Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com

1. Introduction

Bosbec is an automation platform based on low code that makes it possible to easily and safely build your automation and integration solutions. You create these solutions using a workflow builder.

The platform supports various channels to receive and send data, such as http, email, text messages and more, enabling seamless communication with other services and systems. Once you've built your solution, you can immediately deploy it by pressing "live", making the solution ready for use instantly.

Unlike similar platforms where you might need to write your own scripts, Bosbec provides predefined jobs for various tasks. This approach ensures your solutions are constantly tested and verified for optimal performance. Additionally, Bosbec offers a dynamic and flexible environment, allowing you to adjust and modify your workflows effortlessly to meet changing needs and requirements.

2. Conditions for order integration

This is an example of an order integration. In this integration, there are certain conditions if you use it exactly as it is built. You can easily change these conditions if you need to in your workflow.

- Order status is set as Fulfilled.
- We create customer from Shopify, but we don't update the customer if it already exists in Fortnox.
- Billing address in Shopify is set as Customer's address in Fortnox and shipping address is set as delivery address in Fortnox.
- The customer number from Shopify becomes the customer number in Fortnox.
- Order information from Shopify is entered into Fortnox where each product gets its own row and is entered by name without article number with quantity and price.
- Freight is entered into Fortnox from Shopify .
- VAT is fixed at 25%. In this example, it is a web shop that sells to consumers with a price incl. VAT.
- To book the order in Fortnox, the following is done.
    - The order document number from Shopify is used to indicate that the product has been delivered.
    - Invoice is created as "cash invoice"
    - The invoice is bookkeeped

3. Order transfer Webshop to Business system

Every new order that comes in from a webshop goes directly to Fortnox .

This document describes step by step how an integration between Shopify and Fortnox is built with Bosbec's platform. Since this is an example, it should be pointed out that conceptually it could just as easily be another web shop as well as another business system.

The following steps are followed:

1. The integration against Fortnox
2. The integration with Shopify
3. The mapping and management of data between Shopify and Fortnox in Bosbec's solution

3.1. Shopify2Fortnox solution

As a partner to Bosbec, you can create an integration that you can publish in Fortnox or create an integration directly for a customer. If you want to create your own app in Fortnox , you can find more information here https://support.fortnox.se/komigangguide-tjanster/kom-igang-med-utvecklarportalen .

3.2. Authorization to Shopify

To access the Shopify API and generate API keys from your Shopify admin panel, follow these steps:

1. Sign in to your Shopify account.
2. Navigate to ” Apps ”.
3. Go to " App and Sales Channel Listing ".
4. Navigate to "Develop Apps ".
5. Click the "Create an app " button.
6. Give the app a suitable name and click "Create app ".
7. After creating the app, go to the "Configure" tab.
8. Click the "Configure" button for Admin API integration, where you can set the permissions (scopes) for your app , such as read and write access to various Shopify APIs . Select all required permissions for your app and click "Save".
9. In the tab "API references" you get the API key and API secret key. The API key is used for authentication.
10. Click the "Install App " button to install the custom app you have set up.
11. After you install the app, you will receive the " Admin API Access Token", which is your access token to request data from the Admin API.

The next step is to create a webhook in Shopify. You can read at the following link how to do https://shopify.dev/docs/apps/webhooks.

To create endpoints in Bosbec see section 2.4.

3.3. Authorization to Fortnox

The authorization process Fortnox uses is OAuth2. It applies in two stages.

1. First, you get an authorization code where the user approves the application and access to the account.
2. Access token and refresh token are then generated.

An overall description can be found here https://www.fortnox.se/developer/authorization

To begin with, you create a url that the user must click on. To create this, the following parameters are needed;

- Authorization URL that is **https://apps.fortnox.se/oauth-v1/auth**
- The Client ID is the public identifier for the app
- Redirect-uri – Channel created in Bosbec
- Scope (list of scopes can be found at this link https://www.fortnox.se/developer/guides-and-good-to-know/scopes)
- The State parameter is used to store request-specific data and/or prevent CSRF attacks. You can choose a number yourself.
- Access_type - Specifies whether the app can update access tokens when the user is not present in the browser.
- Response_type (required) - Response_type should be code , indicating that the program expects to receive an authorization code if successful.
- Account_type (optional) - Specifies whether to create a service account. The service account must be enabled if you use the Developer Portal. A service account is not associated with any specific user and has a specific set of permissions appropriate for integrations within the requested scopes. There can only be one service account per client_id and customer. Only system administrators at the customer can authorize service accounts during the authorization process. The only valid value is "service", if a service account is to be created.

Fortnox API OAuth 2.0 authorization request URL will look like this:

https://apps.fortnox.se/oauth-v1/auth?client_id={Client-ID}&redirect_uri={kanal från Bosbec}&scope={companyinformation order customer article bookkeeping invoice payment}&state=12345&access_type=offline&response_type=code&account_type=service

For this integration that manage orders, we have used the following scope (in url):

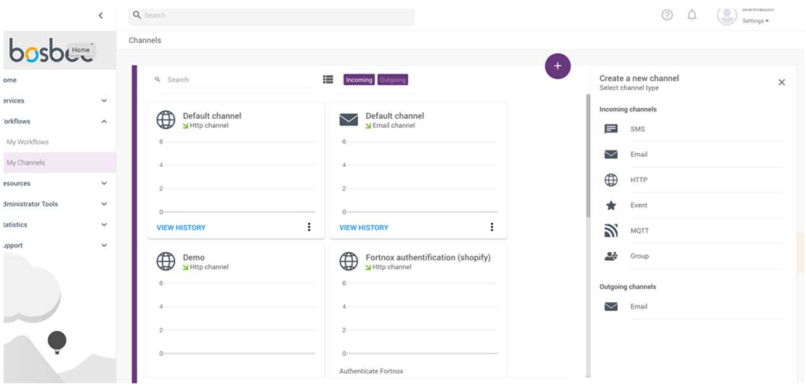
- company information
- order
- customer
- article
- bookkeeping
- invoice
- payment

If you want to integrate another system into the same flow, such as e.g. POS systems, inventory management, etc. you may need more scopes .

