Webhooks

Introduction

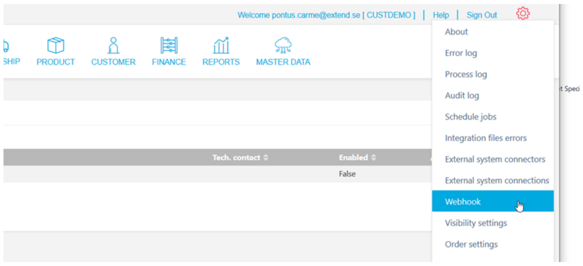
You can create webhooks in Extend.

The webhook will make a call based on the events (triggers) that you define on the webhook.

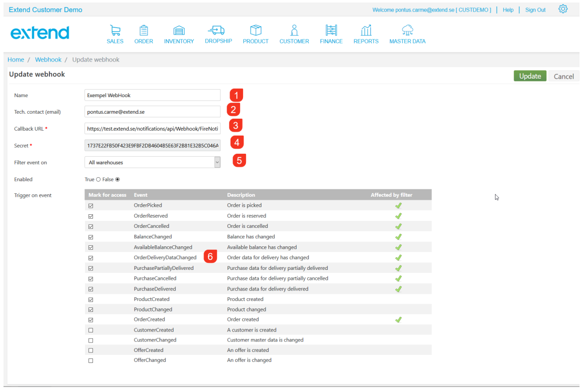
På denna sida:

Add / edit Webhooks

You find the webhook meny under Settings.



When you create or edit a webhook, this page will occure.



Update webhook

Name	Exempel WebHook
Tech. contact (email)	pontus.carme@extend.se
Callback URL *	https://test.extend.se/notifications/api/Webhook/FireNoti
Secret *	1737E22FB50F423E9FBF2DB4604B5E63F2B81E32B5C046A
Filter event on	All warehouses
Enabled	True ○ False ●

Trigger on event

Mark for access	Event	Description	Affected by filter
☑	OrderPicked	Order is picked	✓
☑	OrderReserved	Order is reserved	✓
☑	OrderCancelled	Order is cancelled	✓
☑	BalanceChanged	Balance has changed	✓
☑	AvailableBalanceChanged	Available balance has changed	✓
☑	OrderDeliveryDataChanged	Order data for delivery has changed	✓
☑	PurchasePartiallyDelivered	Purchase data for delivery partially delivered	✓
☑	PurchaseCancelled	Purchase data for delivery partially cancelled	✓
☑	PurchaseDelivered	Purchase data for delivery delivered	✓
☑	ProductCreated	Product created	
☑	ProductChanged	Product changed	
☑	OrderCreated	Order created	✓
□	CustomerCreated	A customer is created	
□	CustomerChanged	Customer master data is changed	
□	OfferCreated	An offer is created	
□	OfferChanged	An offer is changed	

1. The name is a public name that you add to the hook, used in the list of hooks, so you can identify the hook.
2. A email to a technical contact for the webhook.
3. Callback URL
   Here you add your URL to where Extend will send the webhook
4. Secret, here you add a secret. Extend will sign the body in the hook using SHA256 with your secret. The signed value is added in *request header 'ms-signature'*. The body should be verified before processing to avoid spam calls
5. Filter event on,
   Here you can limit the hook on different warehouses. If you for example need to have a hook on availble balance, and your installation in Extend handles several warehosues. you can limit the hook to only handle balances for a specific warehouse.
6. Trigger on events.
   Here you mark on which events you want the Hook to fire on.

Example

The content in *Notifications* array will be different depending on the *Action*

PurchasePartiallyDelivered:

```
{
"Id": "6e249eebb818463086ae0dda6eeba7b2",
"Attempt": 1,
"Properties":
{ "NotificationId": "3b3fd4a6-36d5-46fe-8a3f-e73532529db3" }

,
"Notifications": [
{
"Action": "PurchasePartiallyDelivered",
"Details":
{ "OrderNumber": "RP-101", "Warehouse": "GreenWarehouse" }

}
]
}
```

ProductCreated:

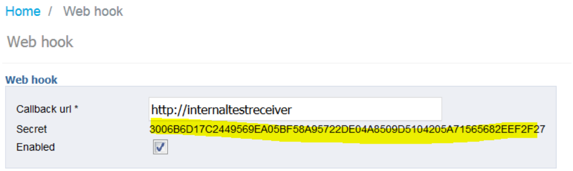
```
{

"Id": "13f078bcbe2243dc997655ed2cdbbdf7",

"Attempt": 1,

"Properties": {

"NotificationId": "90c6340b-1f5c-49e0-9cee-45df5dd145c2"

},

"Notifications": [

{

"Action": "ProductCreated",

"Details": {

"ProductNumber": "Financial Invoice"

}

}

]

}
```

Signing

To make sure the webhook is sent by Extend it is signed. It is optional to use the signature for verification.

Body is signed using SHA256 with clientsecret. The signed value is added in header 'ms-signature'

ClientSecret:



Home / Web hook

Web hook

Web hook

Callback url * http://internaltestreceiver

Secret 3006B6D17C2449569EA05BF58A95722DE04A8509D5104205A71565682EEF2F27

Enabled ☑

Body: the full request body

SignatureFromHeader: request header 'ms-signature'

Sample pseudo code:

```
byte[] secret = Encoding.UTF8.GetBytes(ClientSecret);
using (var hasher = new HMACSHA256(secret))
{
byte[] data = Body.ToByteArray();
actualHash = hasher.ComputeHash(data);
}
byte[] expectedHash = FromHex(SignatureFromHeader).ToByteArray()
if(IsSecretEqual(expectedHash, actualHash))
{
//process web hook
}
```

Sample

ClientSecret = secret123

Body = {'test':'test2'}

BodyAsHex = 7B2774657374273A277465737432277D

ms-signature = sha256=25FB6994568A75CD233E04BA1C653AF1BF476041CC543AF04F82CAAC482C201A