GNTP Documentation

Release 1.0.2

Paul Traylor

March 06, 2016

Contents

Lower level classes for those who want more control in sending messages

CHAPTER 1

Exceptions

exception `gntp.errors.`**`AuthError`**

exception `gntp.errors.`**`ParseError`**

exception `gntp.errors.`**`UnsupportedError`**

CHAPTER 2

GNTP Messages

Classes representing each of the GNTP message types

class `gntp.core.`**`GNTPRegister`**(*data=None*, *password=None*)

Represents a GNTP Registration Command

Parameters

- **`data`** (*`string`*) – (Optional) See decode()
- **`password`** (*`string`*) – (Optional) Password to use while encoding/decoding messages

`add_notification`(*name*, *enabled=True*)

Add new Notification to Registration message

Parameters

- **`name`** (*`string`*) – Notification Name
- **`enabled`** (*`boolean`*) – Enable this notification by default

`add_resource`(*data*)

Add binary resource

Parameters **`data`** (*`string`*) – Binary Data

`decode`(*data*, *password*)

Decode existing GNTP Registration message

Parameters **`data`** (*`string`*) – Message to decode

`encode`()

Encode a GNTP Registration Message

Return string Encoded GNTP Registration message. Returned as a byte string

`set_password`(*password*, *encryptAlgo='MD5'*)

Set a password for a GNTP Message

Parameters

- **`password`** (*`string`*) – Null to clear password
- **`encryptAlgo`** (*`string`*) – Supports MD5, SHA1, SHA256, SHA512

`validate`()

Validate required headers and validate notification headers

class `gntp.core.`**`GNTPNotice`**(*data=None*, *app=None*, *name=None*, *title=None*, *password=None*)

Represents a GNTP Notification Command

Parameters

- **data** (*string*) – (Optional) See decode()
- **app** (*string*) – (Optional) Set Application-Name
- **name** (*string*) – (Optional) Set Notification-Name
- **title** (*string*) – (Optional) Set Notification Title
- **password** (*string*) – (Optional) Password to use while encoding/decoding messages

add_resource(*data*)

Add binary resource

Parameters data (*string*) – Binary Data

decode(*data*, *password*)

Decode existing GNTP Notification message

Parameters data (*string*) – Message to decode.

encode()

Encode a generic GNTP Message

Return string GNTP Message ready to be sent. Returned as a byte string

set_password(*password*, *encryptAlgo='MD5'*)

Set a password for a GNTP Message

Parameters

- **password** (*string*) – Null to clear password
- **encryptAlgo** (*string*) – Supports MD5, SHA1, SHA256, SHA512

validate()

Verify required headers

class gntp.core.**GNTPSubscribe**(*data=None*, *password=None*)

Represents a GNTP Subscribe Command

Parameters

- **data** (*string*) – (Optional) See decode()
- **password** (*string*) – (Optional) Password to use while encoding/decoding messages

add_resource(*data*)

Add binary resource

Parameters data (*string*) – Binary Data

decode(*data*, *password=None*)

Decode GNTP Message

Parameters data (*string*) –

encode()

Encode a generic GNTP Message

Return string GNTP Message ready to be sent. Returned as a byte string

set_password(*password*, *encryptAlgo='MD5'*)

Set a password for a GNTP Message

Parameters

- **password** (*string*) – Null to clear password
- **encryptAlgo** (*string*) – Supports MD5, SHA1, SHA256, SHA512

validate()
Verify required headers

class gntp.core.**GNTPOK**(*data=None*, *action=None*)
Represents a GNTP OK Response

Parameters

- **data** (*string*) – (Optional) See _GNTPResponse.decode()
- **action** (*string*) – (Optional) Set type of action the OK Response is for

add_resource(*data*)
Add binary resource

Parameters data (*string*) – Binary Data

decode(*data*, *password=None*)
Decode GNTP Message

Parameters data (*string*) –

encode()
Encode a generic GNTP Message

Return string GNTP Message ready to be sent. Returned as a byte string

set_password(*password*, *encryptAlgo='MD5'*)
Set a password for a GNTP Message

Parameters

- **password** (*string*) – Null to clear password
- **encryptAlgo** (*string*) – Supports MD5, SHA1, SHA256, SHA512

validate()
Verify required headers

class gntp.core.**GNTPError**(*data=None*, *errorcode=None*, *errordesc=None*)
Represents a GNTP Error response

Parameters

- **data** (*string*) – (Optional) See _GNTPResponse.decode()
- **errorcode** (*string*) – (Optional) Error code
- **errordesc** (*string*) – (Optional) Error Description

add_resource(*data*)
Add binary resource

Parameters data (*string*) – Binary Data

decode(*data*, *password=None*)
Decode GNTP Message

Parameters data (*string*) –

encode()
Encode a generic GNTP Message

Return string GNTP Message ready to be sent. Returned as a byte string

`set_password`(*password*, *encryptAlgo='MD5'*)

