
ssh*paramikoDocumentation*

Release 0.1.2

Jonatan Dellagostin

January 09, 2017

1	ssh_paramiko	1
2	Usage	3
2.1	Executing a simple remote command	3
2.2	Transferring a remote file to a local file	3
2.3	Transferring a local file to a remote file	4
2.4	Installation	4
2.5	Documentation	4
2.6	Source Code	4
2.7	License	4
3	ssh_paramiko package contents:	5
3.1	ssh_paramiko package	5
4	Indices and tables	9
	Python Module Index	11

ssh_paramiko

ssh_paramiko is a wrapper utility around ssh, using paramiko, to execute commands and exchange files remotely

Usage

2.1 Executing a simple remote command

```
>>> from ssh_paramiko import RemoteServer
>>> ssh = RemoteServer('/tmp/sshkey', logFolder='.')
>>> ssh.set_log_level('DEBUG')
Log: Changing log level to DEBUG | Log level:DEBUG | Date:01/11/2016 12:04:40
>>> ssh.set_log_rotate_handler(True) # set bziped log files (ssh_paramiko.debug.log.bz2
# and ssh_paramiko.error.log.bz2) to be rotated
>>> ssh.connect_server('myServer')
Log: Connecting to server myServer | Log level:DEBUG | Date:01/11/2016 12:04:41
Log: Initiating connection with server myServer... | Log level:DEBUG | Date:01/11/2016 12:04:41
Log: Instantiating transport object for sftp... | Log level:DEBUG | Date:01/11/2016 12:04:41
(True, '')
>>> ssh.execute_cmd('whoami')
(True, 'root\n', '')
>>> ssh.close_connection()
Log: Connection with server myServer ended. | Log level:INFO | Date:01/11/2016 12:04:45
True
```

2.2 Transferring a remote file to a local file

```
>>> from ssh_paramiko import RemoteServer
>>> ssh = RemoteServer('/tmp/sshkey', logFolder='.')
>>> ssh.set_log_level('DEBUG')
Log: Changing log level to DEBUG | Log level:DEBUG | Date:01/11/2016 12:04:40
>>> ssh.set_log_rotate_handler(True) # set bziped log files (ssh_paramiko.debug.log.bz2 and
# ssh_paramiko.error.log.bz2) to be rotated
>>> ssh.connect_server('myServer')
Log: Connecting to server myServer | Log level:DEBUG | Date:01/11/2016 12:04:41
Log: Initiating connection with server myServer... | Log level:DEBUG | Date:01/11/2016 12:04:41
Log: Instantiating transport object for sftp... | Log level:DEBUG | Date:01/11/2016 12:04:41
(True, '')
>>> ssh.get_file('local_file.py', '/root/remote_file.py', callBack=ssh.transferProgressBar)
Log: Transferring remote file /root/remote_file.py from server myServer to local file local_file.py
| Log level:DEBUG | Date:01/11/2016 12:08:15
TrueSize: 542 bytes(0.0 MB) || File transfered. [#####] 100.0%
>>> ssh.close_connection()
Log: Connection with server myServer ended. | Log level:INFO | Date:01/11/2016 12:04:45
True
```

2.3 Transferring a local file to a remote file

```
>>> from ssh_paramiko import RemoteServer
>>> ssh = RemoteServer('/tmp/sshkey', logFolder='.')
>>> ssh.set_log_level('DEBUG')
Log: Changing log level to DEBUG | Log level:DEBUG | Date:01/11/2016 12:07:40
>>> ssh.set_log_rotate_handler(True) # set bziped log files (ssh_paramiko.debug.log.bz2 and
# ssh_paramiko.error.log.bz2) to be rotated
>>> ssh.connect_server('myServer')
Log: Connecting to server myServer | Log level:DEBUG | Date:01/11/2016 12:07:41
Log: Initiating connection with server myServer... | Log level:DEBUG | Date:01/11/2016 12:07:41
Log: Instantiating transport object for sftp... | Log level:DEBUG | Date:01/11/2016 12:07:41
(True, '')
>>> ssh.put_file('local_file.py', '/root/remote_file.py', callBack=ssh.transferProgressBar)
Log: Transferring local file local_file.py to remote file /root/remote_file.py in server myServer |
Log level:DEBUG | Date:01/11/2016 12:07:44
TrueSize: 542 bytes(0.0 MB) || File transfered. [#####] 100.0%
>>> ssh.close_connection()
Log: Connection with server myServer ended. | Log level:INFO | Date:01/11/2016 12:07:44
True
```

2.4 Installation

To install ssh_paramiko, simply run:

```
$ pip install ssh_paramiko
```

ssh_paramiko is compatible with Python 2.6+

2.5 Documentation

https://ssh_paramiko.readthedocs.io

2.6 Source Code

Feel free to fork, evaluate and contribute to this project.

