
Client Test PyQt5 Documentation

Version 0.0.1

Stefano

25 March 2015

1	Module principale	3
1.1	Proxy	3
2	Constantes	5
3	Log	7
4	Indices and tables	9
	Index des modules Python	11
	Index	13

Contents :

Module principale

```
class main.FormArticle (parent=None, aSignal=None, aLibelle='', aPrix=0, aDate=<built-in method
                                today of type object at 0x97f780>)
    fenêtre d'édition d'un article utilisé pour ajouter/modifier un article
    AjouterArticle ()
    Annuler ()

class main.MainWindow (parent=None, aConnection=None)
    fenêtre principale permet de lister les articles CRUD Articles
    AfficherFormulaire (aLibelle='', aPrix=0, aDate=datetime.date(2015, 3, 25))
    AjouterArticle ()
    CommitSession ()
    ModifierArticle ()
    RemplirArticles ()
    RollbackSession ()
    SupprimerArticle ()
    afficherArticles ()
    changerProxy ()
    fermerAppli ()
    imprimer ()
    slot_FormArticle_closed (alibelle, aprix, adate)
```

1.1 Proxy

Ce module sert d'interface au serveur. Toutes les méthodes d'accès aux modèles dans la base de données sont décrites ci-après. Created on 18 mars 2015

@author : stefano

Module de connexion au serveur par type de proxy

```
class proxy.Proxy (aConnection=None, aApplication=None)
    classe d'interface Proxy
    afficherMainWindow ()
        Demande l'autorisation de l'affichage MainWindow
        Type retourné boolean
    ajouterArticle (alibelle, aprix, adate)
        Ajoute les champs d'un article
```

Paramètres

- aid** (*int*) – id de l'article à ajouter
- alibelle** (*str*) – libellé de l'article à ajouter
- aprix** (*str*) – prix de l'article à ajouter
- adate** (*str*) – date de l'article à ajouter

listArticles ()

Demande la liste des articles

Retourne liste des articles

Type retourné dict

Voir aussi :

modifierArticle(), supprimerArticle(), ajouterArticle()

Avertissement : retourne tous les articles au format JSON pour serveurweb

Note : serveurweb retourne le dictionnaire des articles sérialisé au format Django

À faire

A implémenter dans les classe dérivant de Proxy

modifierArticle (*aid, alibelle, aprix, adate*)

Modifie les champs d'un article

Paramètres

- aid** (*int*) – id de l'article à modifier
- alibelle** (*str*) – libellé de l'article à modifier
- aprix** (*str*) – prix de l'article à modifier
- adate** (*str*) – date de l'article à modifier

supprimerArticle (*aid*)

Supprime les champs d'un article

Paramètres **aid** (*int*) – id de l'article à supprimer

class proxy.**ProxyREST** (*aConnection=None*)

Classe Proxy pour serveurREST

class proxy.**ProxyWeb** (*aConnection=None*)

Classe Proxy pour serveurweb

class proxy.**ProxyXMLRPC** (*aConnection=None*)

Classe Proxy XMLRPC

Constantes

Cette page représente la documentation des constantes.

`constantes.ADR_REST = '/serveurREST/'`
URL du serveur REST

`constantes.ADR_WEB = '/serveurweb/'`
URL du serveur web - serveur principale

`constantes.ADR_XMLRPC = '/serveurXMLRPC/'`
URL du serveur XML-RPC

`constantes.DEBUG = True`
Développement : DEBUG = True Production : DEBUG = False

`constantes.MODE_ADD = 'add'`
MODE Ajout d'un objet

`constantes.MODE_MOD = 'mod'`
MODE Modification d'un objet

`constantes.PROXY_PWD = 'guest'`
Mot de passe

`constantes.PROXY_USER = 'guest'`
Utilisateur

Log

Created on 18 mars 2015

@author : stefano

Module de configuration des logs

Example

```
logger = logging.getLogger(__name__)
```

```
logger.debug('trace de debug')
```

```
class log.RequireDebugFalse (name='')
    classe de filtre pour DEBUG = False
```

```
class log.RequireDebugTrue (name='')
    classe de filtre pour DEBUG = True
```

```
log.configure()
    Configure les log avec dict_log
```

