



Webhooks – Stock Updates

Document Revision	3
What are Webhooks?	3
Ingram Micro Webhooks	3
How to Subscribe	4
Authentication	5
Ingram Micro Warehouse Identifiers	6
Webhook – Stock Update	8
<i>Field Descriptions</i>	<i>8</i>
<i>Sample Stock Update</i>	<i>10</i>
Recommended Tests	12
Authentication Code Samples	13
<i>C#</i>	<i>13</i>
<i>VB.Net</i>	<i>14</i>
<i>Java</i>	<i>15</i>
<i>Node.js</i>	<i>16</i>

Document Revision

Date	Version	Description
22/01/2021	1.0	Initial draft
27/01/2022	1.1	Added DK and FI information and ESD warehouse IDs Added new availabilityByWarehouse array

What are Webhooks?

A webhook is a means for an application to provide information to other applications in near real-time. A webhook delivers information to subscribed applications based on event triggers in the source application. Subscribed applications are not required to request the information from the source application, they simply wait to receive the information as it is published.

Webhooks are also known as web call-backs and HTTP push APIs.

Ingram Micro Webhooks

Ingram Micro offers webhooks as a method to notify your apps with real-time order status and stock information:

- Webhooks allow you to configure automatic notifications to your app when there is a change in order status or stock levels and pushes notifications as HTTP posts to your pre-defined destination URL
- Webhooks are simpler than API solutions — there is no need for your app to send traditional API requests to obtain information.

Why use Ingram Micro webhooks?

- Ingram Micro webhooks are easier and quicker to configure than APIs.
- Track events in the logs available for each webhook you create.
- Gain easy access to order status or stock level updates with the following options:
 - Change in available stock levels
 - Order placed on hold
 - Order invoiced
 - Order shipped
 - Order voided

You can support multiple notification types with a single webhook or configure one webhook per each notification type. Additional configuration options are available. The webhooks messages also provide HATEOAS (Hypermedia as the Engine of Application State) links to other APIs to easily request additional related information.

How to Subscribe

Webhook subscriptions are configured via My Apps in the customer API Portal at <https://developer.ingrammicro.com>

My Apps

The screenshot displays the 'My Apps' interface. At the top left, there is a blue button labeled '+Add Production App'. Below this, there are two application entries, each in a light gray box. The first entry is '20-123456-ProductionApp', which has a green arrow pointing to the word 'Webhooks', a red button labeled 'Production', and a green button labeled 'Approved'. The second entry is '20-123456-SandboxApp', which includes an information icon and the text 'IMPORTANT: Using Sandbox guidelines', a green arrow pointing to 'Webhooks', an orange button labeled 'Sandbox', and a green button labeled 'Approved'.

When you create your webhook, you'll be asked to provide your destination URL and to which Notification Events you wish to subscribe. You will also be provided with a Secret Key for each webhook, which will allow your web server or application to authenticate the messages received.

Authentication

When a webhook is created, a secret key value is provided in the customer API Portal. The secret key will allow the recipient of the webhook message to determine the message authenticity. The authentication steps are as follows:

- When a webhook message is created, the eventid value from the message is SHA512 encrypted using the secret key then base64 encoded.
- The encoded value is provided as the HTTP header x-hub-signature value.
- On receipt of the message the recipient should also SHA512 encrypt and base64 encode the eventid value in the message using the secret key.
- If the recipient's encrypted value matches the HTTP header x-hub-signature value, then the message is authentic.

Code samples in C#, VB.Net, Java and node.js are available in the [Authentication Code Samples](#) section at the end of this document.

Ingram Micro Warehouse Identifiers

Warehouse identifiers are used in several APIs such as MultiSKUPriceAndStockAPI and OrderCreateAPI. The table below provides a cross-reference for the codes provided in these messages. Please note that only certain product ranges can be ordered directly from the central warehouse. Your local ecommerce representative can advise on options for identifying these products via the FTP price file service. Only products listed as available in your local/central warehouse are available for you to order.

Customer Account Country	ERP	Local Warehouse IDs	Local Warehouse locations	Central Warehouse ID	Central Warehouse location	ESD Warehouse ID (Virtual Products)
AT	Impulse	28 44	Vienna, Austria Straubing, DE	N/A	N/A	51
BE	SAP	NL01 PE02	Tilburg, NL Tilburg, NL	PEDE	Straubing, DE	51
CH	Impulse	27 44	Altishofen, CH Straubing, DE	N/A	N/A	51
DE	Impulse	44	Straubing, DE	N/A	N/A	51
DK	Impulse	30 33	Rosersberg, SE Rosersberg, SE	81	Straubing, DE	51
ES	Impulse	29 78 43	Tarragona, ES Tarragona, ES Madrid, ES	81	Straubing, DE	51
FI	Impulse	31 33	Rosersberg, SE Rosersberg, SE	81	Straubing, DE	51
FR	Impulse	21 82	Lomme, FR Lomme, FR	81	Straubing, DE	51
HU	Impulse	40 44	Fót, HU Straubing, DE	N/A	N/A	51
IT	Impulse	23 86	Settala, IT Settala, IT	81	Straubing, DE	51
NL	SAP	NL01 PE02	Tilburg, NL Tilburg, NL	PEDE	Straubing, DE	51
PT	Impulse	26	Sintra, PT	81	Straubing, DE	51

