

## Pierrick Lozach

---

**To:** Marc Sassoon  
**Subject:** RE: MS Bot => BYOB steps

### Official Documentation:

<https://intranet.genesys.com/display/PureCloud/BYOB+Customer+API+Spec>

### Requirements:

- A PowerVA bot: <https://web.powerva.microsoft.com/>
- An AWS account with access to API Gateway, Lambda, CloudWatch and DynamoDB

### Getting Started:

- Create a PowerVA bot
  - Publish your bot at least once
  - Get Token (Secret 1) from PowerVA web site (Click Manage/Security/Web Channel Security and copy "Secret 1")
  - Open the Automate\_BYOB2MS.py file and replace the MS\_BOT\_AUTHORIZATION\_SECRET value with "Bearer {YOUR SECRET 1}"
- AWS
  - Create a new DynamoDB table (name is "automate-byob2ibm-sessions" by default unless you change it in bot\_sessions.py)
  - Create a new Lambda function (give it a meaningful name like "MSGCConnector")
    - Add both .py files from the "src" directory
    - Add the DynamoDBFullAccess policy to the role used by the lambda function (you can also restrict permissions to the "automate-byob2ibm-sessions" table only)

# Automate\_BYOB2IBMHandler

## ► Function overview [Info](#)

Code

Test

Monitor

**Configuration**

Aliases

Versions

General configuration

Triggers

**Permissions**

Destinations

Environment variables

Tags

VPC

Monitoring and operations  
tools

Concurrency

## Execution role

Role name

[Automate\\_BYOB2IBMHandler-role-2kpq38vm](#) 

## Resource summary



**AWS Key Management Service**

2 actions, 1 resource

To view the resources and actions that your function has permission to a

By action

**By resource**

- Add an API gateway to your lambda
  - Create a POST /utterance that points to your lambda function

Choose the integration point for your new method.

**Integration type** ☒ Lambda Function ⓘ

☐ HTTP ⓘ

☐ Mock ⓘ

☐ AWS Service ⓘ

☐ VPC Link ⓘ

**Use Lambda Proxy integration** ☐ ⓘ

**Lambda Region** eu-central-1 ▼

**Lambda Function** MSGCBotConnector

**Use Default Timeout** ☒ ⓘ

- Edit your new POST action, go to “Integration Request” and add the following under “Mapping Templates”

```
{
  "body": $input.json('$'),
  "headers": {
    #foreach($param in $input.params().header.keySet())
    "$param": "$util.escapeJavaScript($input.params().header.get($param))"
    #if($foreach.hasNext),#end
  }
}
```



APIs

Custom Domain Names

VPC Links

API: **GCConnectorAPI**

Resources

Stages

Authorizers

Gateway Responses

Models

Resource Policy

Documentation

Dashboard

Settings

Usage Plans

API Keys

Client Certificates

Settings

Resources

Actions

[← Method Execution](#) /postutterance - POST - In

Provide information about the target backend that this method will c

Integration type ☒ Lambda Function ⓘ☐ HTTP ⓘ☐ Mock ⓘ☐ AWS Service ⓘ☐ VPC Link ⓘUse Lambda Proxy integration ☐ ⓘ

Lambda Region eu-central-1 ✎

Lambda Function MSGCBotConnector ✎

Execution role ✎

Invoke with caller credentials ☐ ⓘ

Credentials cache Do not add caller credential

Use Default Timeout ☒ ⓘ

▶ URL Path Parameters

▶ URL Query String Parameters

▶ HTTP Headers

▼ Mapping Templates ⓘ

Request body passthrough ☐ When no template match☐ When there are no temp☒ Never ⓘ

Content-Type

application/json

⊕ Add mapping tem

application/json

Generate template: 

```
1 {  
2   "body": $input.json('$'),  
3   "headers": {
```

- Request Body Passthrough should be set to "Never"
- Genesys Cloud
  - Create a new "Genesys Bot Connector" integration
    - Bot Connector Handle Utterance URI: <https://{YOUR API GATEWAY URL}/postutterance>
    - Save and activate the integration
  - Bot configuration
    - Get the PVA bot id from the Microsoft PowerVA (after "Default") URL: <https://web.powerva.microsoft.com/environments/Default-xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxx/bots/{id}/>
    - Get your access token from Genesys Cloud (F12/Application/Local Storage/https://apps.mypurecloud.ie/pc\_auth/authenticated/access\_token)
    - Call the following Genesys Cloud API (check if intents and slots are correct first):

PUT <https://api.mypurecloud.ie/api/v2/integrations/botconnector/{YOUR ID}/bots> GENESYS BOT CONNECTOR INTEGRATION

