# Mambu Exercises

Make sure that the necessary configuration sub-processes (***check the MPO training ppt***) are created, since these sub-processes will be used in all the exercises below.

## 1 - API call (POST)

Create an MPO process that performs an API call (POST) that creates a task for a given user in Mambu.

**Step 1 -** Insert Call Process in the flow and select the *“FS-get Mambu config”*

**Step 2 -** Insert API Call after the Call Process

**Step 3 -** On the API Call fill the blocks with the following information:

**URL API**

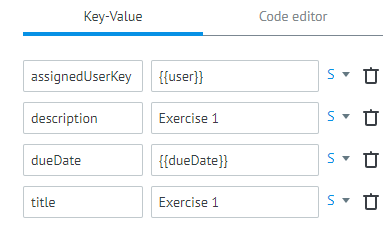
https://{{config.url}}/api/tasks

**Request format:** Default

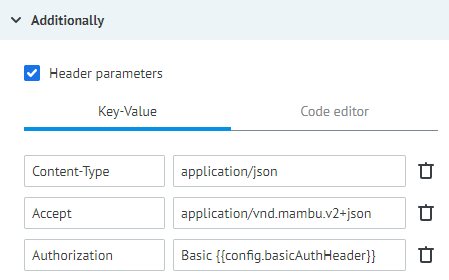
**Request method:** POST

**Content-Type:** Application/Json

**Parameters**



**Additionally**



**Step 4 -** On the the “Task parameters” () menu add the inputs (*“user”* & *“dueDate”*) necessary to run the process

## 2 - API call (GET)

Create an MPO process that performs API call (GET) that obtains the information of a specific transaction performed.

**Step 1 -** Insert Call Process in the flow and select the *“FS-get Mambu config”*

**Step 2 -** Insert API Call after the Call Process

**Step 3 -** On the API Call fill the blocks with the following information:

**URL API**

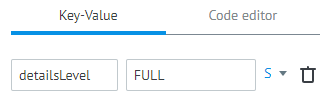
https://{{config.url}}/api/deposits/transactions/{{transactionId}}

**Request format:** Default

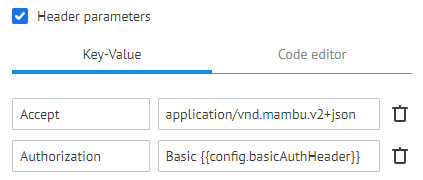
**Request method:** GET

**Content-Type:** Application/Json

**Parameters**



**Additionally**



**Step 4 -** On the the “Task parameters” () menu add the input (*“transactionId”* – select an id from a transaction present in Mambu’s Sandbox) necessary to run the process

## 3 – Setup conditions

Using the MPO process created in exercise 2, create a condition blocker that will return the transaction details if it is higher than 100 and false if it is lower.

Follow **Steps 1** to **4** of exercise 2

**Step 5 –** Add a Condition after the API Call that defines the requested condition (amount >100 get details)

## 4 – Loop’s part 1

Using a similar MPO process to the one created in Exercise 2 replace the ***getById*** call by a ***getAll*** call and perform a loop that will apply the condition created in 3 to all the transactions present in the ***getAll*** call.

Follow **Steps 1** to **3** of exercise 2 but on the API Call replace the following information:

**URL API**

https://{{config.url}}/api/deposits/{{depositAccountId}}/transactions

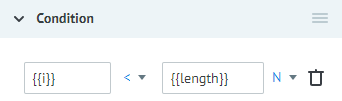
**Step 4 –** Insert a Code block after the API Call with the following:

*data.transactions = [];*

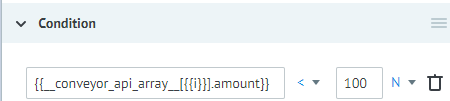
*data.length = data.\_\_conveyor\_api\_array\_\_.length;*

