This python script distribution can be used as a template to create an artifact for MaaS360 App Exchange portal to enable MaaS360 integration with your product/service.

Follow the instructions below to customize the template to suit your integration requirements.

Before you start:

You must have python 2.7 or higher to run these scripts. If you do not have it on your computer, download it from https://www.python.org/downloads/

Contents of the distribution:

The distribution contains 3 python script files and a ReadMe.txt for the customer:

a) Runner.py: This python script will be run by the customer to enable the integration between MaaS360 and your service/product. You do not need to modify it.

b) MaaS360APIs.py: This python script contains the base functionality for calling MaaS360 web service APIs. DO NOT edit this file.

c) PartnerConfig.py: This python script contains the web service calls specific to each partner integration. You need to modify this file to suit your requirements. More details are given below.

d) ReadMe.txt: This readme file has instructions for the customer. You should replace the text <PARTNER NAME> in this file with your service/product. You may choose to update this file with more details about your service/product if required.

Updating the code:

1. The file PartnerConfig.py contains sample web service calls to enable integration with MaaS360. It has sample calls for the following:

- Creation of new custom attributes

- Creation of device group

- Creation of alert on the UI

- Uploading a new iOS or Android app to the MaaS360 App Catalog

- Distribution of uploaded app to devices

Note that these are the most common configuration settings used by our partners. If configuration for your service/product needs create/update of other settings in MaaS360, you can add the relevant web service calls to your code.

2. The file Runner.py only calls one method called createPartnerConfigurations() in PartnerConfig.py. You must ensure that this method contains all the code for configuring the required integration parameters within MaaS360.

3. You can update the existing code in PartnerConfig.py and modify the parameter values or data being passed for various web service calls. Depending on the number of parameters or device groups you need to create, you may need to add or remove the web service calls.

4. If you need to make a new type of web service calls that is not present in MaaS360APIs.py, then you must add the base code to call the web service. It is recommended that you add it to PartnerConfig.py rather than to MaaS360APIs.py.

5. The details of MaaS360 web service APIs can be found in the latest MaaS360 Web service API documentation.

6. After updating PartnerConfig.py to suit your requirements, you must test the code by running Runner.py and ensuring that all required configuration is created in MaaS360 portal.

7. Create a trial account in MaaS360 to test your integration.

8. To test the code, perform the following:

a) Open Runner.py in a text editor of your choice and set the values for following parameters:

1) WS\_SERVER\_BASE: This should point to the base web service URL for your MaaS360 portal e.g. https://services.m3.maas360.com

2) BILLING\_ID: This is the billing ID for your organization's account in MaaS360.

3) USERNAME: The username for your MaaS360 account that you use for web service access.

4) PASSWORD: The password for your MaaS360 account.

b) After editing the file with correct values, save it.

c) Run the script on command line by typing the following:

python Runner.py

d) Confirm that the script runs successfully without any errors.