
pygodaddy Documentation

Release 0.1.6

Jingchao Hu(observerss)

July 15, 2013

CONTENTS

1	Install	3
2	Quick Start	5
3	API Documents	7
4	Indices and tables	9
	Python Module Index	11

PyGoDaddy is a 3rd-party client library, written in Python, for site admins(devs), to make GoDaddy suck less.
Currently, Only A-Record manipulation is supported

INSTALL

To install pygodaddy, simply:

```
pip install pygodaddy
```


QUICK START

First, import and initiate GoDaddyClient

```
from pygodaddy import GoDaddyClient
client = GoDaddyClient()
```

Then we can use login method to login, and update_dns_record method to update dns record (A-Record Only)

```
if client.login(username, password):
    client.update_dns_record('sub.example.com', '1.2.3.4')
```

A list of methods and be found in *API Documents* section below

API DOCUMENTS

class pygodaddy.client.**GoDaddyClient**
GoDaddy Client Library

Typical Usage:

```
>>> from pygodaddy import GoDaddyClient
>>> client = GoDaddyClient()
>>> if client.login(username, password):
...     client.update_dns_record('sub.example.com', '1.2.3.4')
```

delete_dns_record (*hostname*, *record_type='A'*)
delete hostname in accounts

Parameters

- **hostname** – hostname to be deleted
- **record_type** – only 'A' is implemented for now

Returns *True* if successful, else *False*

find_dns_records (*domain*, *record_type='A'*)
find all dns records of a given domain

Parameters **domain** – a typical domain name, e.g. "example.com"

Returns a dict of (hostname -> DNSRecord)

find_domains ()
return all domains of user

is_loggedin (*html=None*)
Test login according to returned html, then set value to self.logged_in

Parameters **html** – the html content returned by self.session.get/post

Returns *True* if there's welcome message, else *False*

login (*username*, *password*)
Login to a godaddy account

Parameters

- **username** – godaddy username
- **password** – godaddy password

Returns *True* if login is successful, else *False*

update_dns_record (*hostname, value, record_type='A', new=True*)

Update a dns record

Parameters

- **hostname** – hostname to update
- **value** – value for hostname to point to, for A record, it's like 'XX.XX.XX.XX'
- **record_type** – only 'A' is implemented for now
- **new** – if *True*, will create a new record if necessary, default to *True*

Returns *True* if update is successful, else *False*

class `pygodaddy.client.GoDaddyAccount` (*username, password*)

This is a context manager for GoDaddyClient

Usage:

```
>>> from pygodaddy import GoDaddyAccount
>>> with GoDaddyAccount(username, password) as client:
...     client.update_dns_record('sub1.exmaple.com', '1.2.3.4')
...     client.update_dns_record('sub2.exmaple.com', '1.2.3.5')
```

INDICES AND TABLES

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

PYTHON MODULE INDEX

p

`pygodaddy.client`, 7