHandler : IWebhookHandler
{
  public async Task<System.Web.Http.IHttpActionResult> ProcessRequestAsync(HttpRequestMessage request, CancellationToken cancellationToken)
  {
    using (var scope = GetAdminScope())
    {
      // Deserialize JSON into our Notification class
      var notification = JsonConvert.DeserializeObject<Notification>(await request.Content.ReadAsStringAsync());

      // We will use this Graph to insert our new item
      var graph = PXGraph.CreateInstance<NonStockItemMaint>();
      foreach (var item in notification.Inserted)
      {
         InventoryItem newItem = graph.Item.Search<InventoryItem.inventoryCD>(item.InventoryID.Trim());

         newItem = graph.Item.Insert(new InventoryItem()
         {
           InventoryCD = item.InventoryID,
           Descr = item.Description,
           ItemType = item.Type
         });

         // Simple way for us to track automatic inserts
         graph.Item.Cache.SetValueExt(newItem, "NoteText", "Created by Webhook");
      }
      graph.Actions.PressSave();
    }
    return new OkResult(request);
  }
}
```

# Recommendations for Webhooks

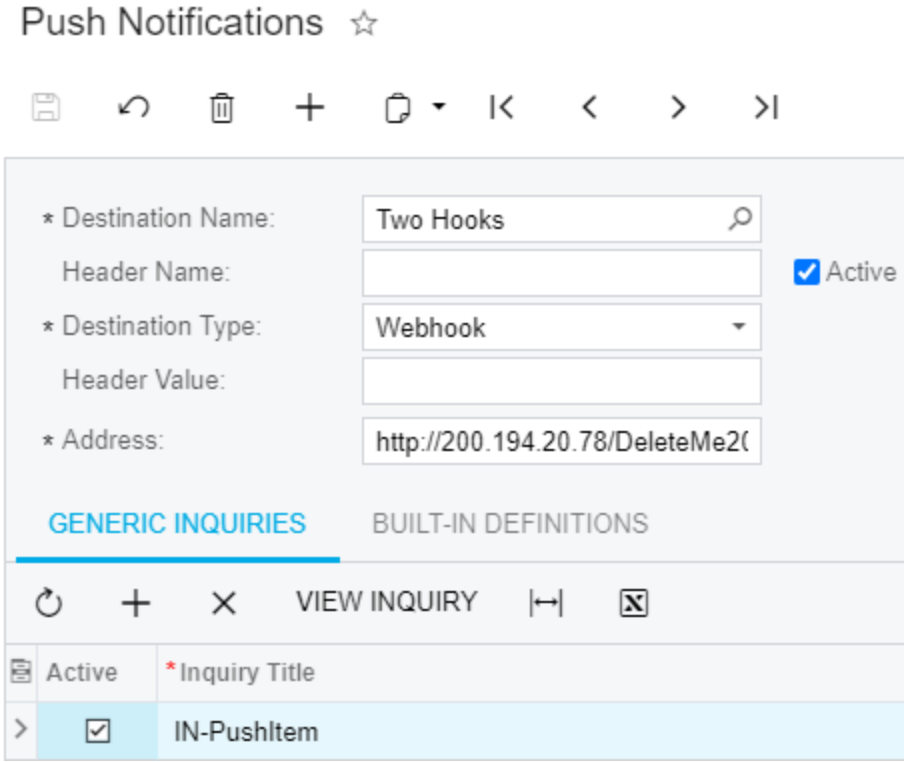

- Keep the handler as simple as possible
- Handle possible errors and reply accordingly
- It’s better to have some security, many webhook subscribers allow adding authentication headers like Basic or Bearer (tokens)
    - Add code in your handler to verify the header before processing the request

# Configuring push notifications in Acumatica

push notifications

1. Define GI for records we need to “monitor”
2. Select webhook as destination type
3. Set URL address where Acumatica will send the notification
4. Add GI we defined to the notification definition

(Screen SM302000)



# Recommendations for Push Notification Data Queries

push notifications

- Use as a simple data query if possible
- Do not use aggregation and grouping in the query
- Do not use joins of multiple detail tables (like Sales Order – Shipments – Shipment lines, i.e. several many-to-many relationships)
- Inner joins in queries may work a bit slower than left joins
- **For generic inquiries, do not use formulas** on the Results Grid tab of the Generic Inquiry (SM.20.80.00) form

**Tips by Vladimir Perov (thanks!)*

# Summary

- Webhooks are awesome
  - Simple and composable integrations
  - Widely supported around the web
  - Available in Acumatica 2020R1
- Push notifications are
  - Easy to configure by using Generic Inquiries
  - Can be used to send data to webhooks in any web app, including Acumatica
- We configured webhooks and push notifications in Acumatica
- Code available here: https://github.com/markoan/acumatica-webhook-example

# Q & A

Acumatica
The Cloud ERP

**Marco Villasenor**
marco@interastar.com


interastar™