*data.i = 0;*

**Step 5 –** Insert a Condition block after the Code with:



**Step 6 –** Insert another Condition block and link it to the previous Condition (**Step 5**):



**Step 7 –** Insert a Code block, linked to the previous Condition (**Step 6**), that will join all transactions above 100:

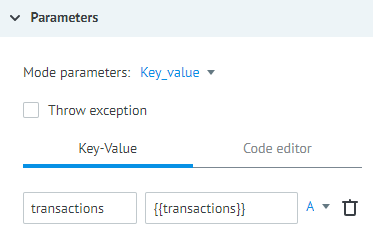
*data.transactions = data.transactions.concat(data.\_\_conveyor\_api\_array\_\_[data.i]);*

**Step 8 –** Insert another Code block that will increment to *“i”* and will be linked to the Condition (**Step 6**), when transaction are below 100, and to the Code block previously created (**Step 7**). Code for **Step 8**:

*data.i = data.i +1;*

**Step 9 –** Link the Code block (**Step 8**) to the Condition (**Step 5**)

**Step 10 –** Link the Condition (**Step 5**) to a Reply to Process containing the following:



**Step 11 –** Link the Reply to Process (**Step 10**) to an End: Success block

**Step 12 -** On the the “Task parameters” () menu add the input (*“depositAccountId”* – select an id from a deposit present in Mambu’s Sandbox) necessary to run the process

## 5 - Loop’s part 2

On the process created in 4, in case the transaction value is higher than 100 create a user task using the process created in 3 – Use the call process component.

**Step 1 -** Insert Call Process in the flow and select the *“FS-get Mambu config”*

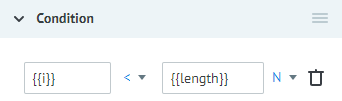
**Step 2 –** Insert another Call Process after the Call Process (**Step 1**) and select the process created on exercise 4

**Step 3 –** Insert a Code block after the Call Process with the following:

*data.length = data.transactions.length;*

*data.i = 0;*

**Step 4 –** Insert a Condition block after the Code with:



**Step 5 –** Insert an API Call after the Condition, insert following information:

**URL API**

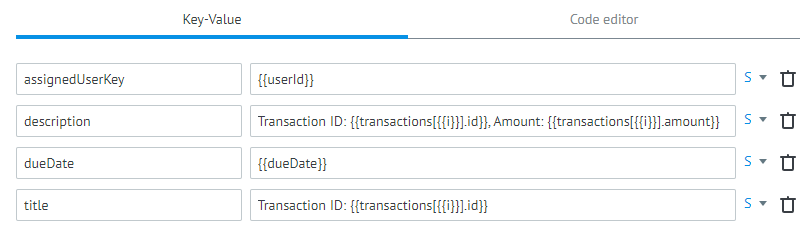
https://{{config.url}}/api/tasks

**Request format:** Default

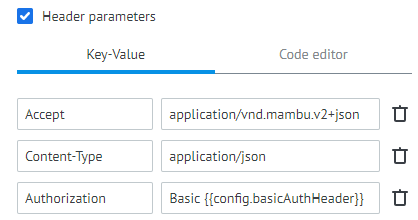
**Request method:** POST

**Content-Type:** Application/Json

**Parameters**



**Additionally**



**Step 6 –** Insert a Code block that will increment to *“i”* and link it back to the Condition **(Step 5)**:

*data.i = data.i +1;*

**Step 7 –** Link the Condition **(Step 5)** to an End: Success block

**Step 8 -** On the the “Task parameters” () menu add the input (*“depositAccountId”*, *“dueDate”* and *“userId”*,) necessary to run the process

For access to the solved examples of the above exercises, request access to the ***‘MPO Learning Exercises’*** group.

**Note**: Do not perform any changes on the exercises present in the “MPO Learning Exercises” folder as they are only to be visualized. In order to test changes on those processes please deploy the zip file in the same folder as the word document on your developer folder in the MPO environment.