Source: https://github.com/jonDel/ssh_paramiko

2.7 License

GPLv3 licensed.

ssh_paramiko package contents:

3.1 ssh_paramiko package

3.1.1 Submodules

3.1.2 ssh_paramiko.ssh_paramiko module

class `ssh_paramiko.ssh_paramiko.RemoteServer` (*key_ssh*, ***kwargs*)

Bases: `loggers.loggers.Loggers`

Access remote server

Provide a layer of abstraction for accessing, executing commands and transferring files between a host and a server.

Parameters

- **key_ssh** (*str*) – path of the ssh private key to connect (must be `None` if using user and password to connect)
- **log_folder** (*str*, **optional**, *default* = `None`) – folder where the log files of this class will be generated
- **username** (*str*, *optional*, *default* = `root`) – username using the connection
- **password** (*str*, *optional*, *default* = `None`) – password for connection if using user and password instead of key
- **ssh_port** (*str*, *optional*, *default* = `22`) – ssh tcp port
- **server_has_dns** (*bool*, *optional*, *default* = `True`) – if the server is not registered in a `_d_n_s` domain and/or has not its `_d_n_s` name equals to its hostname, this flag must be set to `False`, otherwise this condition will be checked to certify we are trully connected to the right server.

close_connection ()

Closes remote server connection

Returns *True* if successfully disconnected, *False* otherwise

Return type `bool`

connect_server (*server*, *ping* = `True`)

Connects a host and a server via ssh

Parameters

- **server** (*str*) – remote server
- **ping** (*bool*, *default = True*) – if False, ignores if the remote server does not ping back

Returns True if successfully connected, False otherwise

Return type *ret* (*bool*)

Returns Message of error if cannot connect, empty string otherwise

Return type *msg* (*str*)

execute_cmd (*cmd*, *timeout=20*)

Executes a command in a remote server shell

Parameters

- **cmd** (*str*) – command
- **timeout** (*str*) – timeout to the command execution (default: 20)

Returns True if command successfully executed, False otherwise

Return type *ret* (*bool*)

Returns command standard output

Return type *output* (*str*)

Returns command standard error

Return type *error* (*str*)

get_file (*local_file_path*, *remote_file_path*, *callback=None*)

Transfers a remote file to a local file

Parameters

- **local_file_path** (*str*) – path of the local file
- **remote_file_path** (*str*) – path of the remote file
- **callback** (*callback*) – callback that reports file transfer status (bytes transferred and total bytes) _default: None

Returns True if successfully transferred, False otherwise

Return type *bool*

static ping_server (*server*, *tries=4*)

Connects a host and a server via ssh

Parameters

- **server** (*str*) – remote server
- **ping** (*bool*, *default = True*) – if False, ignores if the remote server does not ping back

Returns True if successfully connected, False otherwise

Return type *ret* (*bool*)

Returns Message of error if cannot connect, empty string otherwise

Return type *msg* (*str*)

put_file (*local_file_path*, *remote_file_path*, *callback=None*)

Transfers a local file to a remote file

Parameters

- **local_file_path** (`str`) – path of the local file
- **remote_file_path** (`str`) – path of the remote file
- **callback** (`callback`) – callback that reports file transfer status (bytes transfered and total bytes) _default: None

Returns *True* if successfully transfered, *False* otherwise

Return type `bool`

static transfer_progress_bar (*transfered_bytes*, *total_bytes*)

Provides a transfer progress bar

Given a file to be transfered, print a progress bar according to the bytes transfered yet and the file size to be used as a callback for the methods `put_file` and `get_file`.

Parameters

- **transfered_bytes** (`str`, `int` or `float`) – bytes transfered
- **total_bytes** (`int` or `float`) – size of file in bytes

validate_files (*local_file_path*, *remote_file_path*)

_checks if a remote and local files has the same sha1sum

Parameters

- **local_file_path** (`str`) – path of the local file to be validated
- **remote_file_path** (`str`) – path of the remote file to be validated

Returns *True* if files' sha1sums are the same, *False* otherwise

Return type `bool`

3.1.3 Module contents

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`ssh_paramiko`, 7

`ssh_paramiko.ssh_paramiko`, 5

C

`close_connection()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
method), 5

`connect_server()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
method), 5

E

`execute_cmd()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
method), 6

G

`get_file()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
method), 6

P

`ping_server()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
static method), 6

`put_file()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
method), 6

R

`RemoteServer` (class in `ssh_paramiko.ssh_paramiko`), 5

S

`ssh_paramiko` (module), 7

`ssh_paramiko.ssh_paramiko` (module), 5

T

`transfer_progress_bar()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
static method), 7

V

`validate_files()` (`ssh_paramiko.ssh_paramiko.RemoteServer`
method), 7