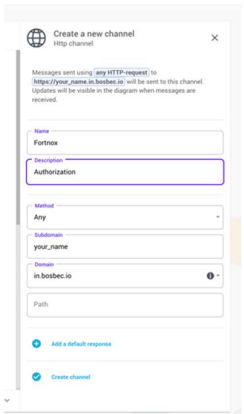
2.4. Channel in Bosbec

We start by creating a channel in Bosbec.

1. In the platform you go to "My Channels" and create an http channel



2. Write in the fields and save the channel.

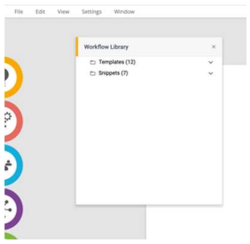


2.5. Build authentication for Fortnox in Bosbec

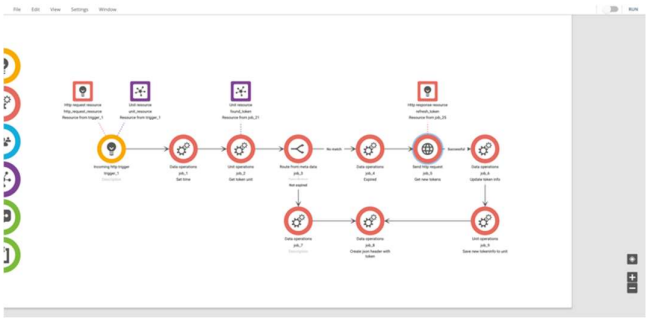
To begin with, you open a workflow in Bosbec. You do this under " Workflows" and then "My Workflows ". You choose " Create new workflow" and then "Empty workflow".

Now you can start building your solution and we start with the authentication process.

For Fortnox authentication, there is a template so you can quickly create a workflow. You then select "Edit" and then "Workflow library " and then your Fortnox template.



When you then import the workflow, you get it on your workspace.

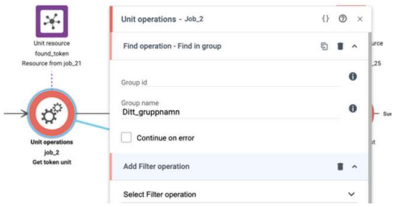


Now you can start configuring your workflow. The advantage of having it in the workflow is that if Fortnox changes something in its process, it is easy to adjust it in the workflow.

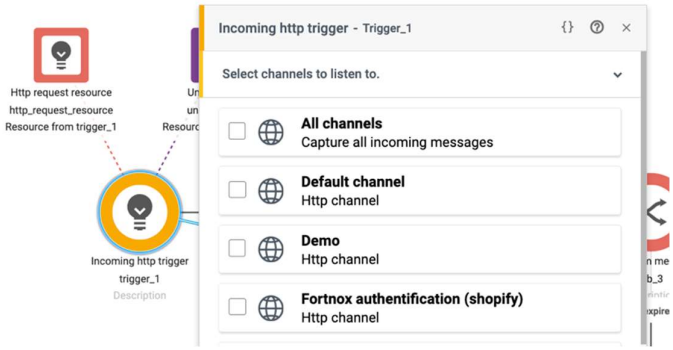
Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com

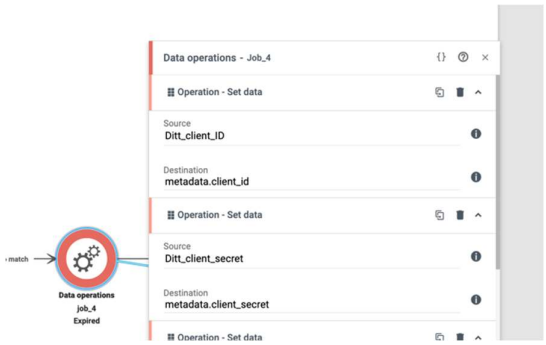
1. Start by creating a group in Bosbec where you save your Token. This is to be able to manage the refresh token in the process. You do this under "Resources" and "Groups".



2. Start by selecting the channel you created. If you choose to check "Is Public", you do not need a token. It can be good when you test. When you then want it to be live, you create a token and do so under "Administrator Tools" and "API Tokens".



3. Now you can open job_4 and fill in the following information under "Source".
   - Client ID
   - Client Secret



4. In order for it to be live, you " toggle " it in the upper right corner. Don't forget to continuously save your workflow.