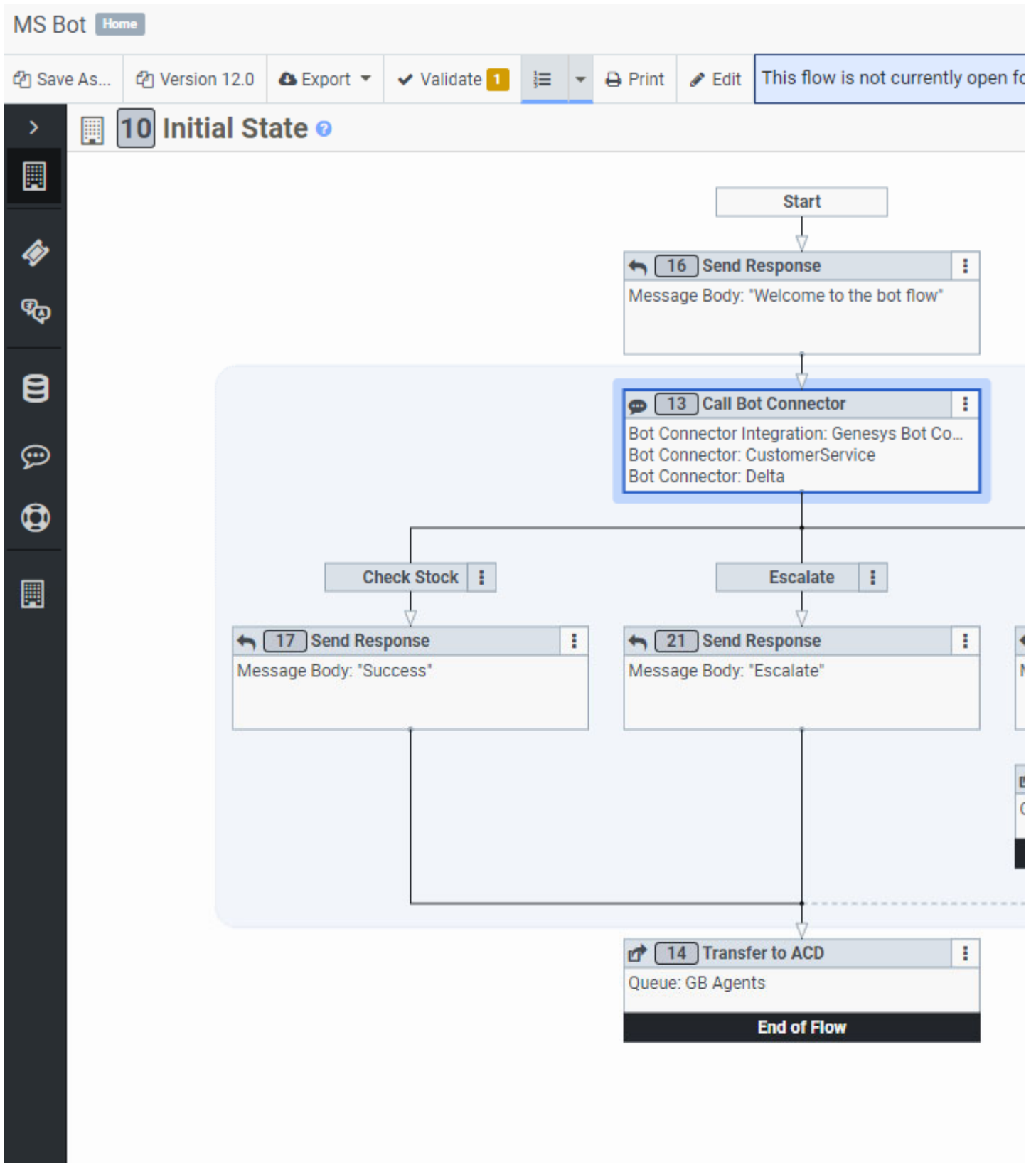
Authorization: bearer {YOUR GC ACCESS TOKEN}

Content-Type: application/json

```
{
  "chatBots": [
    {
      "id": "{PVA BOT ID}",
      "name": "{DISPLAY NAME}",
      "versions": [
        {
          "version": "Delta",
          "supportedLanguages": [
            "en-us",
            "es"
          ],
          "intents": [
            {
              "name": "Check Stock",
              "slots": {
                "The_answer": {
                  "name": "The_answer",
                  "type": "String"
                }
              }
            },
            {
              "name": "Escalate",
              "slots": {
                "The_answer": {
                  "name": "The_answer",
                  "type": "String"
                }
              }
            }
          ]
        }
      ]
    }
  ]
}
```

```
]
}
]
}
```

- You should receive a 204 No Content if all works well
- Create a new inbound message flow and use the new bot connector integration



- Web Messaging
  - Create a configuration
  - Create a deployment
    - Send messages to your message flow

Deployment Details - Genesys Cl xMS Bot - 12.0 - Call Bot Connect x+

←→↻🏠

apps.mypurecloud.ie/directory/#/admin/messenger/deployments/275b7649-01cc-45f9-

GenesysPersoTravelCC PortalAppsGC AppsDev CenterZoomUltimate

ActivityDirectoryDocumentsPerformanceReportsAppsAdmin

Message / Messenger Deployments / MS GC Bot (Pierrick)

Platforms

SMS Number Inventory

Threading Timeline

Messenger Configurations

Messenger Deployments

**Restrict domain access\***

List your trusted domains where Messenger can run.

☒ Allow all domains

☐ Run on specified domains only

**Apply your flow logic\***

Select a published inbound message flow to trigger bots or connect

MS Bot x

**Get your deployment key**

If you build a custom Messenger, you need this deployment key to a

275b7649-01cc-45f9-86b6-b4d65be6557b

Copy to Clipboard

**Deploy your snippet**

Copy and paste this snippet on all of the website pages where you v

deploy the snippet, you can refine the configuration from within this

```
<script type="text/javascript" charset="utf-8">
(function (g, e, n, es, ys) {
  g['_genesysJs'] = e;
  g[e] = g[e] || function () {
    (g[e].q = g[e].q || []).push(arguments)
  };
  g[e].t = 1 * new Date();
  g[e].c = es;
  ys = document.createElement('script'); ys.async
```

Copy to Clipboard



- Chat Page
  - Open the index.html and replace the <script> section with the code from your web messaging deployment (after the “Welcome to the test web chat” text)

Copy to Clipboard

- Open the index.html file in a browser (or on a web server)
- Chat away! 😊