Set a password for a GNTP Message

Parameters

- **`password`** (*`string`*) – Null to clear password
- **`encryptAlgo`** (*`string`*) – Supports MD5, SHA1, SHA256, SHA512

`validate`()

Verify required headers

CHAPTER 3

Helper Functions

`gntp.core.`**`parse_gntp`**`(`*data*, *password=None*`)`

Attempt to parse a message as a GNTP message

Parameters

- **`data`** (*`string`*) – Message to be parsed
- **`password`** (*`string`*) – Optional password to be used to verify the message

CHAPTER 4

GNTP Basics

Python bindings for the Growl Notification Transport Protocol

Bugs can be reported at the GitHub issue tracker

4.1 Simple Message Sending

```
from gntp.notifier import mini
# Send a simple growl message with mostly default values
mini("Here's a quick message", callback="http://github.com/")
```

`gntp.notifier.`**`mini`**(*description*, *applicationName=’PythonMini’*, *noteType=’Message’*, *title=’Mini Message’*, *applicationIcon=None*, *hostname=’localhost’*, *password=None*, *port=23053*, *sticky=False*, *priority=None*, *callback=None*, *notificationIcon=None*, *identifier=None*, *notifierFactory=<class ‘gntp.notifier.GrowlNotifier’>*)

Single notification function

Simple notification function in one line. Has only one required parameter and attempts to use reasonable defaults for everything else :param string description: Notification message

> **Warning:** For now, only URL callbacks are supported. In the future, the callback argument will also support a function

4.2 Detailed Message Sending

class `gntp.notifier.`**`GrowlNotifier`**(*applicationName=’Python GNTP’*, *notifications=[]*, *defaultNotifications=None*, *applicationIcon=None*, *hostname=’localhost’*, *password=None*, *port=23053*)

Helper class to simplfy sending Growl messages

Parameters

- **`applicationName`** (*`string`*) – Sending application name
- **`notification`** (*`list`*) – List of valid notifications
- **`defaultNotifications`** (*`list`*) – List of notifications that should be enabled by default

- **applicationIcon** (*string*) – Icon URL
- **hostname** (*string*) – Remote host
- **port** (*integer*) – Remote port

The GrowlNotifier class is intended to mostly mirror the older python bindings for growl

`GrowlNotifier.`**`register`**`()`

Send GNTP Registration

> **Warning:** Before sending notifications to Growl, you need to have sent a registration message at least once

`GrowlNotifier.`**`notify`**`(`*noteType*, *title*, *description*, *icon=None*, *sticky=False*, *priority=None*, *callback=None*, *identifier=None*, *custom={}*`)`

Send a GNTP notifications

> **Warning:** Must have registered with growl beforehand or messages will be ignored

Parameters

- **noteType** (*string*) – One of the notification names registered earlier
- **title** (*string*) – Notification title (usually displayed on the notification)
- **description** (*string*) – The main content of the notification
- **icon** (*string*) – Icon URL path
- **sticky** (*boolean*) – Sticky notification
- **priority** (*integer*) – Message priority level from -2 to 2
- **callback** (*string*) – URL callback
- **custom** (*dict*) – Custom attributes. Key names should be prefixed with X- according to the spec but this is not enforced by this class

> **Warning:** For now, only URL callbacks are supported. In the future, the callback argument will also support a function

`GrowlNotifier.`**`subscribe`**`(`*id*, *name*, *port*`)`

Send a Subscribe request to a remote machine

4.3 Complete Example

```
import gntp.notifier

# Simple "fire and forget" notification
gntp.notifier.mini("Here's a quick message")

# More complete example
growl = gntp.notifier.GrowlNotifier(
        applicationName = "My Application Name",
        notifications = ["New Updates","New Messages"],
        defaultNotifications = ["New Messages"],
        # hostname = "computer.example.com", # Defaults to localhost
        # password = "abc123" # Defaults to a blank password
)
```

```
growl.register()

# Send one message
growl.notify(
        noteType = "New Messages",
        title = "You have a new message",
        description = "A longer message description",
        icon = "http://example.com/icon.png",
        sticky = False,
        priority = 1,
)

# Try to send a different type of message
# This one may fail since it is not in our list
# of defaultNotifications
growl.notify(
        noteType = "New Updates",
        title = "There is a new update to download",
        description = "A longer message description",
        icon = "http://example.com/icon.png",
        sticky = False,
        priority = -1,
)

# Send the image with the growl notification
image = open('/path/to/icon.png', 'rb').read()
growl.notify(
        noteType = "New Messages",
        title = "Now with icons",
        description = "This time we attach the image",
        icon = image,
)
```

4.4 GNTP Configfile Example

class gntp.config.**GrowlNotifier**(*args, **kwargs)

ConfigParser enhanced GrowlNotifier object

For right now, we are only interested in letting users overide certain values from ~/.gntp

```
    [gntp]
    hostname = ?
    password = ?
    port = ?
```