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

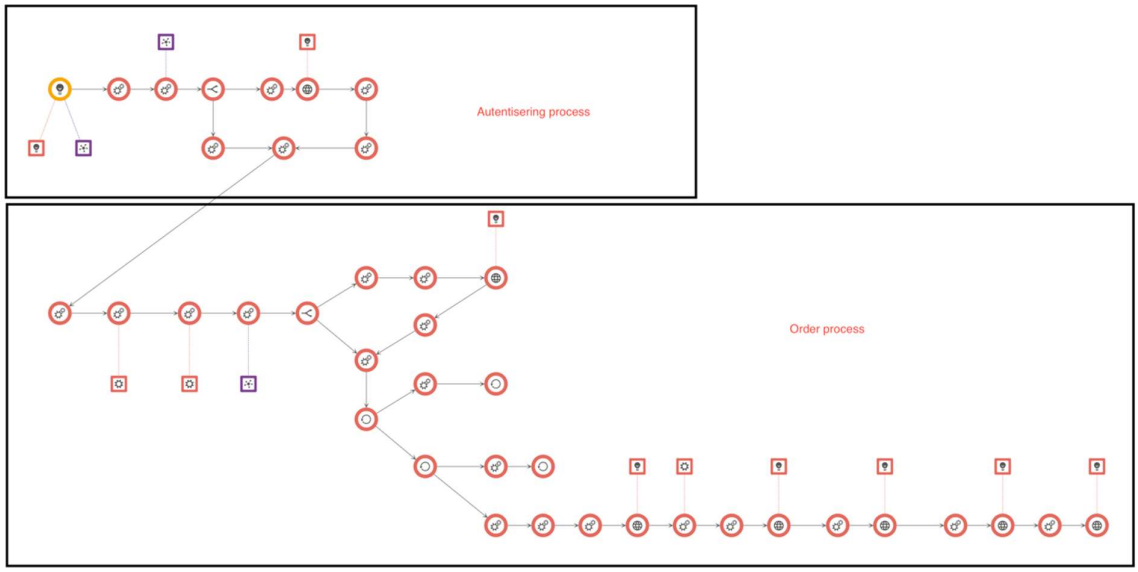
Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com

Now you are done with the authentication process and can start building your order flow in Bosbec.

2.6. Building the order flow in Bosbec

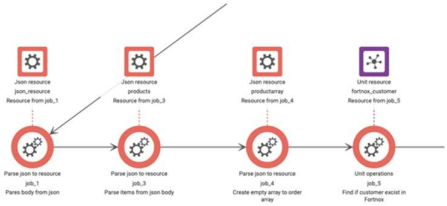
In the same way as the authentication, there is a template for this flow. In this instruction, we will describe in detail how the workflow was built.

To get an overview, the workflow (authentication and order flow) will look like below when it is ready.



1. Below is a step-by-step description of how to create your workflow. It's about getting order data from Shopify and then mapping this information to Fortnox. In order to be able to map the data more easily, we will create json resources that we can then use when we make a "Post" to Fortnox .

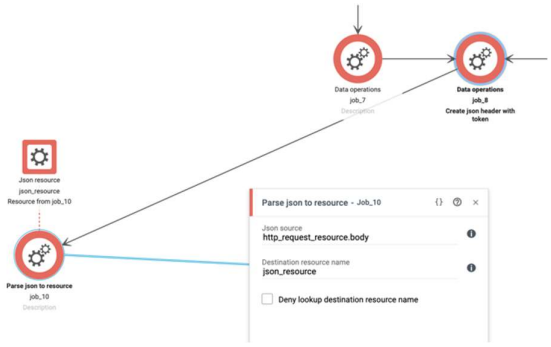
2. The first part is to get data from Fortnox to Bosbec which we can work with. We need to have order data, customer information and customer data for this process.



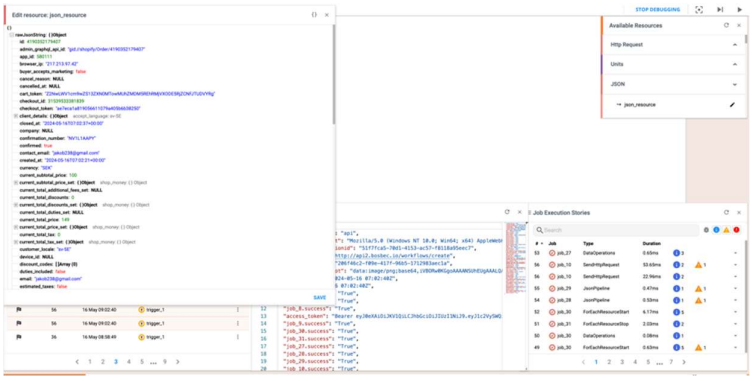
Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com

- We start by taking out a job called " Parse json to resource ". We create a resource from the http call. Then drag an arrow from job in the authentication process "job_8" to get the call connected.



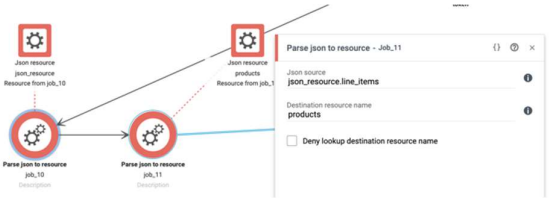
- Now we have created a json resource and get data to Bosbec. To see what data you get in, you can "debug" and then look at your json resource. Create an order in the Shopify store then go to your workflow in Bosbec. Then select " View " and then "Execution History ". There you then get your runs. Click on a run to enter debug mode. In debug mode, you can then under "Available Resources" choose which of your resources you want to look at. In this case, we have created a json resource and to see the data there, select "JSON" and then "json_resource " and click on the pen. This function helps you continuously monitor your workflow (you can read more about debugging here https://help.bosbec.com/knowledge-base/troubleshooting-your-workflow/).



