pynio Documentation

Release 0.1

n.io

March 05, 2015

Contents

pynio is a Python SDK for interacting with a running n.io instance.

Contents:

CHAPTER 1

Instance

class pynio.instance.**Instance**(*host='127.0.0.1'*, *port=8181*, *creds=None*)

Interface for a running n.io instance.

This is the main part of pynio. All communication with the n.io instance goes through this object. Blocks and Services contain a reference to an Instance in order to communication with the running instance.

Parameters

- **host** (*str, optional*) – Host ip address of running n.io instance. Default is '127.0.0.1'.
- **port** (*int, optional*) – Port of runing n.io instance. Default is 8181.
- **creds** (*(str, str), optional*) – Username and password for basic authentication. Default is ('Admin', 'Admin').

blocks

dict of Block

A collection of block names with their Block instance.

services

dict of Service

A collection of service names with their Service instance.

DELETE_ALL()

Deletes all blocks and services from an instance.

add_block(*block*, *overwrite=False*)

Add block to instance.

Parameters

- **block** (*Block*) – Block instanace to add.
- **overwrite** (*bool, optional*) – If True, configuration will be updated if *block* is already in Instance. Default is False.

Raises ValueError – If *overwrite* is False and *block* is already in the instance.

add_service(*service*, *overwrite=False*, *blocks=False*)

Add service to instance.

Parameters

- **service** (*Service*) – Service instanace to add.
- **overwrite** (*bool, optional*) – If True, configuration will be updated if *service* is already in Instance. Default is False.

Raises `ValueError` – If *overwrite* is False and *block* is already in the instance.

create_block(*name*, *type*, *config=None*)
Create a block in the instance.

Parameters

- **name** (*str*) – Name of new block.
- **type** (*str*) – BlockType of new block.
- **config** (*dict, optional*) – Optional configuration of block properties.

create_service(*name*, *type=None*, *config=None*)
Create a service in the instance.

Parameters

- **name** (*str*) – Name of new service.
- **type** (*str, optional*) – ServiceType of new service.
- **config** (*dict, optional*) – Optional configuration of service.

nio()
Returns nio version info.

save()
Save all blocks and all services in instance.

CHAPTER 2

Service

class pynio.service.**Service**(*name*, *type='Service'*, *config=None*, *instance=None*)

n.io Service

Parameters

- **name** (*str*) – Name of new service.
- **type** (*str, optional*) – ServiceType of new service.
- **config** (*dict, optional*) – Optional configuration of service.
- **instance** (*Instance, optional*) – Optional instance to add the service to.

name

str

Name of service.

type

str

ServiceType of service.

config

dict

Configuration of service.

status

str

Status of service.

blocks

Return a list of blocks used by this service.

command(*command*, *block=None*, ***request_kwargs*)

Send a command to the service or to a block in the service.

Parameters

- **command** (*str*) – The name of the command.
- **block** (*str, optional*) – The name of the block if commanding a block.
- **request_kwargs** – Keyword arguments are passed to http request. Examples: data, timeout (set the request timeout).

connect(*blk1*, *blk2=None*)
Connect two blocks.

Parameters

- **blk1** (*Block*) – Sends signals to *blk2*. If *blk2* is not specified then *blk1* is added to the service with no receivers.
- **blk2** (*Block, optional*) – Receives signal form *blk1*.

create_block(*name*, *type*, *config=None*)
Create a new block and add it to the service.

Parameters

- **name** (*str*) – Name of new block.
- **type** (*str, optional*) – BlockType of new block.
- **config** (*dict, optional*) – Optional configuration of block.

delete()
Delete the service from the instance

remove_block(*block*)
Remove a block from service. Does NOT delete the block.

Parameters **block** (*Block*) – Block to remove from service.

save()
PUTs the service config to nio.

Will create a new service if one does not exist by this name. Otherwise it will update the existing service config.

Raises `Exception` – If service is not associated with an instance.

start()
Starts the nio Service.

stop()
Stops the nio Service.

CHAPTER 3

Block

class pynio.block.**Block**(*name*, *type*, *config=None*, *instance=None*)

n.io Service

Blocks are units inside of services. They perform functions on incomming signals and put the signals out.

Parameters

- **name** (*str*) – Name of new block.
- **type** (*str, optional*) – BlockType of new block.
- **config** (*dict, optional*) – Optional configuration of block.
- **instance** (*Instance, optional*) – Optional instance to add the block to.

name

str

Name of service.

type

str

ServiceType of service.

config

dict

Configuration of service.

status

str

Status of service.

delete()

Delete the block from the instance

in_use()

Return a list of services that use this block.

Raises TypeError – If block is not associated with an instance.

save()

PUTs the block config to nio.

Will create a new block if one does not exist by this name. Otherwise it will update the existing block config.

Raises `Exception` – If service is not associated with an instance.

CHAPTER 4

Indices and tables

- *genindex*
- *modindex*
- *search*

p