



## Webhooks – Order Updates

<b>Document Revision</b>	<b>3</b>
<b>What are Webhooks?</b>	<b>3</b>
<b>Ingram Micro Webhooks</b>	<b>3</b>
<b>How to Subscribe</b>	<b>4</b>
<b>Authentication</b>	<b>5</b>
<b>Ingram Micro Warehouse Identifiers</b>	<b>6</b>
<b>Webhook – Order Status Update</b>	<b>8</b>
<i>Field Descriptions</i>	8
<i>Order Status Notification Events</i>	9
<i>Samples</i>	10
Hold Order Sample	10
Order Shipped Sample	11
Order Void Sample	13
Invoiced Order Sample	14
<b>Recommended Tests</b>	<b>17</b>
<b>Authentication Code Samples</b>	<b>18</b>
<i>C#</i>	18
<i>VB.Net</i>	19
<i>Java</i>	20
<i>Node.js</i>	21

## Document Revision

Date	Version	Description
13/01/2020	1.0	Initial draft
27/01/2022	1.1	Added DK and FI information and ESD warehouse IDs

## 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 shows a 'My Apps' interface with the following elements:

- A blue button at the top left: '+Add Production App'
- Two application cards:
  - Card 1: '20-123456-ProductionApp' with a green arrow pointing to 'Webhooks', a red 'Production' button, and a green 'Approved' button.
  - Card 2: '20-123456-SandboxApp' with an information icon and text 'IMPORTANT: Using Sandbox guidelines', a green arrow pointing to 'Webhooks', an orange 'Sandbox' button, and a green 'Approved' button.

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 – Order Status Update

### Field Descriptions

Name	Parent	Type	Data Type	Min,Max Occurrence	Description	Notes
topic		Parameter	String	1,1	API topic of the message	"resellers/orders" for this webhook
event		Parameter	String	1,1	Webhook event type field	"im::updated" for this webhook
eventTimeStamp		Parameter	String	1,1	Event timestamp in UTC	
eventId		Parameter	String	1,1	Unique event ID	
resource		Object		1,1	Resource container	
eventType	resource	Parameter	String	1,1	Resource event type	<a href="#">Order Status Events</a>
orderNumber	resource	Parameter	String	1,1	Ingram Micro order number	
customerPoNumber	resource	Parameter	String	1,1	Customer purchase order reference.	
orderEntryTimeStamp	resource	Parameter	String	1,1	Order timestamp	
lines	resource	Array		1,1	Order line collection	
lineNumber	lines	Parameter	Integer	1,1	Ingram Micro's ERP line number.	
subOrderNumber	lines	Parameter	String	1,1	Ingram Micro sub-order number.	This is the Ingram Micro sales order number at shipment level
lineStatus	lines	Parameter	String	1,1	Line status code	
ingramPartNumber	lines	Parameter	String	1,1	Ingram Micro part number. Also referred to as SKU (stock keeping unit) or material number.	
vendorPartNumber	lines	Parameter	String	1,1	Vendor part number	
requestedQuantity	lines	Parameter	Integer	1,1	Requested order quantity.	
shippedQuantity	lines	Parameter	Integer	1,1	Shipped quantity	
backOrderedQuantity	lines	Parameter	Integer	1,1	Backorder quantity.	
shipmentDetails	lines	Object		1,1	Shipment Details container	



shipmentDate	shipmentDetails	Parameter	String	1,1	Ship date.	Null if order not invoiced or voided
shipFromWarehouseId	shipmentDetails	Parameter	String	1,1	Ship from warehouse ID	(see <a href="#">Ingram Micro Warehouse Identifiers</a> )
warehouseName	shipmentDetails	Parameter	String	1,1	Ship from warehouse name	
carrierCode	shipmentDetails	Parameter	String	1,1	Ingram Micro carrier code	
carrierName	shipmentDetails	Parameter	String	1,1	Ingram Micro carrier description	
proNumber	shipmentDetails	Parameter	String	0,1	Carrier pro number	
packageDetails	shipmentDetails	Array		0,1	Package details collection for shipped orders.	
cartonNumber	packageDetails	Parameter	String	1,1	Sequential package number.	
quantityInbox	packageDetails	Parameter	String	1,1	Number of items in this package.	
trackingNumber	packageDetails	Parameter	String	1,1	Package ID.	
serialNumberDetails	resource	Array		0,1	Serial numbers collection	
serialNumber	serialNumberDetails	Parameter	String	1,1	Product serial number.	
links	resource	Array		1,1	HATEOAS links collection	
topic	links	Parameter	String	1,1	API topic	
href	links	Parameter	String	1,1	HATEOAS link to topic of webhook message	
type	links	Parameter	String	1,1	HTTP request method to use with link in the href parameter	

