



IGNITION INTEGRATION WITH MICROSOFT TEAMS

Easy Guide

Authored by:

Wishah Naseer
And
Maryam Saeed

JANUARY 26, 2023
QANARE ENGINEERING
V1.0

Contents

| | |
|--|----|
| About the Authors: | 2 |
| Chapter 1: Introduction | 3 |
| 1.1. Motivational Background:..... | 3 |
| 1.2. Introduction | 3 |
| 1.3. Approach using Ignition: | 3 |
| Chapter 2: Methodology& Implementation | 4 |
| Step 01: Create and Configure Webhook in MS Teams Channel..... | 4 |
| Step 02: Teams API Script in Ignition Project..... | 6 |
| Chapter 3: Results | 9 |
| References | 10 |

About the Authors:

Maryam Saeed and Wishah Naseer are Computer Systems Engineers. They are working as Junior SCADA Engineers at Qanare Engineering and have a keen interest in Industrial Automation, Digital Transformation and Industry 4.0.

Chapter 1: Introduction

1.1. Motivational Background:

We came across a post from OPC Router (<https://www.opc-router.com/microsoft-teams-rest-connection-en/>) where integration was shown between OPC Router and Microsoft Teams, we sought to replicate the same between Ignition SCADA and MS Teams to find out if it was possible.

1.2. Introduction

Ignition can be configured to send its tag values, alerts or notifications to MS Teams. Ignition is an industrial application platform that is used for collecting data, designing and deploying industrial applications throughout an enterprise. It empowers the creation of any kind of industrial application including SCADA (Supervisory Control and Data Acquisition), MES (Manufacturing Execution Systems), IIoT (Industrial Internet of Things), reporting, alarming and more. *[1]

MS Teams is a communications platform, it was developed to coordinate between the people of a company, to post notification, share files and to chat or video call. MS Teams has a REST API using which we can post updates on any team channel automatically. Further automation can be done using the MS Teams API. *[2]

1.3. Approach using Ignition:

In order to establish communication between Teams and Ignition we follow the steps below.

1. Create and Configure Webhook in MS Teams.
2. Script the Teams API configuration in an Ignition project.

Chapter 2: Methodology & Implementation

Step 01: Create and Configure Webhook in MS Teams Channel

- a. Now for pushing the results into Microsoft Teams, first create a team and make a channel within it.

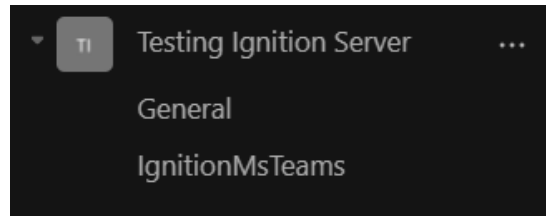


Figure 1: Channel created "IgnitionMSTeams"

- b. Go to more options of the channel and select "Connectors".

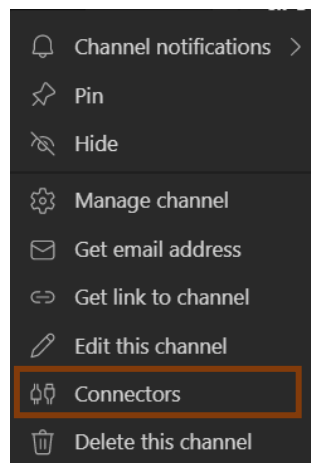


Figure 2: Option connectors selected.

- c. From "Connectors" add "Incoming Webhook"

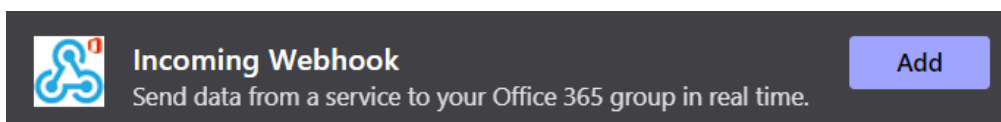


Figure 3: Incoming Webhook added

d. Configure the Webhook and create it.

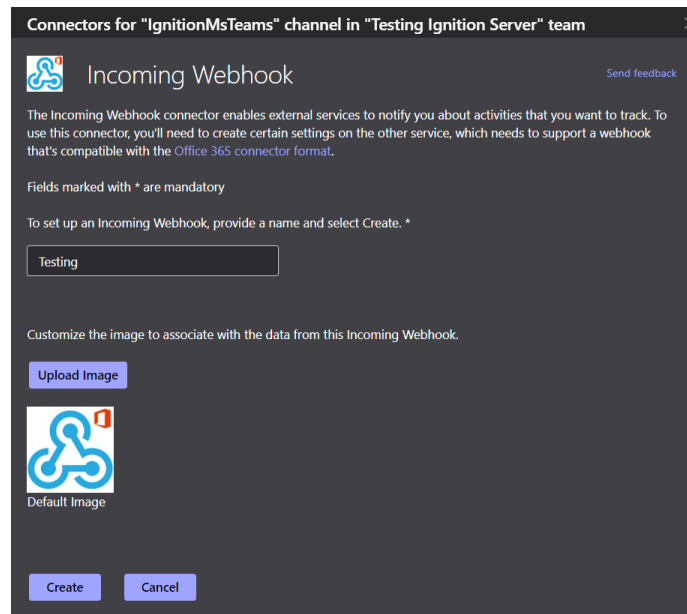


Figure 5: Configured and Created Webhook

e. Copy the Webhook link for future use.

