Tabscanner- Receipt OCR

Blue Prism Integration Guide

June 9th 2019

Version 1.1

authored by Tabscanner

Tabscanner- Receipt OCR
Blue Prism Integration Guide

VBO Dependencies

Newtonsoft.Json.dll
System.Core.dll
System.Collections.Specialized
Newtonsoft.Json.Linq

Overview

Tabscanner is an API based system which runs in the cloud. A Blue Prism business object is available to speed the integration of the Tabscanner API with your implementation, however, there is nothing to stop a Blue Prism developer calling the API directly from their own Business Object.

Calls to process a receipt are a two stage process:

Stage 1
Post the receipt image you wish to process and receive a queue token. The image is then put in a processing queue.

Stage 2
Retrieve the results by polling the API for the result which is returned as a string of json formatted text. It is left up to the Blue Prism developer how to process the json text.

The Tabscanner Blue Prism Business Objects have been split into two Actions to handle these two stages:

- PostImage
- GetResult

This document will describe how to configure these Actions to process a receipt image. For full documentation for the Tabscanner API please visit https://tabscanner.com and register.

Configuring Blue Prism to Call Tabscanner

Step 1. Register with Tabscanner

Go to https://tabscanner.com and register for an account. Once registered go to the API help section to get your API Key.

Step 2. Import the Tabscanner Blue Prism Business Object

In Blue Prism go to File→Import and browse the downloaded release file "BPA Object - Tabscanner- Receipt OCR"

Tabscanner is now available as a Business Object within Blue Prism

Step 3. Configuring the Tabscanner PostImage Action

In the Studio tab of Blue Prism:

- Right click on Processes and select "Create Process"
- Name the process Tabscanner Test and click through to finish
- Double click the new process to edit it
- Add a Data Item. We will use this to hold our test image
- Double click on the Data Item and name it receiptImage
- Set the Data Type to Binary
- Import an jpeg image of a receipt into the Data Item
- Click OK to return to the Process page.
- Add two Data Items named token and errorMessagePost of data type Text

We are now ready to create the PostImage Action

- Add a new action
- Double click on the action
- In the Business Object drop-down select Tabscanner- Receipt OCR.
- In the Action drop down select PostImage.

Fill out the following details in **Inputs**:

Name	**Data Type**	**Value**
url	Text	"https://api.tabscanner.com"
APIKey	Text	Enter the API key from step 1

file	Binary	[receiptImage]
contentType	Text	“image/jpeg”

And in **Outputs**:

Name	**Data Type**	**Store In**
result	Text	token
errorMessage	Text	errorMessagePost

- Click OK to return to the Process page
- Draw a link from Start to PostImage
- Draw a link from PostImage to End

Step 4: Testing the PostImage Action

- Check that the Page does not contain any error
- Click Reset
- Click Go
- The Page should run, upload the image and the token Data Item should be automatically populated with a token e.g. sLAsOtseVrU18zKz

Step 5. Configuring the Tabscanner GetResult Action

- Add two Data Items named result and errorMessage of data type Text
- Add a new action
- Double click on the action
- In the Business Object drop-down select Tabscanner- Receipt OCR.
- In the Action drop down select GetResult.

Fill out the following details in **Inputs**:

Name	**Data Type**	**Value**
url	Text	“https://api.tabscanner.com”
APIKey	Text	Enter the API key from step 1
token	Text	[token]

And in **Outputs**:

Name	**Data Type**	**Store In**
result	Text	result

errorMessage	Text	errorMessage

- Remove the existing Link between PostImage and End
- Draw a Link between PostImage and GetResult
- Draw a Link between GetResult and End

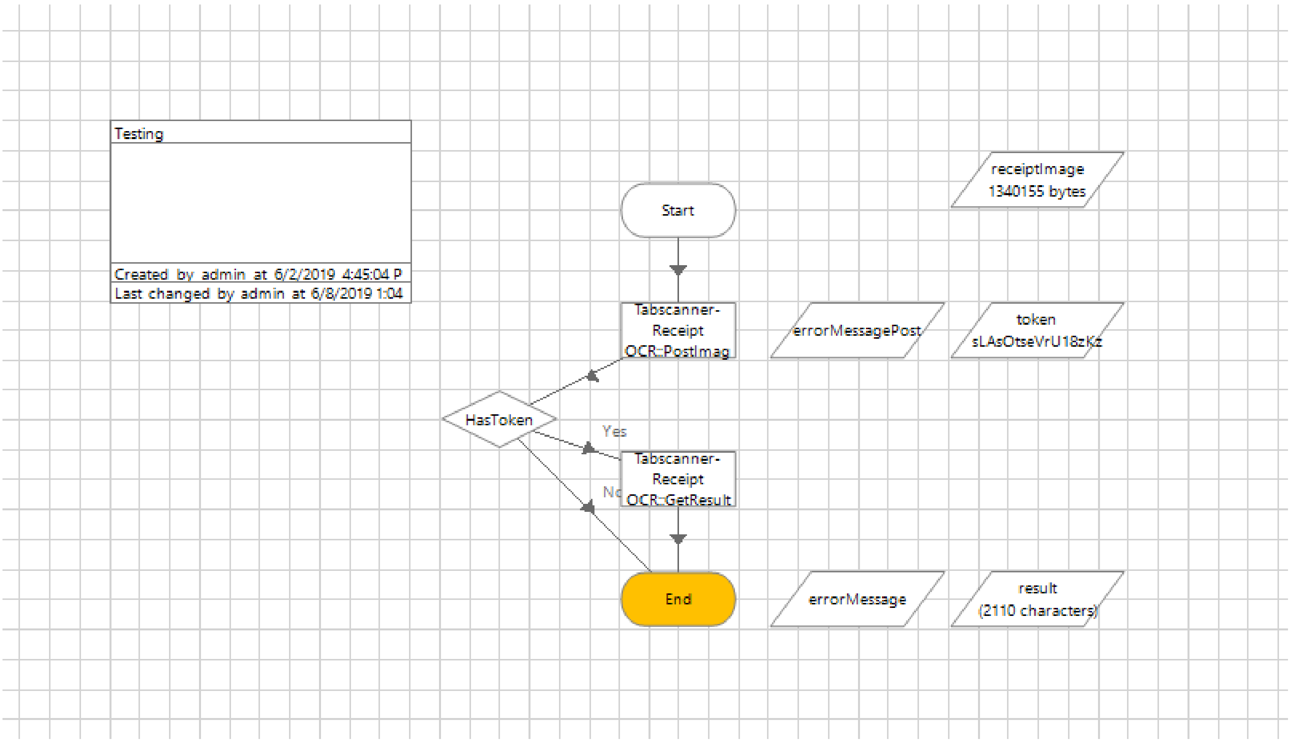
Step 6. Testing the Process

- Check that the Page does not contain any error
- Click Reset
- Click Go
- The Page should run, upload the image and the result Data Item should be automatically populated with a json formatted text e.g.
`{"message":"SUCCESS: Result available","status":"done","status_code":3,…`

Step 7. Extra Configuration

- Configure a Decision between PostImage and GetResult so that in the event of a failed upload the GetResult action is not called

The final process will look like the below screen-shot:


Testing
Created by admin at 6/2/2019 4:45:04 P
Last changed by admin at 6/8/2019 1:04
Start
receiptImage
1340155 bytes
Tabscanner-Receipt OCR::PostImag
errorMessagePost
token
sLAsOtseVrU18zKz
HasToken
Yes
No
Tabscanner-Receipt OCR::GetResult
End
errorMessage
result
(2110 characters)