# TA (Things Adaptor) Plugin implementation

Depending on the needs of each BETaaS configuration, a plugin for TA can be implemented for coping with the different protocol of underlying devices. Producing a specific plugin for the TA module is essentially the implementation of two (2) major interfaces residing at the betaas-adaptation-plugin module:

1. *IAdaptorPlugin* and
2. *IAdaptorListener*

The Interface *IAdaptorPlugin* is responsible for initiating and managing the communication of the TA module with the devices managed by BETaaS. It includes the following methods:

1. void **setListener**(IAdaptorListener listener).

The method imposes the implementing class as the listener for registering and unregistering itself to and from the Things.

1. Vector<HashMap<String, String>> **discover**().

This method instigates the connectivity with the devices of the underlying cloud of Things and should implement the particular protocols discovery mechanisms. After that call is completed the system shall at least have gathered the IDs of the devices (Thing IDs) that are essential to the further functioning of BETaaS. This method should be run on a regular basis depending on the needs of the developer if for example new devices are to be connected with the system after it’s startup.

1. boolean **register**(String sensorID\*)

This method is used to register a specific device using its ID as described by the device protocol or developer’s needs. Once the Thing is registered there will be notifications published on behalf of it by the plugin (see method *notify* of *IAdaptorListener*)

1. boolean **unregister**(String sensorID)

Once a device is no longer available or needed it can be removed and the implementation of this method should provide the details of such operation.

1. String **getData**(String sensorID)

This method should implement how the TA will retrieve the value of a particular device based solely on its ID.

1. public String **setData**(String sensorID, String value)

This method should provoke the change of value for, typically, an actuator identified by the sensorID and its new value.

The second implemented class IAdaptorListener is responsible for publishing notifications for “registered” things. This is made with the following methods:

1. boolean **notify**(String type, String resourceID, HashMap<String, String> value).

This method should be called every time there is a change in the value of a sensor, registered by the system (i.e. the above *register* method is used for it).

1. boolean **removeThing**(String thingId).

A thing is removed by the mechanism implemented by the method above in case of error or when the *unregister* method is called.

\**Please not that terms deviceID and ThingID are used interchangeably but essentially refer to the same aspect, that of a discernible and particular sensor/actuator identification number.*

After the implementation is completed the code must be bundled (<packaging>bundle</packaging> in the pom file) and all the api of the TAplugin must be imported to it by seting the following inside the project’s pom for bundle configuration:

<instructions>

<Bundle-SymbolicName>${project.artifactId}</Bundle-SymbolicName>

<Bundle-Version>${project.version}</Bundle-Version>

<Import-Package>eu.betaas.adaptation.plugin.api, \*</Import-Package>

</instructions>

Following the successful build of the bundle we perform the following actions:

1. Remove the BETaaS Adaptation Simulator

karaf

uninstall <ID of the bundle>

1. Deploy the TA adapter inside karaf (assuming the MVN name is “betaas-adaptation-implemented”)

osgi:install mvn:eu.betaas/betaas-adaptation-implemented/0.0.1-SNAPSHOT