

Client Python



Cette page n'est plus actualisée. À partir de BlueMind 4.8, veuillez consulter la [nouvelle documentation BlueMind](#)

Version de python

Le client python supporte python 2.7 ou la version courante de python 3. Nous recommandons l'utilisation de python 3 pour tous les nouveaux projets, python 2 étant en fin de vie.

Installation

Le client python peut s'installer de deux manières différentes, soit en installant le paquet *python-bm-client* disponible dans les dépôts BlueMind (python 2.7 seulement), soit en l'installant via pip (python 2.7 ou python 3).

Nous recommandons l'installation via pip, si possible dans un [environnement virtuel](#) (virtualenv ou venv) de façon à ne pas interférer avec le système. La mise en place d'un tel environnement dépassant le cadre de cette documentation, les instructions suivantes se limitent à l'utilisation de la distribution python du système.

```
Debian/Ubuntu CentOS/Redhat Mac OSX
aptitude install python-pip
pip install --upgrade setuptools
pip install netbluemind4
```

```
yum install python-pip
pip install --upgrade setuptools
pip install netbluemind4
```

```
curl -O https://bootstrap.pypa.io/get-pip.py
sudo /usr/bin/python get-pip.py
sudo /usr/local/bin/pip install --upgrade pip
pip install --upgrade setuptools
sudo /usr/local/bin/pip install netbluemind4
```

Utilisation

Authentification

Le client s'initialise avec la classe `BMClient` :

```
from netbluemind.python.client import BMClient

# BEGIN CONF
URL="http://localhost:8090/api"
# END CONF

f = open('/etc/bm/bm-core.tok', 'r')
KEY = f.readline()
f.close()

client = BMClient(URL, KEY)
```

URL définit l'url d'accès au serveur BlueMind, depuis l'extérieur l'url sera *https://bluemind.domain.net/api*.

KEY correspond au mot de passe ou token de l'admin0, dans cet exemple on utilise le token admin0 disponible sur le serveur afin d'avoir des droits étendus au niveau de l'API, il est possible de s'authentifier en tant qu'utilisateur avec :

```
from netbluemind.python.client import BMClient

# BEGIN CONF
URL="http://localhost:8090/api"
# END CONF

client = BMClient(URL)
client.login('login@domain.net', 'password')
```

A noter qu'il est aussi possible d'utiliser la clé d'API d'un utilisateur à la place de son mot de passe.

Utilisation

BMClient

La classe BMClient est un helper qui permet aussi d'accéder aux différentes classes pour manipuler les utilisateurs, groupes, etc

```
from netbluemind.python.client import BMClient

# BEGIN CONF
URL="http://localhost:8090/api"
# END CONF

client = BMClient(URL)
client.login("login@domain.net", "password")
print client.apiKey

iUser = client.users("domain.net")
user = iUser.byEmail("login@domain.net")
print user.value.login
```

Users

iUser (instance de IUser) permet d'accéder et modifier toutes les informations sur un utilisateur (mot de passe, photo, contact, statut archivé, etc), les différentes méthodes retournent une instance d'un objet *ItemValue* qui permet d'accéder à l'object en question via l'attribut *value*.

Domains

Il est possible de créer ou modifier des domaines en utilisant l'interface *IDomains*.

Lors de la création d'un domain, il est important que l'uid du domain soit égale à son nom, la création se fait donc ainsi :

```
domain = Domain()
domain.name = "local.lan"
domain.label = "local.lan"
domain.description = "Domain of " + "local.lan"
domain.properties = {}
# global doit toujours être à False
domain.global_ = False
# permet de définir les alias du domaine
domain.aliases = []

idomains.create("local.lan", domain)
```

Il est possible de définir la configuration des paramètres pour les plugins *ImportLDAP* et *ImportAD* en utilisant l'attribut *properties* :

```

properties = dict()
properties['import_ldap_hostname'] = 'ldap.local'
properties['import_ldap_password'] = 'ldap-password'
properties['import_ldap_user_filter'] = '(mail=*)'
properties['import_ldap_accept_certificate'] = True
properties['import_ldap_ext_id_attribute'] = 'entryUUID'
properties['import_ldap_relay_mailbox_group'] = ''
properties['import_ldap_enabled'] = True
properties['import_ldap_group_filter'] = '(objectClass=posixGroup)'
properties['import_ldap_protocol'] = 'tls'
properties['import_ldap_base_dn'] = 'dc=lan,dc=local'
properties['import_ldap_login_dn'] = 'uid=admin,cn=users,cn=accounts,dc=lan,dc=local'
properties['lang'] = 'fr'
properties['im_public_auth'] = True # allow Instant messaging for all users
properties['domain_max_users'] = None
properties['mail_routing_relay'] = ''
properties['mailbox_max_user_quota'] = '0'
properties['mailbox_default_user_quota'] = '0'
properties['mailbox_max_publicfolder_quota'] = '0'
properties['mailbox_default_publicfolder_quota'] = '0'
properties['filehosting_max_filesize'] = '0'
properties['filehosting_retention'] = '365'
properties['mail_autoDetachmentLimit'] = '0'
properties['mail_forward_unknown_to_relay'] = False

```

Pour l'import Active Directory il suffit de remplacer *ldap* par *ad*.

Exemples

De nombreux exemples de possibilités sont accessibles ici : <https://forge.bluemind.net/stash/projects/BA/repos/bluemind-samples/browse/python-api-examples>

Démarrer un backup

```

client = BMCClient(URL, KEY)
idataprotect = client.instance(IDataProtect)

taskRef = idataprotect.saveAll()
status = client.waitTask(taskRef)
print "\nBackup done with status [" + status.state.value + "]"

```