Methods Description

The main class of the project is the class **OpenhabManager**.

This class implements the interface OpenhabInterface and overrides these methods:

- *startPolling*: this method starts the execution of the polling service, it creates a PollingThread object (if there isn't another in execution);
- *stopPolling*: this method stops the execution of the polling service;
- *getJson*: this method returns the entire Json file, with all items and states, updated by the polling service (if the polling service is not active, it returns a log WARN);
- *getList*: it returns a list of Item objects, this list is created by the JsonArray memorized (if the polling service is not active, it returns a log WARN);
- *getHashmap*: it returns an HashMap, with the item name as key and the item state as value, (if the polling service is not active, it returns a log WARN);
- *getStatusFromJson*: this method returns the state of the item specified, it returns the state memorized in the JsonArray (if the polling service is not active, it returns a log WARN);
- *getStatus*: this method allows to get the item state through a Rest call to the OpenHab system;
- *setStatus*: this method allows to modify the status of a specific item, this method executes a PUT rest call to Openhab;
- *sendCommand*: this method executes a POST rest call on Openhab, it allows to send a command to a specific Openhab item. This method cannot be invoked for some kinds of items, for example for Contact items.

The OpenhabManager class has been created with the Singleton pattern. To create an OpenhabManager object there are two methods:

- *createOpenhabManager(secondsloop: int):* this static method can be invoked to create an istance of this class, if there isn't another OpenhabManager object, it will be created, on the contrary it will return the reference of this object. With this method the user can specify the frequency of the polling service.
- *createOpenhabManager*: this method has the same functions of the previous, but in this case the user don't specify the frequency of the polling service, but it is set by default to 1 second.

In the package openhabBroker there is also the class **PollingThread**, used by the previous class to execute the polling service.

This class implements the interface Runnable, with whom is created a Thread to execute a GET rest call on all Openhab items and memorize them in the JsonArray data of the OpenhabManager object.

This project is composed also by the package item.openhabBroker. It contains the class **Item**. Each object Item is characterized by a name and a state, accessible by the methods getItemName, getItemValue, setItemValue and setItemValue.