## Order Status Notification Events

Notification Event	Event Code	Sub-event code	Description
Shipped	im::updated	im::order_shipped	The order has shipped
Invoiced	im::updated	im::order_invoiced	The order has invoiced
On Hold	im::updated	im::order_hold	The order is on hold
Voided	im::updated	im::order_voided	The order was cancelled or voided

## Samples

## Hold Order Sample

```
{
  "topic": "resellers/orders",
  "event": "im:updated",
  "eventTimeStamp": "2020-11-27T03:15:15.267-08:00",
  "eventId": "HUP1KMOA5KT2WWTWAR",
  "resource": {
    "eventType": "IM:order_hold",
    "orderNumber": "20-EXORD",
    "customerPoNumber": "123456789",
    "orderEntryTimeStamp": "2020-11-27 03:13:52",
    "lines": [
      {
        "lineNumber": "001",
        "subOrderNumber": "20-EXORD-11",
        "lineStatus": "IM:SALES_HOLD",
        "ingramPartNumber": "091EH49",
        "vendorPartNumber": "DELL-U2520D",
        "requestedQuantity": "1",
        "shippedQuantity": "1",
        "backOrderedQuantity": "0",
        "shipmentDetails": {
          "shipmentDate": null,
          "shipFromWarehouseId": "20",
          "warehouseName": "INGRAM MICRO (UK) LTD.",
          "carrierCode": "FN",
          "carrierName": "NEXT DAY PARCELFORCE"
        }
      }
    ]
  },
  "links": [
    {
      "topic": "orders",
      "href": "/resellers/v5/orders/20-EXORD",
      "type": "GET"
    }
  ]
}
```

## Order Shipped Sample

```
{
  "topic": "resellers/orders",
  "event": "im:updated",
  "eventTimeStamp": "2021-02-15T23:30:40.581Z",
  "eventId": "WF8LN3MCENM0K3U2F7",
  "resource": {
    "eventType": "IM:order_shipped",
    "orderNumber": "20-WNFR9",
    "customerPoNumber": "123456",
    "orderEntryTimeStamp": "2021-02-15T11:11:55Z",
    "lines": [
      {
        "lineNumber": "001",
        "subOrderNumber": "20-WNFR9-11",
        "lineStatus": "IM:SHIPPED",
        "ingramPartNumber": "2M34EV1",
        "vendorPartNumber": "1KR65B#BHC",
        "requestedQuantity": "2",
        "shippedQuantity": "2",
        "backOrderedQuantity": "0",
        "shipmentDetails": {
          "shipmentDate": "2021-02-15",
          "shipFromWarehouseId": "20",
          "warehouseName": "INGRAM MICRO (UK) LTD.",
          "carrierCode": "NX",
          "carrierName": "CONSOLIDATE",
          "packageDetails": [
            {
              "cartonNumber": "1",
              "quantityInbox": "2",
              "trackingNumber": "EXAMPLE12345"
            }
          ]
        }
      },
      {
        "serialNumberDetails": [
          {
            "serialNumber": "TH08E3Q1B3"
          },
          {
            "serialNumber": "TH08E3Q1B4"
          }
        ]
      }
    ]
  },
  "links": [
    {
```

```
"topic": "orders",  
  "href": "/resellers/v5/orders/20-WNFR9?isoCountryCode=UK&customerNumber=20-123456",  
  "type": "GET"  
}  
]  
}
```