dictionnaire de configuration des logs *dict_log* :

```
dict_log = {
    'version': 1,
    'disable_existing_loggers': False,
    'filters': {
        'require_debug_false': {
            '()': 'log.RequireDebugFalse',
        },
        'require_debug_true': {
            '()': 'log.RequireDebugTrue',
        },
    },
    'formatters': {
        'simple': {
            'format': '[%(asctime)s] %(levelname)s %(message)s',
            'datefmt': '%Y-%m-%d %H:%M:%S'
        },
        'verbose': {
            'format': '[%(asctime)s] %(levelname)s [%(name)s.%(funcName)s:%(lineno)d] %(message)s',
            'datefmt': '%Y-%m-%d %H:%M:%S'
        },
    },
    'handlers': {
        'null': {
```

```
        'class': 'logging.NullHandler',
    },
    'console': {
        'level': 'DEBUG',
        'filters': ['require_debug_true'],
        'class': 'logging.StreamHandler',
        'formatter': 'simple'
    },
    'development_logfile': {
        'level': 'DEBUG',
        'filters': ['require_debug_true'],
        'class': 'logging.handlers.RotatingFileHandler',
        'filename': '/tmp/clientPyQt_dev.log',
        'maxBytes': 1024*1024*5, # 5 MB
        'backupCount' : 7,
        'formatter': 'verbose'
    },
    'production_logfile': {
        'level': 'ERROR',
        'filters': ['require_debug_false'],
        'class': 'logging.handlers.RotatingFileHandler',
        'filename': '/tmp/clientPyQt_prod.log',
        'maxBytes': 1024*1024*5, # 5 MB
        'backupCount' : 7,
        'formatter': 'simple'
    },
},
'loggers': {
    'py.warnings': {
        'handlers': ['console', 'development_logfile'],
    },
    '': {
        'handlers': ['console', 'production_logfile', 'development_logfile'],
        'level': "DEBUG",
    },
}
```

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

c

constantes, 5

l

log, 7

m

main, 3

p

proxy, 3

A

ADR_REST (dans le module constantes), 5
ADR_WEB (dans le module constantes), 5
ADR_XMLRPC (dans le module constantes), 5
afficherArticles() (méthode main.MainWindow), 3
AfficherFormulaire() (méthode main.MainWindow), 3
afficherMainWindow() (méthode proxy.Proxy), 3
AjouterArticle() (méthode main.FormArticle), 3
AjouterArticle() (méthode main.MainWindow), 3
ajouterArticle() (méthode proxy.Proxy), 3
Annuler() (méthode main.FormArticle), 3

C

changerProxy() (méthode main.MainWindow), 3
CommitSession() (méthode main.MainWindow), 3
configure() (dans le module log), 7
constantes (module), 5

D

DEBUG (dans le module constantes), 5

F

fermerAppli() (méthode main.MainWindow), 3
FormArticle (classe dans main), 3

I

imprimer() (méthode main.MainWindow), 3

L

listerArticles() (méthode proxy.Proxy), 4
log (module), 7

M

main (module), 3
MainWindow (classe dans main), 3
MODE_ADD (dans le module constantes), 5
MODE_MOD (dans le module constantes), 5
ModifierArticle() (méthode main.MainWindow), 3
modifierArticle() (méthode proxy.Proxy), 4

P

Proxy (classe dans proxy), 3
proxy (module), 3
PROXY_PWD (dans le module constantes), 5
PROXY_USER (dans le module constantes), 5
ProxyREST (classe dans proxy), 4
ProxyWeb (classe dans proxy), 4
ProxyXMLRPC (classe dans proxy), 4

R

RemplirArticles() (méthode main.MainWindow), 3
RequireDebugFalse (classe dans log), 7
RequireDebugTrue (classe dans log), 7
RollbackSession() (méthode main.MainWindow), 3

S

slot_FormArticle_closed() (méthode
main.MainWindow), 3
SupprimerArticle() (méthode main.MainWindow), 3
supprimerArticle() (méthode proxy.Proxy), 4