- In the next job, we want to create a resource for the products. We again choose a ”Parse json to resource” job and in json the item line_items that we want as a separate resource. We enter json_ resource.line _items under ”Json resource” and name the resource to ”products”. Then drag an arrow from "job_10" to the new job.

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com



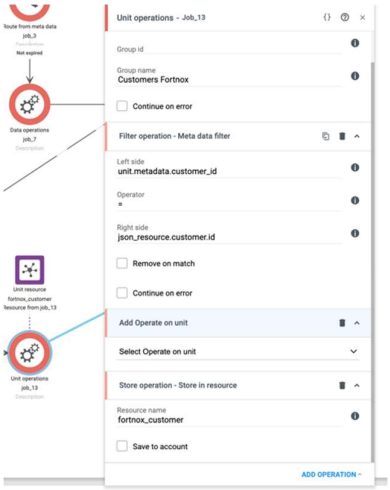
- Since line_items contains array, for the next job we will create an empty array where we can then insert product details as a resource. In "Json source" we insert an empty array ”[ ]” and name it ”productarray”.

- In the next job I want to see if the customer is already in Fortnox. We create a group that we name " Customers Fortnox”. We save the customer ID in this group and compare whether it is in this group in future orders .

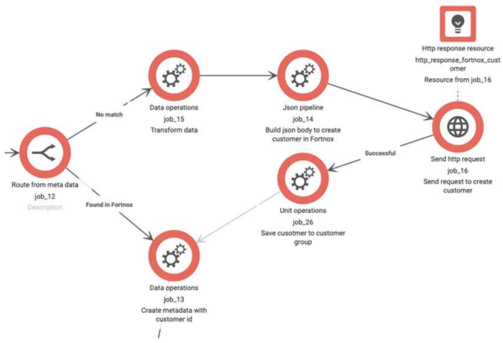
  We select a "Unit operations" job and then we select "Find in group" under "Group name " we write "Customers Fortnox”. At ”Add filter operation" then you select "Meta data filter" and fill in the conditions. At ”Left side" you write "unit.metadata.customer_id" and as "Operator" you write "=" and on "Right side" you write "json_resource.customer.id". At "Store operation" you select "Store in resource" and write in "Resource name” fortnox_customer . We choose not to save this in the account, but only in the run.

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

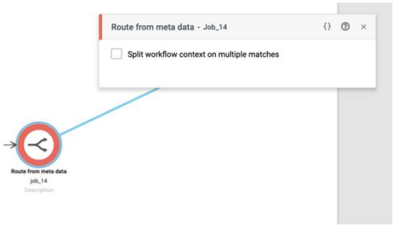
Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com



3. Now we will start creating the data structure so that it maps to Fortnox's structure. We go through this part of the workflow that handles customer data.



- We choose a job called "Route from metadata". This is used to see if the customer exists in Fortnox or if a new customer is to be created in Fortnox . This should then be connected with two additional jobs, one to create a customer in Fortnox and a job to create metadata for customer_id .



Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com

We start by creating the url to enter the customer in Fortnox and for that we select job "Data operation" and then "Concat". In Shopify, you get the customer's name as first_name and last_name . In Fortnox, we want to merge this into one name. Under "Source with format" we write "{json_ resource.customer .first_name}{json_resource.customer.last_name}" and then at destination we change it to metadata and write "metadata.name".

We continue to add to a new operation and again choose "Concat". Now we want to make the call to Fortnox's customer register metadata. We then enter the url to the call and json the data for customer ID, "https://api.fortnox.se/3/customers/{json_resource.customer.id}", and turn it into metadata, and write "metadata.customer _url". When we then drag the arrow to this job, we must configure the connection and then select "Route destination on no match".

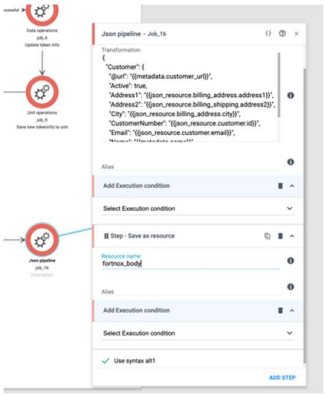
- Now the data structure for customer data should be created between Shopify and Fortnox. We then select job " Json pipeline" and step "transform". On the left side we write the structure Fortnox uses and on the right the structure from Shopify. It will look like below.

```
{
"Customer": {
"@ url ": "{{metadata.customer _url}}",
"Active": true,
"Address1": "{{json_ resource.billing _address.address1}}",
"Address2": "{{json_ resource.billing _shipping.address2}}",
"City": "{{json_ resource.billing _address.city}}",
" CustomerNumber ": "{{json_resource.customer.id}}",
"Email": "{{json_ resource.customer .email}}",
"Name": "{{metadata.name}}",
"Type": "PRIVATE"
  }
}
```

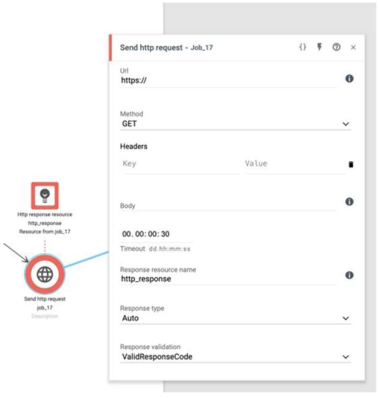
Type in Fortnox indicates whether it is a private person or company. In our

example, we have chosen not to take it from Shopify, but directly in the structure enter it as a private person. You can choose how you want to handle this.

