
aioazure

Release 0.1.0

Aug 22, 2019

Contents:

1	aioazure	1
1.1	aioazure package	1
2	aioazure	3
2.1	Installation	3
2.2	How to use aioazure	3
2.3	Currently supported Azure services, operation groups and operations	4
2.4	Adding further Azure services, operation groups and operations	4
3	Indices and tables	5
	Python Module Index	7
	Index	9

1.1 aioazure package

1.1.1 Subpackages

aioazure.interfaces package

```
class aioazure.interfaces.Proxy
    Bases: object

class aioazure.interfaces.ProxyDecorator (proxy: aioazure.interfaces._proxy.Proxy)
    Bases: aioazure.interfaces._proxy.Proxy
```

aioazure.resources package

1.1.2 Submodules

aioazure.auth module

```
class aioazure.auth.Authenticator (app_id: str, password: str, tenant_id: str)
    Bases: object

    get_token() → str

    is_token_valid
```

aioazure.client module

```
class aioazure.client.AzureClient (api_url: str, subscription_id: str, resource_group_name: str, auth: aioazure.auth.Authenticator, timeout: int = 60)
    Bases: object
```

aioazure.decorator module

```
class aioazure.decorator.AsyncOperationDecorator (proxy: Proxy)
    Bases: aioazure.interfaces._decorator.ProxyDecorator

    async_polling (url, retry_after)

class aioazure.decorator.AuthDecorator (proxy: Proxy, auth: aioazure.auth.Authenticator)
    Bases: aioazure.interfaces._decorator.ProxyDecorator

class aioazure.decorator.PagingDecorator (proxy: Proxy)
    Bases: aioazure.interfaces._decorator.ProxyDecorator

    static get_next_page (next_link: str)

    get_pages (response)

class aioazure.decorator.ResponseDecorator (proxy: Proxy)
    Bases: aioazure.interfaces._decorator.ProxyDecorator
```

aioazure.proxy module

```
class aioazure.proxy.ApiProxy (api: simple_rest_client.api.API, auth: aioazure.auth.Authenticator)
    Bases: aioazure.interfaces._proxy.Proxy

    add_resource (api_root_url=None, resource_name=None, resource_class=None, params=None, headers=None, timeout=None, append_slash=False, json_encode_body=False)

class aioazure.proxy.ResourceProxy (resource: simple_rest_client.resource.AsyncResource)
    Bases: aioazure.interfaces._proxy.Proxy

    pass_through_parameters = ['api_root_url', 'resource_name', 'params', 'headers', 'time
```

CHAPTER 2

aioazure

aioazure is a simplistic python REST client for the Microsoft Azure REST API utilizing asyncio. The client itself has been developed against the [Microsoft Azure REST API documentation](#).

2.1 Installation

The goal is to distribute this package via [PyPi](#), so a simple

```
pip install aioazure
```

would be needed to install the package. The release will happen once the client has been tested with Azure.

2.2 How to use aioazure

```
from aioazure.auth import Authenticator
from aioazure.client import AzureClient

auth = Authenticator(app_id="your_app_id",
                     password="your_secret",
                     tenant_id="your_tenant_id")

client = AzureClient(api_url="url_of_the_azure_api",
                    subscription_id="your_subscription_id",
                    resource_group_name="your_resource_group_name",
                    auth=auth,
                    timeout=60) # <- this is optional

await client.compute.virtualmachines.create_or_update("my-vm-name",
                                                       location="Antarctica")
```

(continues on next page)

(continued from previous page)

```
await client.compute.virtualmachines.instance_view("my-vm-name")

await client.compute.virtualmachines.power_off("my-vm-name")

await client.compute.virtualmachines.delete("my-vm-name")
```

2.3 Currently supported Azure services, operation groups and operations

- **Azure Compute**
 - **Virtual Machines**
 - * `create_or_update`
 - * `instance_view`
 - * `power_off`
 - * `delete`

2.4 Adding further Azure services, operation groups and operations

Each Azure service (compute, storage services, etc.) is represented by a `yaml` file in the `models` directory. This `yaml` file contains mapping nodes for each operation group (virtualmachines, manageddisks, etc.). Each operation group consists of two mapping nodes, the version of api to use (`api_version`) and the supported operations (`actions`) in this operation group.

```
rest_operation_group:
  api_version: "2018-06-01"
  actions:
    action_1:
      method: GET
      url: Microsoft.Compute/rest_operations_group/{}
    ...
    action_n:
      method: POST
      url: Microsoft.Compute/rest_operations_group/{}
rest_operation_group_2:
  ...
```

The Azure service, operation groups and operation can then be called in Python as described below.

```
await client.<service_name>.<rest_operation_group>.<action>(args, kwargs)
```

In case you add additional services, operation groups and operations, please submit a pull request so that others can profit as well from the work you have done. Thank you!

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`

a

- `aioazure`, 1
- `aioazure.auth`, 1
- `aioazure.client`, 1
- `aioazure.decorator`, 2
- `aioazure.interfaces`, 1
- `aioazure.proxy`, 2
- `aioazure.resources`, 1

A

`add_resource()` (*aioazure.proxy.ApiProxy* method), 2

`aioazure` (module), 1

`aioazure.auth` (module), 1

`aioazure.client` (module), 1

`aioazure.decorator` (module), 2

`aioazure.interfaces` (module), 1

`aioazure.proxy` (module), 2

`aioazure.resources` (module), 1

`ApiProxy` (class in *aioazure.proxy*), 2

`async_polling()` (*aioazure.decorator.AsyncOperationDecorator* method), 2

`AsyncOperationDecorator` (class in *aioazure.decorator*), 2

`AuthDecorator` (class in *aioazure.decorator*), 2

`Authenticator` (class in *aioazure.auth*), 1

`AzureClient` (class in *aioazure.client*), 1

G

`get_next_page()` (*aioazure.decorator.PagingDecorator* static method), 2

`get_pages()` (*aioazure.decorator.PagingDecorator* method), 2

`get_token()` (*aioazure.auth.Authenticator* method), 1

I

`is_token_valid` (*aioazure.auth.Authenticator* attribute), 1

P

`PagingDecorator` (class in *aioazure.decorator*), 2

`pass_through_parameters` (*aioazure.proxy.ResourceProxy* attribute), 2

`Proxy` (class in *aioazure.interfaces*), 1

`ProxyDecorator` (class in *aioazure.interfaces*), 1

R

`ResourceProxy` (class in *aioazure.proxy*), 2