## Order Void Sample

```
{
  "topic": "resellers/orders",
  "event": "im:updated",
  "eventTimeStamp": "2021-02-12T12:15:01.677Z",
  "eventId": "PXL5H2ZMNG5AHTB45",
  "resource": {
    "eventType": "IM:order voided",
    "orderNumber": "20-WN91P",
    "customerPoNumber": "TETSTT",
    "orderEntryTimeStamp": "2021-02-12T12:02:30Z",
    "lines": [
      {
        "lineNumber": "001",
        "subOrderNumber": "20-WN91P-11",
        "lineStatus": "IM:VOIDED",
        "ingramPartNumber": "S60DMP9",
        "vendorPartNumber": "10SQ0067UK",
        "requestedQuantity": "20",
        "shippedQuantity": "20",
        "backOrderedQuantity": "0",
        "shipmentDetails": {
          "shipmentDate": "2021-02-12",
          "shipFromWarehouseId": "20",
          "warehouseName": "INGRAM MICRO (UK) LTD.",
          "carrierCode": "NX",
          "carrierName": "CONSOLIDATE"
        }
      }
    ]
  },
  "links": [
    {
      "topic": "orders",
      "href": "/resellers/v5/orders/20-WN91P?isoCountryCode=UK&customerNumber=20-123456",
      "type": "GET"
    }
  ]
}
```

## Invoiced Order Sample

```
{
  "topic": "resellers/orders",
  "event": "im:updated",
  "eventTimeStamp": "2021-02-16T02:33:12.024Z",
  "eventId": "OWJJ0XL4IBYWWN226B",
  "resource": {
    "eventType": "IM:order invoiced",
    "orderNumber": "20-WMV7F",
    "customerPoNumber": "1464651",
    "orderEntryTimeStamp": "2021-02-08T10:27:46Z",
    "lines": [
      {
        "lineNumber": "001",
        "subOrderNumber": "20-WMV7F-12",
        "lineStatus": "IM:INVOICED",
        "ingramPartNumber": "1413609",
        "vendorPartNumber": "WD20SPZX",
        "requestedQuantity": "14",
        "shippedQuantity": "14",
        "backOrderedQuantity": "0",
        "shipmentDetails": {
          "shipmentDate": "2021-02-15",
          "shipFromWarehouseId": "85",
          "warehouseName": "INGRAM MICRO (UK) LTD.",
          "carrierCode": "NX",
          "carrierName": "CONSOLIDATE",
          "packageDetails": [
            {
              "cartonNumber": "1",
              "quantityInbox": "14",
              "trackingNumber": "Z722314224456931"
            }
          ]
        }
      }
    ]
  }
},
{
  "lineNumber": "009",
  "subOrderNumber": "20-WMV7F-12",
  "lineStatus": "IM:INVOICED",
  "ingramPartNumber": "1411520",
  "vendorPartNumber": "WD5003AZEX",
  "requestedQuantity": "8",
  "shippedQuantity": "8",
  "backOrderedQuantity": "0",
  "shipmentDetails": {
    "shipmentDate": "2021-02-15",
```

```
"shipFromWarehouseId": "85",
"warehouseName": "INGRAM MICRO (UK) LTD.",
"carrierCode": "NX",
"carrierName": "CONSOLIDATE",
"packageDetails": [
  {
    "cartonNumber": "1",
    "quantityInbox": "8",
    "trackingNumber": "EXAMPLETRACKINGNUMBER12345"
  }
],
"serialNumberDetails": [
  {
    "serialNumber": "WCC6Y0TVDJHY"
  },
  {
    "serialNumber": "WCC6Y1HLYK7T"
  },
  {
    "serialNumber": "WCC6Y1HP9DAE"
  },
  {
    "serialNumber": "WCC6Y1XDN2TJ"
  },
  {
    "serialNumber": "WCC6Y1XDNETN"
  },
  {
    "serialNumber": "WCC6Y1XDNYRY"
  },
  {
    "serialNumber": "WCC6Y3TCYHXJ"
  },
  {
    "serialNumber": "WCC6Y5DT9PTA"
  }
]
},
"links": [
  {
    "topic": "orders",
    "href": "/resellers/v5/orders/20-WMV7F?isoCountryCode=UK&customerNumber=20-123456",
    "type": "GET"
  },
  {
```


```
"topic": "invoices",  
  "href": "/resellers/v5/invoices/20-WMV7F-12?isoCountryCode=UK&customerNumber=20-123456",  
  "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?](#) 

Generate New Key

Destination URL (HTTPS only)

Send Test Event

*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;  
}
```