Python API documentation

Jump to bottom

Roberto Riggio edited this page on Aug 18, 2019 · 1 revision

Table of Contents

Network Apps

A Network Apps are essentially Python modules running on top of the EmPOWER Runtime. Network Apps can be loaded either at bootstrap or using the Controller REST interface without affecting the network operations. A command line utility for loading Network Apps is also available.

Writing a Hello World App

An EmPOWER App typically consists of a single Python class and of a *launch* method. The listing below shows a simple EmPOWER App which periodically prints the "Hello! World." message.

```
from empower.core.app import EmpowerApp

class HelloWorld(EmpowerApp):
    def loop(self):
        print("Hello! World.")

def launch(tenant_id, every=5000):
    return HelloWorld(tenant_id=tenant_id, every=every)
```

In the example above the class has a single *loop* method which is periodically called by the EmPOWER Runtime.

The *launch* method must return an instance of the *EmpowerApp* class and must at least specify the *tenant_id* parameter. This parameter must be passed by name to the Network App class. An *every* parameter can also be used in order to specify how often (in ms) the *loop* method must be called.

Create a folder named *helloworld2* under the *empower-runtime/apps* directory. Inside create two files: `__init__.py` and `helloworld2.py` . Leave the first file empty and copy the previous script inside the second file.

You can now start the controller with your Hello World App using:

```
./empower-runtime.py apps.helloworld2.helloworld2 --tenant_id=<your tenant id>
```

The output should be something like this:

```
INFO:core:Starting EmPOWER Runtime
INFO:core:Generating default accounts
INFO:core:Loading EmPOWER Runtime defaults
INFO:root:Importing module: empower.restserver.restserver
INFO:root:Importing module: empower.lvnfp.lvnfpserver
INFO:root:Importing module: empower.lvapp.lvappserver
INFO:root:Importing module: empower.vbsp.vbspserver
INFO:root:Importing module: empower.lvapp.lvap_stats.lvap_stats
INFO:root:Importing module: empower.lvapp.bin_counter.bin_counter
INFO:root:Importing module: empower.lvapp.wtp_bin_counter.wtp_bin_counter
INFO:root:Importing module: empower.lvapp.txp_bin_counter.txp_bin_counter
INFO:root:Importing module: empower.lvapp.trq_bin_counter.trq_bin_counter
INFO:root:Importing module: empower.lvapp.cqm.ucqm
INFO:root:Importing module: empower.lvapp.cqm.ncqm
INFO:root:Importing module: empower.lvapp.wifi_stats.wifi_stats
INFO:root:Importing module: empower.lvapp.rssi.rssi
INFO:root:Importing module: empower.lvapp.summary.summary
INFO:root:Importing module: empower.vbsp.rrc_measurements.rrc_measurements
INFO:root:Importing module: empower.vbsp.mac_reports.mac_reports
INFO:root:Importing module: empower.lvnfp.lvnf_get.lvnf_get
INFO:root:Importing module: empower.lvnfp.lvnf_set.lvnf_set
INFO:root:Importing module: empower.lvnfp.lvnf_stats.lvnf_stats
INFO:root:Importing module: empower.apps.helloworld.helloworld2
INFO:core:Registering 'empower.restserver.restserver'
INFO:core.service:REST Server available at 8888
INFO:core:Registering 'empower.lvnfp.lvnfpserver'
INFO:core.service:LVNF Server available at 4422
INFO:core:Registering 'empower.lvapp.lvappserver'
INFO:core.service:LVAP Server available at 4433
INFO:core:Registering 'empower.vbsp.vbspserver'
INFO:core.service:VBSP Server available at 2210
INFO:core:Registering 'empower.ibnp.ibnpserver'
INFO:core:Registering 'empower.lvapp.lvap_stats.lvap_stats'
INFO:core:Registering 'empower.lvapp.bin_counter.bin_counter'
INFO:core:Registering 'empower.lvapp.wtp_bin_counter.wtp_bin_counter'
```

```
INFO:core:Registering 'empower.lvapp.txp_bin_counter.txp_bin_counter'
INFO:core:Registering 'empower.lvapp.trq_bin_counter.trq_bin_counter'
INFO:core:Registering 'empower.lvapp.cqm.ucqm'
INFO:core:Registering 'empower.lvapp.cqm.ncqm'
INFO:core:Registering 'empower.lvapp.wifi_stats.wifi_stats'
INFO:core:Registering 'empower.lvapp.rssi.rssi'
INFO:core:Registering 'empower.lvapp.summary.summary'
INFO:core:Registering 'empower.vbsp.rrc_measurements.rrc_measurements'
INFO:core:Registering 'empower.vbsp.mac_reports.mac_reports'
INFO:core:Registering 'empower.lvnfp.lvnf_get.lvnf_get'
INFO:core:Registering 'empower.lvnfp.lvnf_set.lvnf_set'
INFO:core:Registering 'empower.lvnfp.lvnf_stats.lvnf_stats'
INFO:core:Registering 'empower.apps.helloworld.helloworld2'
INFO:core.app:Setting control loop interval to 5000ms
Hello! World.
Hello! World.
Hello! World.
```