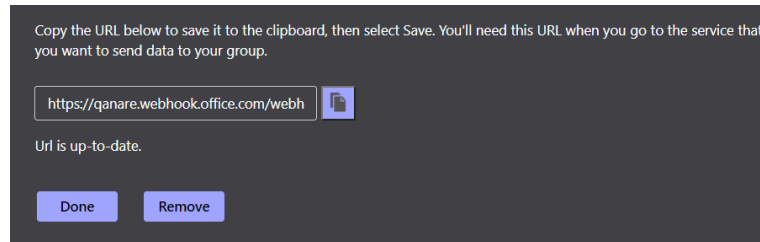


Figure 6: Link Copied

Step 02: Teams API Script in Ignition Project

We are going to test our functionality on a normal button in Ignition. For this purpose we make a project in Ignition and initially integrate using a normal button.

Open the Designer and go to Vision. Next, create a window named “Testing”.

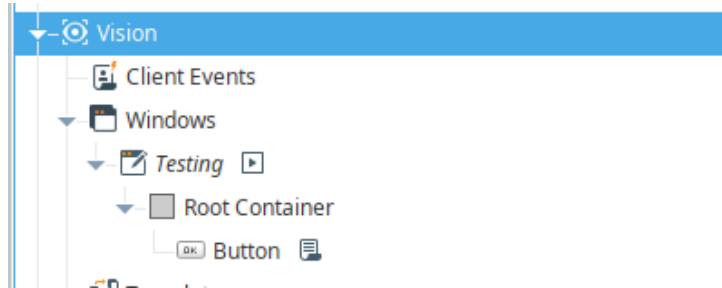


Figure 7: Vision Window Created

Now, from component palette add a normal button on your window.

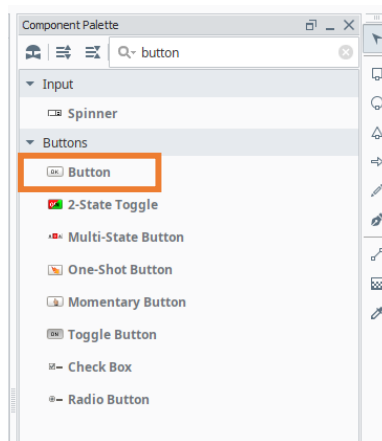


Figure 8: Normal Button Added

Right click on the button and choose “Scripting” option.

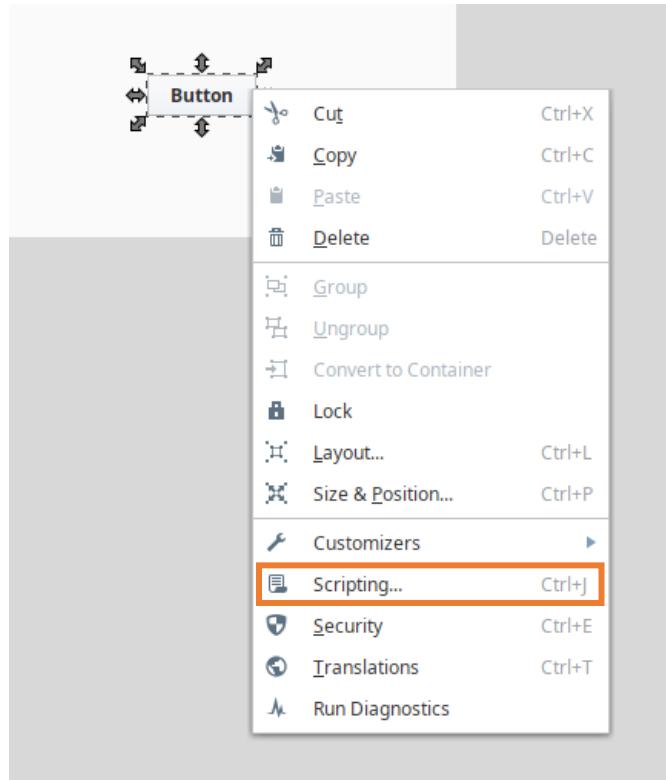


Figure 9: Button Scripting

In the left-side panel, choose the “actionPerformed” option as the Event Handler. An event is fired when the ‘action’ of the component occurs, and it depends on the type of component used. In this case, when the button is clicked via a mouse click or key is pressed. *[3]