We choose to create a new "Step" and select "Save as a resource" and at "Resource body" we write "fortnox_body". This is now saved as a resource for us. Also click in the box "Use syntax alt1".



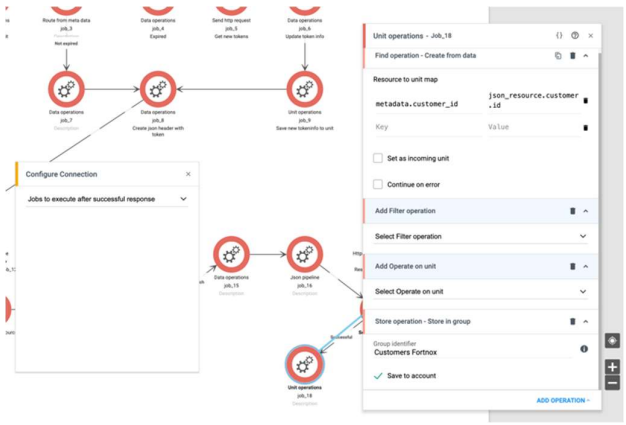
- Now we will make calls to Fortnox . We select job "Send http request". We write the url, https://api.fortnox.se/3/customers , which calls Fortnox's customer register and selects "POST" as "Method". In " Headers " we write
  content-type - application/ json
  authorization - metadata.access_token
  In "Body" we write "fortnox_body". This is the resource name that we saved in job_16.



- Now we want to create a customer from the response and store the customer ID in the group "Customers Fortnox". We choose job "Unit operation" and then "Create from data". Under "Resource to unit map" so we write at "key", "metadata.customer_id" and at value "json_resource.customer.id". At "Store

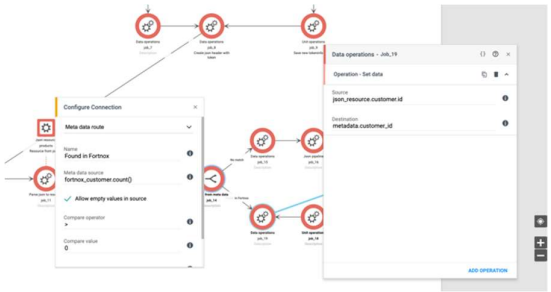
operation" you select "Store in group" and write "Customers Fortnox" and then click the box "Save to account" to save it to the group.

When you then connect jobs, you get a box with " Configure Connection" and there you select "Jobs to execute after successful response".



- Now we want to set the Customer ID on the customer. We then select a job "Data operation" and then we select "Set data". In source we take the customer ID from Shopify, and in destination we write "metadata.customer _id"

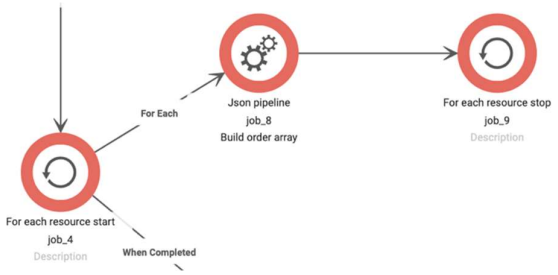
  When we then drag the arrow from "Route to metadata", job_14, in "Configuration settings" the conditions are specified. We select "Metadata data route" and name the route to "Found in Fortnox" and for the conditions we write at "Compare operator" the sign for greater than , > , and at " Compare value " we write 0. A customer ID greater than zero from Fortnox will be registered.



4. Now we will start creating the data structure so that it maps to Fortnox's structure. We go through this part of the workflow that handles the product data.

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com



- We start by choosing a job "For each resource start". This job loops through the order with the products so we get all the products from the order. We iterate over the "products" resource and name it "product".

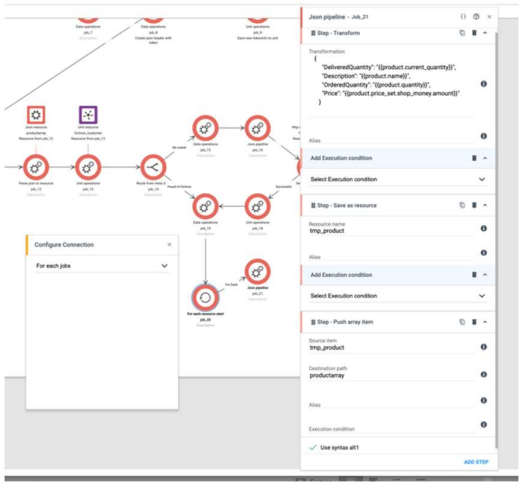
- Now we will create array for product. We select job "Json pipeline" and then step "Transform". On the left is structure from Fortnox and on the right is data from Shopify array.

```
 {
" DeliveredQuantity ": "{{ product.current _quantity }}",
"Description": "{{product.name}}",
" OrderedQuantity ": "{{ product.quantity }}",
"Price": "{{ product.price _set.shop_money.amount }}"
    }
```

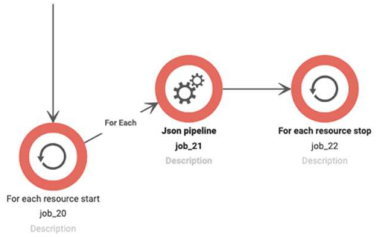
We save it as a resource and add a step "Save as a resource" and name the resource "tmp_product". Select another step "Push array item" where source should be "tmp_product" and at "Destination path" we write "productarray". When we then drag the arrow to this job, we configure it as "For each jobs".

