gs.core Documentation

Release 2.2.0

GroupServer.org

April 28, 2016

Contents

This package provides some useful utility functions that are used throughout GroupServer.

Contents:

CHAPTER 1

gs.core API Reference

The package exports the following API symbols.

1.1 String conversion

gs.core.**to_ascii**(*stringOrUnicode*)

Convert a string to ASCII, with a reasonable chance of success.

Parameters **stringOrUnicode** – The instance to convert.

Returns The object converted to a string

Return type str

The to_ascii function, ultimately, calls unicode.encode('ascii', 'ignore'), but it has a couple of advantages.

1. It takes up less space when writing code.
2. If passed a string then the string will be decoded as UTF-8, before being re-encoded as ASCII.

The second point may seem to be redundant, but it avoids the dreaded **Unicode Decode Error** from occurring.

Example: Ensure the filename with the group identifier is ascii:

```
filename = to_ascii('{0}-members.csv'.format(self.groupInfo.id))
```

gs.core.**to_unicode_or_bust**(*stringOrUnicode*, *encoding=u'utf-8'*)

Convert an object to a Unicode instance, with reasonable chance of success.

Parameters

- **stringOrUnicode** – The instance to convert to a unicode.
- **encoding** – The encoding for the object. Defaults to utf-8.

Returns The object converted to a Unicode.

Return type unicode (or str in Python 3)

Sometimes text-input has... uncertain... origins, and it is hard to know encoding it is. The to_unicode_or_bust has a good stab at converting the input to a Unicode instance.

Example: Convert some input into Unicode:

```
filename = gs.core.to_unicode_or_bust(someInput)
```

Acknowledgements: Taken from an excellent presentation on Unicode in Python by Kumar McMillan.

1.2 Lists

gs.core.**comma_comma_and**(*l*, *conj=u'and'*)

Turn a list of strings into a single string, with commas.

Parameters

- **l** (*sequence*) – The strings to convert.
- **conj** (*str*) – The conjunctive to use.

Returns

Either

- An empty string if the list `l` is empty,
- The one item from `l` if `l` is a single item long, or
- A single string that contains all the items from `l` separated by commas (`,`), except for the last two items that are separated by a comma and the conjunctive (`conj`).

Return type `str`

This utility turns a list (such as `['this', 'that', 'the other thing']`) into a single string (`this, that, and the other thing`). It is useful when reporting back from forms.

1.3 Date functions

gs.core.**curr_time**()

Get the current time, in UTC, as a `datetime.datetime`.

Returns The current time, as a class:*datetime.datetime* instance, with the timezone set to `UTC`.

Return type `datetime.datetime`

This function returns the current time, with a timezone, as a standard Python `datetime.datetime` instance. It saves quite a few imports!

1.4 Identifiers

gs.core.**to_id**(*s*)

Create a random identifier, using a string as a seed.

Parameters **s** (*str*) – The string to be used as a seed.

Returns A base-62 encoded string, 22 characters long.

Return type str

Many things require unique identifiers, such as users, posts, topics, password-reset links, and email-verification links. The `to_id` function takes a string and converts it to a fixed-length base-62 encoded string that can be used as an ID.

Example:

Create a verification identifier for an email address:

```
email = emailUser.get_delivery_addresses()[0]
verificationId = to_id(email)
```

gs.core.**convert_int2b62**(*num*)

Convert an integer to a base-62 encoded string.

Parameters **num** (*int*) – The number to convert.

Returns A base-62 encoded string.

Return type str

1.5 Creating a mailto URI

gs.core.**mailto**(*toAddress*, *subject*, *body*)

Create a mailto URI

Parameters

- **toAddress** (*str*) – The address to send the email to (the *To* header).
- **subject** (*str*) – The subject of the new message (the *Subject* header).
- **body** (*str*) – The body of the message

Returns A mailto URI (mailto:).

Return type str

It is possible to create a URI that will create an email message when clicked. Such URIs start with mailto: (**RFC 6068**). However, care must be taken to quote the parameters correctly. This function creates a mailto URI with the correct quoting.

CHAPTER 2

changelog

2.1 2.2.0 (2015-10-15)

- Adding the `mailto` function

2.2 2.1.3 (2015-04-15)

- Fixing an error due to using a list as a default argument

2.3 2.1.2 (2014-09-16)

- Renamed all the `*txt` files as `*rst` files, for GitHub support.

2.4 2.1.1 (2014-04-29)

- Switched to Sphinx documentation
- Tweaked the `to_unicode_or_bust` slightly

2.5 2.1.0 (2014-04-25)

- Python 3 support
- Added unit tests

2.6 2.0.0 (2014-03-07)

- Initial version
  - `to_ascii`
  - `to_unicode_or_bust`
  - `curr_time`

- to_id
- comma_comma_and

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

C

M

R

T