Modifying the Hello World App

We will now extend the Hello World App in order add basic network monitoring features. In particular we want to print to screen the list of neighbours of all the WTPs in our network.

Notice that this section assumes that you are using a setup with at least one WTP like the one depicted here.

However, before we can poll a WTP for the list of its neighbours, we must know its address. This can be achieved in several ways. Here we will do it by leveraging on the *events* subsystems.

Events allow the network programmer to react to certain situations. For example, using the *wtp_up* primitive it is possible to trigger a callback when a new WTP connects to the controller.

In order to register a *wtp_up* you need to use the following code:

```
def wtp_up(self, wtp):
    for block in wtp.supports:
        self.ucqm(block=block,
                  every=self.every,
                  callback=self.ucqm_callback)
```

Finally, you need to implement the *self.ucqm_callback* callback:

```
def ucqm_callback(self, poller):
    for entry in poller.maps.values():
        self.log.info("Address: %s RSSI %.2f", entry['addr'], entry['mov_rssi'])
```

The output should be something like this:

```
...
INFO:core.app:Address: B0:C5:54:02:50:06 RSSI -68.00
INFO:core.app:Address: 08:21:EF:A3:23:AC RSSI -92.00
INFO:core.app:Address: 00:36:76:68:98:67 RSSI -83.00
INFO:core.app:Address: A0:4C:5B:72:59:AB RSSI -92.00
INFO:core.app:Address: 8C:57:9B:DE:2D:5A RSSI -91.00
...
```

The complete documentation of the EmPOWER Wi-Fi API can be found here.

Launching the Hello World App

Network Apps can also be loaded from the Runtime REST interface, for example using the *curl* command:

```
curl -X POST -d "{ \"version\":\"1.0\", \"argv\" : \"apps.helloworld2.helloworld2\" }" htt
```

Using the Network Apps REST interface

Each active Network App registers a REST URL. For example if you open the following URL using the *curl* command:

```
curl http://127.0.0.1:8888/api/v1/tenants/52313ecb-9d00-4b7d-b873-b55d3d9ada26/components/
```

you will get an output like this

```
{
    "app_name": "helloworld2",
    "every": 5000,
    "params": [
        "every"
    ],
    "tenant_id": "52313ecb-9d00-4b7d-b873-b55d3d9ada26",
    "ui_url": "/apps/tenants/52313ecb-9d00-4b7d-b873-b55d3d9ada26/helloworld2/"
}
```

Apps Parameters

Network apps can have additional configuration parameters. For example let's assume that we want to personalise the hello message. In order to do so we need to modify the Hello World app in the following way:

```
from empower.core.app import EmpowerApp

class HelloWorld(EmpowerApp):

    def loop(self):
        """Periodic job."""

        print("Hello! %s." % self.message)

    @property
    def message(self):
        """Return message."""

        return self.__message

    @message.setter
    def message(self, value):
        """Set message."""

        self.__message = value


def launch(tenant_id, message="World", every=5000):
    return HelloWorld(tenant_id=tenant_id, message=message, every=every)
```

Notice how we added a parameter called *message* to the app implementation. The default value of this parameter is "World". However we can change the message at run time by launching the application in the following way:

```
./empower-runtime.py apps.helloworld2.helloworld2 --tenant_id=<your tenant id> --message=E
```