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

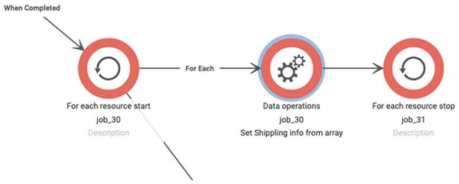
Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com



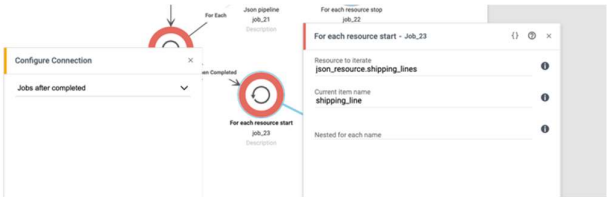
- To end the loop, we select job "For each resource stop”.



5. Now we want delivery data from array in Shopify data.



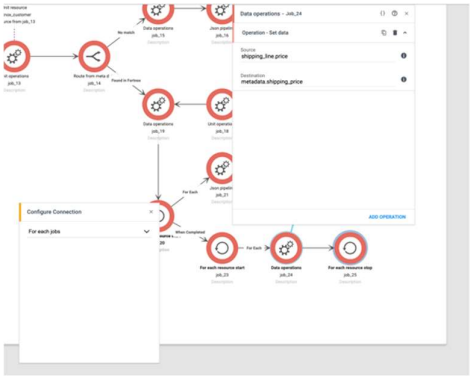
- We start by choosing a job "For each resource start”. This job loops through the price of the products so we get all the prices from the order. We iterate over the resource ”json_ resource.shipping _lines” and name it ”shipping_line”. When the arrow is drawn to this job, "Jobs after completed”.



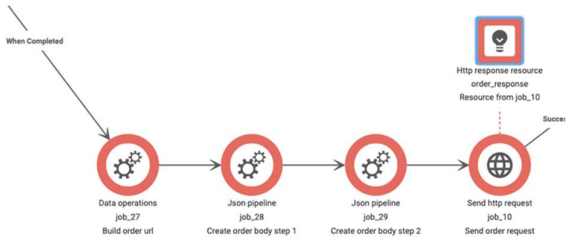
Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com

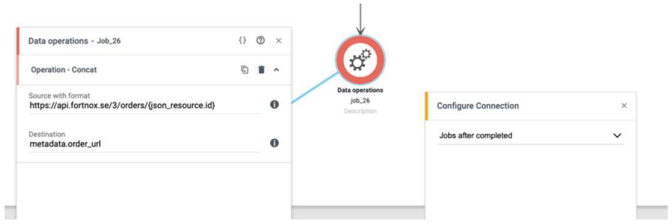
- The next jobs are "Data operation" and "Set data". In " Resource" we select the resource "shipping_ line.price" and create metadata that we name "metadata.shipping_price". In ”Configuration settings” we select ”For each jobs”. We end the loop with job "For each resort stop”.



6. Now it's time to build the structure for the order call.



- We start by choosing a "Data operation" job and a "Concat" operation. We specify the url for calling Fortnox orders with the Shopify ID of the order, https://api.fortnox.se/3/orders/{json_resource.id} , then we create metadata from this, metadata.order _url . When we then drag the arrow to this job, we select configuration "Jobs after completed”.

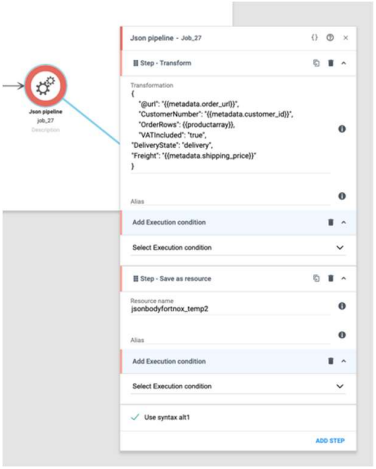


- Now we want to create the structure for the order. We select job "Json pipeline" and step "Transformation". On the left is structure from Fortnox and on the right is data from Shopify.

```
{
"@ url ": "{{metadata.order _url}}",
" CustomerNumber ": "{{metadata.customer _id}}",
" OrderRows ": {{productarray}},
" VATIncluded ": "true",
" DeliveryState ": "delivery",
"Freight": "{{metadata.shipping _price}}"
    }
```

We state VATIncluded because that VAT is included in the total price. We previously chose that this was sold to consumers and then the stated price includes VAT. We also state in plain text that the DeliveryState is delivery, and this is so that it will be considered that the order has been paid.

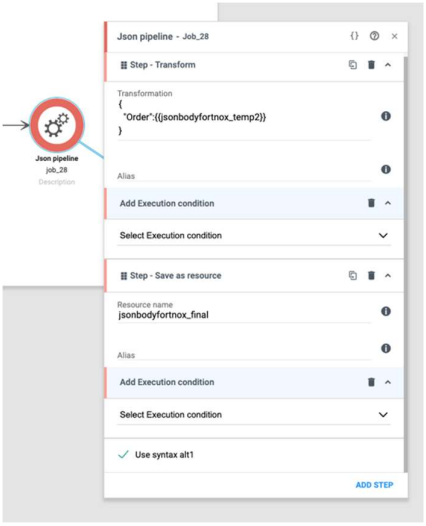
Now we select another step which is "Save as resource" and select as" Resource name" - "jsonbodyfortnox_temp2". Click in the box "Use syntax alt1".