SE	Impulse	33 84	Rosersberg, SE Rosersberg, SE	81	Straubing, DE	51
UK	Impulse	20 85	Crick, UK Crick, UK	81	Straubing, DE	51

Webhook – Stock Update

The stock update webhook will push a list of all products that have had a change in their available stock quantities. It is recommended that the initial stock quantities are fully refreshed daily in your system, prior to the start of business, via our FTP hourly availability files. The webhook updates can then be used for the remainder of the day to update products where available stock quantities have changed. Your local ecommerce representative can provide information on accessing the FTP hourly availability files.

Field Descriptions

Name	Parent	Type	Data Type	Min,Max Occurrence	Description	Notes
topic		Parameter	String	1,1	API topic	"resellers/catalog" for this webhook
event		Parameter	String	1,1	Webhook event type	"im::updated" for this webhook
eventTimeStamp		Parameter	String	1,1	Event timestamp (UTC)	
eventId		Parameter	String	1,1	Unique event ID	
resource		Array		1,1	Resource collection	
eventType	resource	Parameter	String	1,1	Resource event type	"IM::STOCK_UPDATE" for this webhook
ingrampartnumber	resource	Parameter	String	1,1	Ingram Micro part number	
ingramPartNumber	resource	Parameter	String	1,1	Vendor part number	
vendorName	resource	Parameter	String	1,1	Vendor name	
upcCode	resource	Parameter	String	1,1	UPC or EAN code	
skuStatus	resource	Parameter	String	1,1	Item status	"A" or null = active *, N, V or D = discontinued
backOrderFlag	resource	Parameter	String	1,1	Item backorder flag	"Y" or "N" – Indicates whether Ingram Micro will allow backorders
totalAvailability	resource	Parameter	Integer	1,1	Total available quantity in all warehouses servicing the customer account	
availabilityByWarehouse	resource	Array		1,1	Warehouse collection	

warehouseId	availabilityByWarehouse	Parameter	String	1,1	Warehouse identifier	
location	availabilityByWarehouse	Parameter	String	1,1	Warehouse description	
quantityAvailable	availabilityByWarehouse	Parameter	Integer	1,1	Available stock quantity	
quantityBackordered	availabilityByWarehouse	Parameter	Integer	1,1	Backordered stock quantity	
quantityBackorderedEta	availabilityByWarehouse	Parameter	Date	0,1	Backordered stock ETA date	
links	resource	Array		1,1	HATEOAS links collection	
topic	links	Parameter	String	1,1	API topic	"catalog" for this webhook
href	links	Parameter	String	1,1	HATEOAS link to topic of webhook message	Provides a link to the ProductSearchAPI for this item
type	links	Parameter	String	1,1	HTTP request method to use with link in href	

Sample Stock Update


```
{
  "topic": "resellers/catalog",
  "event": "im::updated",
  "eventTimeStamp": "2021-05-10T05:05:01.298+02:00",
  "eventId": "KVMS02V2Q9AHSWZ1UJ",
  "resource": [
    {
      "eventType": "IM::STOCK_UPDATE",
      "ingramPartNumber": "3F11053",
      "vendorPartNumber": "P018-003",
      "vendorName": "TRIPPLITE - CABLES AND CONNECTIVITY",
      "upcCode": "0037332168436",
      "skuStatus": "",
      "backOrderFlag": "Y",
      "totalAvailability": 1000,
      "availabilityByWarehouse": [
        {
          "warehouseId": "20",
          "location": "Crick, United Kingdom",
          "quantityAvailable": 1000,
          "quantityBackordered": 0
        }
      ],
      "links": [
        {
          "topic": "catalog",
          "href": "/resellers/v5/catalog?isoCountryCode=UK&customerNumber=20-123456&partNumber=3F11053",
          "type": "GET"
        }
      ]
    },
    {
      "eventType": "IM::STOCK_UPDATE",
      "ingramPartNumber": "CB07490",
      "vendorPartNumber": "P018-003",
      "vendorName": "TRIPPLITE - NETWORKING",
      "upcCode": "0037332168436",
      "skuStatus": "",
      "backOrderFlag": "Y",
      "totalAvailability": 500,
    }
  ]
}
```

```
"availabilityByWarehouse": [  
  {  
    "warehouseId": "20",  
    "location": "Crick, United Kingdom",  
    "quantityAvailable": 500,  
    "quantityBackordered": 750,  
    "quantityBackorderedEta": "2022-04-06"  
  }  
],  
"links": [  
  {  
    "topic": "catalog",  
    "href": "/resellers/v5/catalog?isoCountryCode=UK&customerNumber=20-123456&partNumber=CB07490",  
    "type": "GET"  
  }  
]  
]  
}
```

Recommended Tests

Test messages can be requested to your endpoint via the customer API Portal when you create a new webhook. Click on the “Send Test Event” button.