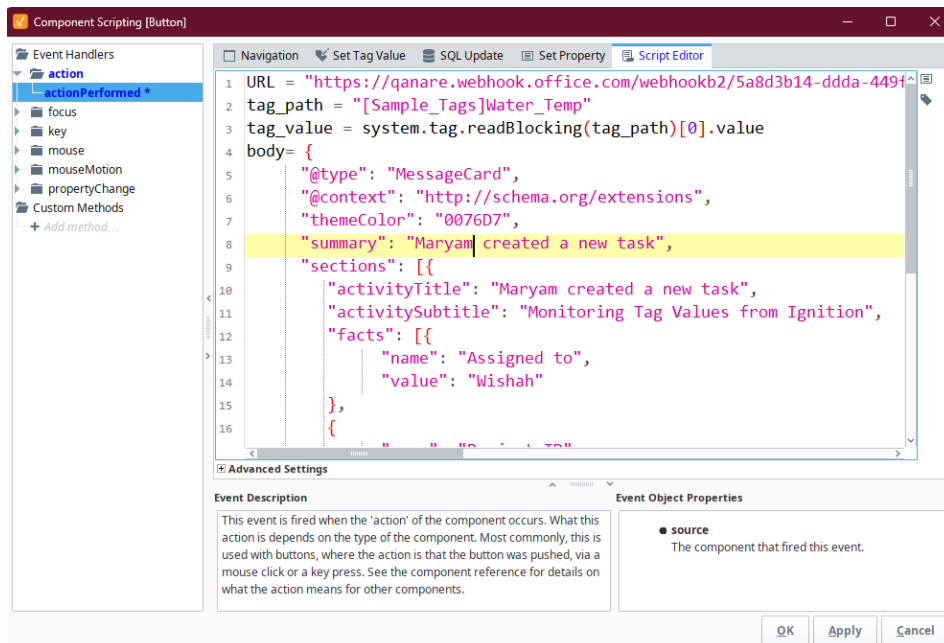


Figure 10: Button (Component Scripting)

In the Script Editor, the Python code will be written and executed. This is the action event to code the connection. Click on the “Script Editor” tab and enter the following Python code:

```
URL ="PASTE YOUR URL HERE"
tag_value= //Initialize value of any tag to this tag_value variable
tag_name = //Initialize name of the tag to this tag_name variable
body={
"@type": "MessageCard",
"themeColor": "0076D7",
"summary": "Maryam created a new task",
"sections": [{
"activityTitle": "Maryam created a new task",
"activitySubtitle": "Monitoring Tag Values from Ignition",
"facts": [{
"name": "Assigned to",
"value": "Wishah"
}],
{
"name": "Project ID",
"value": "01"
},
{
"name": "Tag Name",
"value": "%s"%tag_name
},
{
"name": "Tag Value",
"value": "%s"%tag_value
}],
}]
}
system.net.httpPost(URL, "Application/Json", body, 10000, 60000)
```

To save the connection script, click on the “Apply” and the “OK” button. Then put the project into preview mode, and click on the button, you should receive a notification in the Microsoft Teams channel.

Chapter 3: Results

The following figure shows the post on MS Teams channel after the steps in the above document are followed.

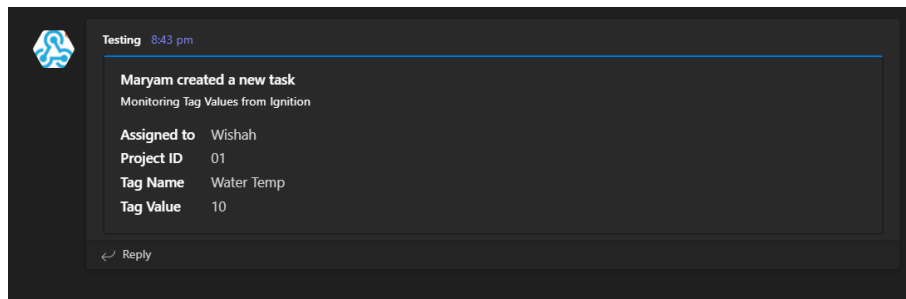


Figure 4: Post results on MS Teams channel

References

1. “Ignition by Inductive Automation”, <https://inductiveautomation.com/ignition/>
2. “REST APIs of MS Teams”, <https://learn.microsoft.com/en-us/graph/api/resources/teams-api-overview?view=graph-rest-beta>
3. “Sending Ignition notifications to Microsoft Teams”, <https://element8.co.za/how-to-guide-sending-ignition-notifications-to-microsoft-teams/#h-6-downloads>