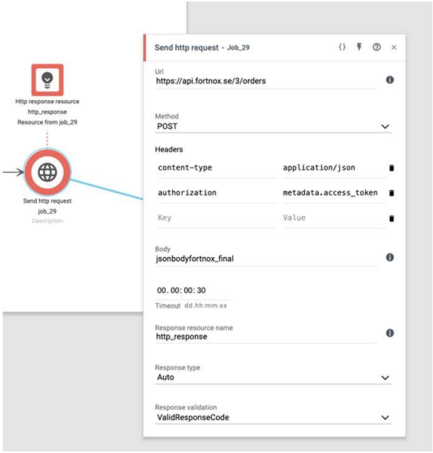
- In the next job, we create the order that we will use in calls to Fortnox. We select a "Json pipeline" job and select the step "Transform". There we take the resource we previously created and write from Shopify.

```
{
"Order ":{{jsonbodyfortnox_temp2}}
}
```

Then we save it as resource which we call "jsonbodyfortnox_final". Choose step "Save as a resource". Click in box "Use syntax alt1.



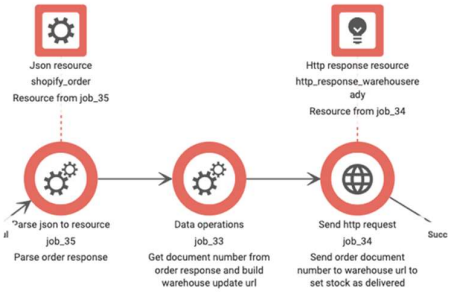
- Now we will make the order-call to Fortnox and select the job " Send http request ". We choose url https://api.fortnox.se/3/orders and a POST-call, at the headers we write like previous calls to Fortnox and in the body we write "jsonbodyfortnox_final".



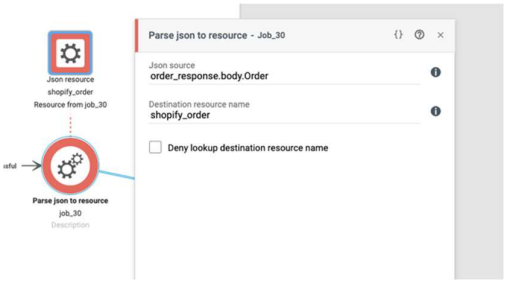
7. Now we will receive the order as delivered and update the warehouse.

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

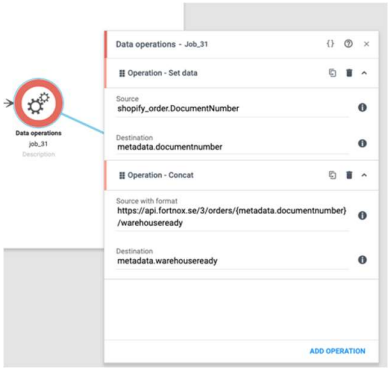
Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com



- We choose job "parse json to resource" and in json source we write "order_ response.body.Order" and at destination "shopify_order".

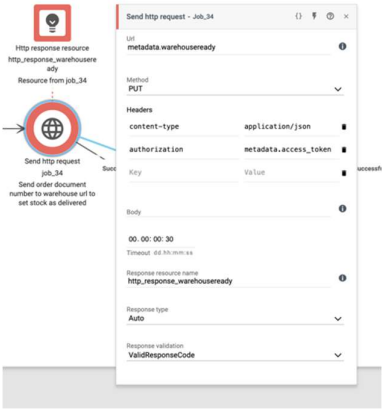


- Now we're going to fetch data from Shopify to update the inventory. In Shopify, it's "DocumentNumber" where it is. We select job "Data operations" and then "Set data" as operation. In "Source" we write "shopify_ order.DocumentNumber" and in "Destination" we create metadata "metadata.documentnumber". We choose a new operation "Concat" where we specify calls to Fortnox, https://api.fortnox.se/3/orders/{metadata.documentnumber}/warehouseready and then in "Destination" name it to "metadata.warehouseready" .

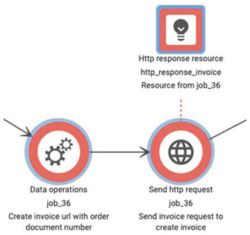


- Now we will update the warehouse and then make a PUT call to Fortnox. We have previously created metadata of the url so we enter "metadata.warehouseready" select "PUT" and fill in "Headers" as before. We

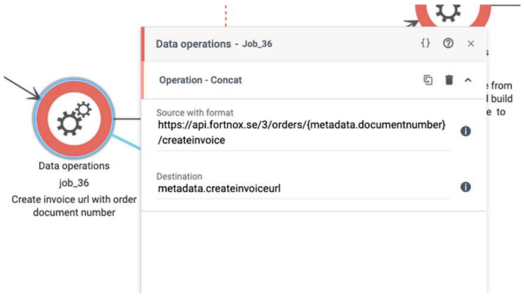
write the response to "http_response_warehouseready".