New Webhook

Secret Key [Why is this needed?](#) 

Destination URL (HTTPS only)

Please note that the x-hub-signature HTTP header is not provided when requesting test messages.

Authentication Code Samples

The code samples below demonstrate functions that can be used to authenticate webhook messages received from Ingram Micro.

Disclaimer: The sample code is for demonstration purposes only and is provided "as is". Any express or implied warranties, including the implied warranties of merchantability and fitness for a particular purpose are disclaimed.

The argument values passed to these functions are:

- secretKey – The secret key value for your webhook as provided in the customer API portal.
- eventID – The eventId JSON parameter value from the webhook message.
- xHubSignature – The HTTP header x-hub-signature value from the webhook message.

If the function returns true, then the message is authentic.

C#

Code

```
public bool isXhubSignatureValid(string secretKey, string eventID, string xHubSignature)
{
    byte[] secretkeyBytes = System.Text.UTF8Encoding.UTF8.GetBytes(secretKey);
    byte[] eventIDBytes = System.Text.UTF8Encoding.UTF8.GetBytes(eventID);
    bool isValid;

    using (System.Security.Cryptography.HMACSHA512 hmac = new System.Security.Cryptography.HMACSHA512(secretkeyBytes))
    {
        byte[] resultBytes = hmac.ComputeHash(eventIDBytes);
        string resultB64 = System.Convert.ToBase64String(resultBytes);
        if (resultB64 == xHubSignature)
            isValid = true;
    }

    return isValid;
}
```

VB.Net

Code

```
Function isXhubSignatureValid(ByVal secretKey As String, ByVal eventID As String, ByVal xHubSignature As String) As Boolean

    Dim secretkeyBytes As Byte() = System.Text.UTF8Encoding.UTF8.GetBytes(secretKey)
    Dim eventIDBytes As Byte() = System.Text.UTF8Encoding.UTF8.GetBytes(eventID)
    Dim isValid As Boolean

    Using hmac As System.Security.Cryptography.HMACSHA512 = New System.Security.Cryptography.HMACSHA512(secretkeyBytes)
        Dim resultBytes As Byte() = hmac.ComputeHash(eventIDBytes)
        Dim resultB64 As String = System.Convert.ToBase64String(resultBytes)
        If resultB64 = xHubSignature Then
            isValid = True
        End If
    End Using

    Return isValid

End Function
```

Java

Dependencies

```
java.io.UnsupportedEncodingException  
java.security.InvalidKeyException  
java.security.NoSuchAlgorithmException  
java.util.Base64  
javax.crypto.Mac  
javax.crypto.spec.SecretKeySpec
```

Code

```
public static boolean isXhubSignatureValid(String secretKey, String eventID, String xHubSignature){  
  
    boolean isValid = false;  
  
    try {  
        byte[] secretkeyBytes = secretKey.getBytes("UTF-8");  
        byte[] eventIDBytes = eventID.getBytes("UTF-8");  
  
        Mac hmac = Mac.getInstance("HmacSHA512");  
        SecretKeySpec secretSpec = new SecretKeySpec(secretkeyBytes, "HmacSHA512");  
        hmac.init(secretSpec);  
  
        byte[] resultBytes = hmac.doFinal(eventIDBytes);  
        String resultB64 = Base64.getEncoder().encodeToString(resultBytes);  
  
        if (resultB64.equals(xHubSignature)){  
            isValid = true;  
        }  
  
    } catch (UnsupportedEncodingException e) {  
        e.printStackTrace();  
    } catch (NoSuchAlgorithmException e) {  
        e.printStackTrace();  
    } catch (InvalidKeyException e) {  
        e.printStackTrace();  
    }  
  
    return isValid;  
}
```

Node.js

Code

```
function isXhubSignatureValid(secretKey, eventID, xHubSignature){  
  
    var crypto = require("crypto");  
    var isValid = false;  
  
    var hmac = crypto.createHmac("sha512", secretKey);  
    hmac.update(eventID);  
    var resultB64 = hmac.digest("base64");  
  
    if (resultB64 == xHubSignature){  
        isValid = true;  
    }  
  
    return isValid;  
}
```