This should produce an output like this:

```
INFO:core:Starting EmPOWER Runtime
INFO:core:Generating default accounts
INFO:core:Loading EmPOWER Runtime defaults
INFO:root:Importing module: empower.restserver.restserver
INFO:root:Importing module: empower.lvnfp.lvnfpserver
INFO:root:Importing module: empower.lvapp.lvappserver
INFO:root:Importing module: empower.vbsp.vbspserver
INFO:root:Importing module: empower.lvapp.lvap_stats.lvap_stats
INFO:root:Importing module: empower.lvapp.bin_counter.bin_counter
INFO:root:Importing module: empower.lvapp.wtp_bin_counter.wtp_bin_counter
INFO:root:Importing module: empower.lvapp.txp_bin_counter.txp_bin_counter
INFO:root:Importing module: empower.lvapp.trq_bin_counter.trq_bin_counter
INFO:root:Importing module: empower.lvapp.cqm.ucqm
INFO:root:Importing module: empower.lvapp.cqm.ncqm
INFO:root:Importing module: empower.lvapp.wifi_stats.wifi_stats
INFO:root:Importing module: empower.lvapp.rssi.rssi
```

```
INFO:root:Importing module: empower.lvapp.summary.summary
INFO:root:Importing module: empower.vbsp.rrc_measurements.rrc_measurements
INFO:root:Importing module: empower.vbsp.mac_reports.mac_reports
INFO:root:Importing module: empower.lvnfp.lvnf_get.lvnf_get
INFO:root:Importing module: empower.lvnfp.lvnf_set.lvnf_set
INFO:root:Importing module: empower.lvnfp.lvnf_stats.lvnf_stats
INFO:root:Importing module: empower.apps.helloworld.helloworld2
INFO:core:Registering 'empower.restserver.restserver'
INFO:core.service:REST Server available at 8888
INFO:core:Registering 'empower.lvnfp.lvnfpserver'
INFO:core.service:LVNF Server available at 4422
INFO:core:Registering 'empower.lvapp.lvappserver'
INFO:core.service:LVAP Server available at 4433
INFO:core:Registering 'empower.vbsp.vbspserver'
INFO:core.service:VBSP Server available at 2210
INFO:core:Registering 'empower.ibnp.ibnpserver'
INFO:core:Registering 'empower.lvapp.lvap_stats.lvap_stats'
INFO:core:Registering 'empower.lvapp.bin_counter.bin_counter'
INFO:core:Registering 'empower.lvapp.wtp_bin_counter.wtp_bin_counter'
INFO:core:Registering 'empower.lvapp.txp_bin_counter.txp_bin_counter'
INFO:core:Registering 'empower.lvapp.trq_bin_counter.trq_bin_counter'
INFO:core:Registering 'empower.lvapp.cqm.ucqm'
INFO:core:Registering 'empower.lvapp.cqm.ncqm'
INFO:core:Registering 'empower.lvapp.wifi_stats.wifi_stats'
INFO:core:Registering 'empower.lvapp.rssi.rssi'
INFO:core:Registering 'empower.lvapp.summary.summary'
INFO:core:Registering 'empower.vbsp.rrc_measurements.rrc_measurements'
INFO:core:Registering 'empower.vbsp.mac_reports.mac_reports'
INFO:core:Registering 'empower.lvnfp.lvnf_get.lvnf_get'
INFO:core:Registering 'empower.lvnfp.lvnf_set.lvnf_set'
INFO:core:Registering 'empower.lvnfp.lvnf_stats.lvnf_stats'
INFO:core:Registering 'empower.apps.helloworld.helloworld2'
INFO:core.app:Setting control loop interval to 5000ms
Hello! Bob.
Hello! Bob.
Hello! Bob.
```

Apps parameters can be modified using the REST interface. For example in order to specify a new value for the *message* parameter you can use the following *curl* call:

```
curl -X PUT -d "{ \"version\":\"1.0\", \"params\" : { \"message\": \"Alice\" } }" http://
```

▸ **Pages** 25

Getting Started

- Introduction
- Terminology

Clone this wiki locally

```
https://github.com/clicknf/clicknf.github.io.wiki.git
```