8. Now Fortnox will create an invoice so that the order is booked as paid.



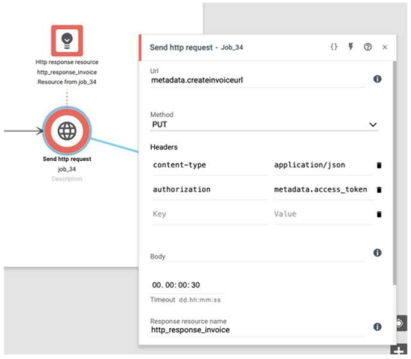
- We start by selecting job "Data operation" and operation "Concat". In "Source with format" we specify Fortnox url, https://api.fortnox.se/3/orders/{metadata.documentnumber}/createinvoice and make metadata of the url, “metadata.createinvoiceurl”.



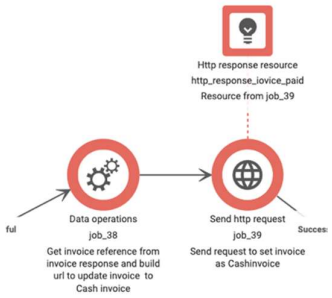
- Now we will create an invoice and then make a PUT call to Fortnox. We have previously created metadata of the url so we enter "metadata.createinvoiceurl" select "PUT" and fill in "Headers” as before. We write the response to "http_response_invoice".

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

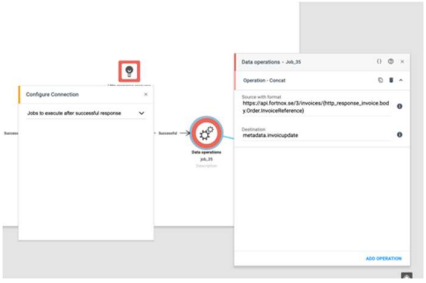
Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com



9. Now we will update the invoice to a cash invoice so the order is paid when it arrives at Fortnox.



- We select job "Data operation" and operation "Concat". In "Source with format" we specify Fortnox url, https://api.fortnox.se/3/invoices/{http_response_ invoice.body.Order.InvoiceReference} and make metadata of the url, “metadata.invoicupdate”.

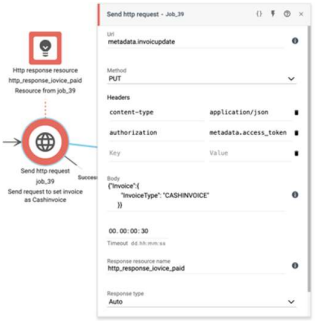


- From the previous call, we pick the invoice reference and build a url that makes the status of the invoice become cash invoice. We then make a PUT call to Fortnox. We use the metadata url so we enter "metadata.invoicupdate" select

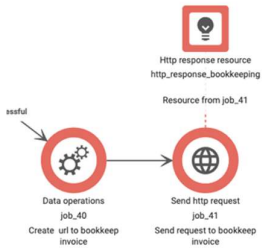
"PUT" and fill in "Headers" as before. In "Body" we write.

{"Invoice":{
"InvoiceType": "CASHINVOICE"
}}

And "Response resource name" then we name it to "http_response_iovice_paid".



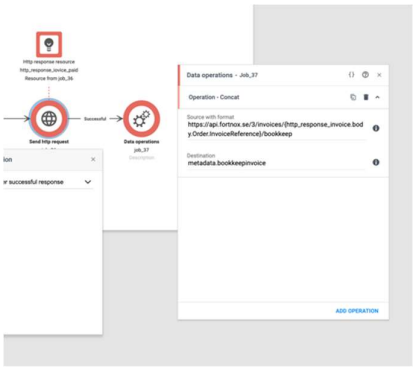
10. Finally, the order must be booked in Fortnox.



- We select job "Data operation" and operation "Concat". In "Source with format" we enter the url to call Fortnox, https://api.fortnox.se/3/invoices/{http_response_invoice.body.Order.InvoiceReference}/bookkeep . Then we make metadata of this url and name it "metadata.bookkeepinvoice" in "Destination".

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

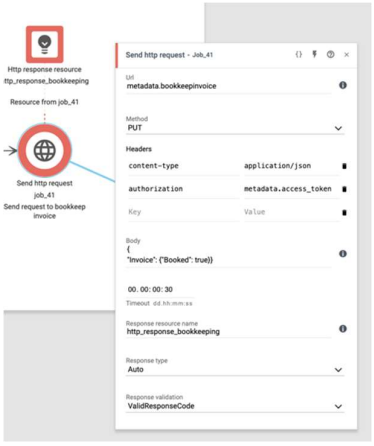
Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com



- Now there is only the last job left before the workflow is complete and it is a PUT call to post the order. We select job "Send http request". We then make a PUT call to Fortnox. We use the metadata url so we enter "metadata.bookkeepinvoice" select "PUT" and fill in "Headers" as before. In "Body" we write.

  {
  "Invoice": {"Booked":true}}

  In ”Response Resource Name” we name it ”http_response_bookkeeping”.



Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com

3. Final words

Building workflow is very flexible in Bosbec. This gives a degree of freedom to adjust and change when it changes in the systems you integrate or in the authentication you have against the systems. Other systems can also be added to the integration for increased automation. For example. as in this example, you might want to book purchases from a POS system in Fortnox so that it is directly in the accounting, then you just add the system and build on your workflow.

Bosbec AB
Address:
Norrtullsgatan 6, 5 tr
113 29 Stockholm
VAT No: SE556663223701

Phone: +46 (8) 22 25 25
Email : support@bosbec.com
www.bosbec.com