pyrt Documentation

Release 0.2.2

Author

October 02, 2015

Contents

1 pyrt Package **3**
1.1 `pyrt` Package 3
1.2 `pyrt` Module 3

2 Indices and tables **7**

Python Module Index **9**

Contents:

CHAPTER 1

pyrt Package

1.1 pyrt Package

1.2 pyrt Module

Module for Request tracker wrapper.

exception BadRequestException (*message*)

Bases: `exceptions.Exception`

Exception for bad requests.

exception ParseError (*message*)

Bases: `exceptions.Exception`

Error in parsing.

class Ticket (*id_*, *subject*, *data*, *rt*)

Bases: `object`

Represents RT ticket.

Parameters

- **id_** (*str*) – ticket ID
- **subject** (*str*) – ticket subject
- **data** (*str*) – data
- **rt** (*RT*) – RT instance

Raises `TypeError` – if rt is None

comment (*text*)

Add a comment to the ticket.

Parameters **text** (*str*) – the comment text

Returns None

load_all ()

Load all data.

Returns None

load_history ()

Load the history.

Returns None

map_data(*data*)
Map the data to attributes.

Parameters **data ({str** – str}): the data

Returns None

class TicketHistory(*id_*, *rt*)
Bases: `object`

Store and offer views for history.

Parameters

- **id_** (*str*) – the ticket ID
- **rt** (*RT4*) – the RT4 instance

load()
Load all data into the object.

Returns None

class TicketList(*data*, *rt*)
Bases: `object`

Container for tickets.

Parameters **data ({str** – str}/{'id': 'Subject'}): the tickets

list_all()
Return tickets info.

Returns tuple of (int, str)

class RT4(*rest_url=u'http://localhost/REST/1.0/'*)
Bases: `object`

Request tracker.

Parameters **rest_url** (*str*) – REST API URL

add_comment(*id_*, *message*)
Add comment to ticket.

Parameters

- **id_** (*str*) – Ticket ID
- **message** (*str*) – Comment text

Return type None

check_reply(*reply*)
Check a head of a reply and return data without the head.

Parameters **reply** (*str*) – the reply text

Raises `BadRequestException` – if the reply from RT is not OK

Returns *str* – cleaned data

create_group(*group_data*)
Create group.

Parameters **group_data** (*dict - {'content': group data}*) – Group raw data

Returns str

create_ticket (*ticket_data*)
Create ticket and return info.

Parameters **ticket_data** (*dict - {'content': ticket body}*) – Ticket data

Returns str

create_user (*user_data*)
Create user.

Parameters **user_data** (*dict - {'content': user data}*) – User raw data

Returns str

edit_group (*groupname*, *group_data*)
Edit group - limited.

Returns str

get_ticket (*id_*)
Return ticket object with data.

Parameters **id_** (*str*) – Ticket ID

Return type Ticket

get_userlang (*username*)
Return user's language.

Parameters **username** (*str*) – Username

Returns str

get_usermail (*username*)
Try to find user's mail.

Parameters **username** (*str*) – Username

Returns str

load_history (*id_*)
Load history data for ticket.

Parameters **id_** (*str*) – Ticket ID

Return type {str: {str: str}}

load_ticket (*id_*)
Load ticket data and return it as dictionary.

Parameters **id_** (*str*) – Ticket ID

Return type {str: str}

login (*login_name*, *password*)
Save the credentials.

Parameters

- **login_name** (*str*) – login
- **password** (*str*) – password

Returns None

parse_history_reply(*reply*)
Parse history data from string.

Parameters **reply** (*str*) – History reply text

Returns {str: {str: str}}

parse_reply(*reply*)
Parse data from string.

Parameters **reply** (*str*) – Reply text

Returns {str: str}

search_ticket(*query*)
Search tickets according to query and return TicketList.

Parameters **query** (*str*) – Query

Returns `TicketList`

set_userlang(*username*, *user_data*)
Edit user's language. Need root user.

Parameters **username** (*str*) – Username

Returns str

user_exists(*username*)
Try to find user in RT and return boolean value.

It depends on 'Disabled' field from RT user reply.

Parameters **username** (*str*) – Username

Returns bool

CHAPTER 2

Indices and tables

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

p

pyrt.__init__, 3
pyrt.pyrt, 3

A

add_comment() (RT4 method), 4

B

BadRequestException, 3

C

check_reply() (RT4 method), 4
comment() (Ticket method), 3
create_group() (RT4 method), 4
create_ticket() (RT4 method), 5
create_user() (RT4 method), 5

E

edit_group() (RT4 method), 5

G

get_ticket() (RT4 method), 5
get_userlang() (RT4 method), 5
get_usermail() (RT4 method), 5

L

list_all() (TicketList method), 4
load() (TicketHistory method), 4
load_all() (Ticket method), 3
load_history() (RT4 method), 5
load_history() (Ticket method), 3
load_ticket() (RT4 method), 5
login() (RT4 method), 5

M

map_data() (Ticket method), 4

P

parse_history_reply() (RT4 method), 5
parse_reply() (RT4 method), 6
ParseError, 3
pyrt.__init__ (module), 3
pyrt.pyrt (module), 3

R

RT4 (class in pyrt.pyrt), 4

S

search_ticket() (RT4 method), 6
set_userlang() (RT4 method), 6

T

Ticket (class in pyrt.pyrt), 3
TicketHistory (class in pyrt.pyrt), 4
TicketList (class in pyrt.pyrt), 4

U

user_exists() (RT4 method), 6