

---

# **sPy Documentation**

***Release 0.1***

**Rafał Selewońko**

January 10, 2012



# CONTENTS

<b>1</b>	<b>Structure</b>	<b>3</b>
<b>2</b>	<b>User's guide</b>	<b>7</b>
2.1	Configuration . . . . .	7
2.2	Running sPy . . . . .	7
<b>3</b>	<b>Requirements</b>	<b>9</b>
<b>4</b>	<b>License</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



sPy is Python based script tracking changes at any online or offline location. After detecting changes it sends emails notifying about differences.



# STRUCTURE

```
class spy.AbstractDecorator (subject)
    Decorator pattern.

    Default interface for Site.

    compare ()
    download_new_content ()
    download_old_content ()
    get_diff ()
    get_filename ()
    get_location ()
    get_name ()
    get_new_content ()
    get_old_content ()
    has_changed ()
    notify ()
    parse_new_content (content)
    prepare_notification ()
    save_new_content ()
    set_diff (value)
    set_new_content (value)
    set_old_content (value)

class spy.AbstractProxy (*args, **kwargs)
    Proxy pattern

    Default proxy interface to Site.

    compare ()
    download_new_content ()
    download_old_content ()
    get_diff ()
```

```
get_filename()
get_location()
get_name()
get_new_content()
get_old_content()
has_changed()
notify()
parse_new_content(content)
prepare_notification()
save_new_content()
set_diff(value)
set_new_content(value)
set_old_content(value)

class spy.BinarySite(*args, **kwargs)
    Extends default Site behaviour providing md5 hash check for binaryfiles.

    parse_new_content(content)

class spy.HTMLSite(*args, **kwargs)
    Extends default Site behaviour providing more information about changes in html resource.

exception spy.ImproperlyConfigured

class spy.Mailer
    Singleton pattern.

    Collects, creates and sends notification emails.

    exception NotSingle

    Mailer._Mailer__instance = None

    Mailer.configure(sender, default_recipients, host, port, username, password, use_tls)
        Configures mailer.

    classmethod Mailer.get_instance()
        Returns instance of Mailer. If no instance ever existed it creates one.

    Mailer.make_email(sender, recipient, data)
        Creates and returns email message for single recipient.

    Mailer.make_messages()
        Creates messages for all recipients.

    Mailer.prepare_notification(name, diff, location, recipients=None)
        Adds single notification about single site.

    Mailer.send_messages()
        Sends all messages.

class spy.NewSiteDecorator(*args, **kwargs)
    Supports initial check for new sites.

    compare()
        Nothing to compare ;)
```



```
class spy.OfflineSiteDecorator (subject)
    Provides access to content offline.

    download_new_content ()

class spy.OnlineSiteDecorator (subject)
    Provides access to content online.

    download_new_content ()

class spy.SPy (site_slugs=None)
    Facade pattern.

    configure ()

    get_sites ()

    initialize_mailer ()

    notify ()

    run ()
        Runs everything.

    spy ()

class spy.Site (name, location, slug, recipients=None)
    Represents single resource

    compare ()
        Checks if new content is different than old content.

    download_new_content ()
        Dummy. Left for customization in decorators or proxies.

    download_old_content ()
        Opens old file (downloaded during last check) and returns it's content.

    get_diff ()

    get_filename ()

    get_location ()

    get_name ()

    get_new_content ()
        Returns new (downloaded) content.

    get_old_content ()
        Return old content.

    has_changed ()
        Return True if resource has changed.

    parse_new_content (content)
        Parses new content after downloading.

    prepare_notification ()
        Feeds Mailer.

    save_new_content ()
        Saves new content for future comparisons.

    set_diff (value)
```

**set\_new\_content** (*value*)  
Sets new content from *value*.

**set\_old\_content** (*value*)  
Sets old content from *value*.

**class** `spy.SiteFactory`  
Factory pattern.

**get\_site** (*section*)  
Generates `Site` object.

**class** `spy.TextSite` (*\*args, \*\*kwargs*)  
Extends default `Site` behaviour providing more information (unified diff) about changes in text resource.

**compare** ()

`spy.main` ()  
Parses system args and runs `SPy`.

`spy.slugify` (*value*)  
Normalizes string, converts to lowercase, removes non-alpha characters, and converts spaces to hyphens.

# USER'S GUIDE

## 2.1 Configuration

Config file should be under `~/ .spyrc`. For full example see `.spyrc` in sources.

### 2.1.1 SPY section

SPY section is about email. Here's an example:

```
[SPY]
email_from = sPy <selerto@gmail.com>
email_to = Rafa Selewoko <rselewonko@gmail.com>, Zenek Mietek <selerto@gmail.com>
smtp_host = smtp.gmail.com
smtp_port = 587
smtp_username = selerto@gmail.com
smtp_password = *****
smtp_tls = True
```

### 2.1.2 Sites sections

Each section that's name is not SPY represents site to check. Here's an example:

```
[Rozkład zajęć]
# choices are text, html, binary
type = html
# choices are online or offline
site = online
# starts with ``http`` if online, with ``/`` if offline
location = http://www.degra.wi.pb.edu.pl/rozklady/rozklad.php?page=st

# slug is only required if you want to check each site separately
slug = rozkladzajec
```

## 2.2 Running sPy

To check all sites:

```
./spy.py or python3 /location/of/spy.py
```

To check specified sites:

```
python3 /location/of/spy.py site1_slug site2_slug
```

It's the best to run sPy from **cron** or **Windows Task Sheduler**.

# REQUIREMENTS

- python3



# LICENSE

sPy: Python based script tracking changes at any url Copyright (C) 2011 Rafał Selewońko <[rafal@selewonko.com](mailto:rafal@selewonko.com)>

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.





# PYTHON MODULE INDEX

## S

`spy`, 3



# INDEX

## Symbols

`_Mailer__instance` (spy.Mailer attribute), 4

## A

`AbstractDecorator` (class in spy), 3

`AbstractProxy` (class in spy), 3

## B

`BinarySite` (class in spy), 4

## C

`compare()` (spy.AbstractDecorator method), 3

`compare()` (spy.AbstractProxy method), 3

`compare()` (spy.NewSiteDecorator method), 4

`compare()` (spy.Site method), 5

`compare()` (spy.TextSite method), 6

`configure()` (spy.Mailer method), 4

`configure()` (spy.SPy method), 5

## D

`download_new_content()` (spy.AbstractDecorator method), 3

`download_new_content()` (spy.AbstractProxy method), 3

`download_new_content()` (spy.OfflineSiteDecorator method), 5

`download_new_content()` (spy.OnlineSiteDecorator method), 5

`download_new_content()` (spy.Site method), 5

`download_old_content()` (spy.AbstractDecorator method), 3

`download_old_content()` (spy.AbstractProxy method), 3

`download_old_content()` (spy.Site method), 5

## G

`get_diff()` (spy.AbstractDecorator method), 3

`get_diff()` (spy.AbstractProxy method), 3

`get_diff()` (spy.Site method), 5

`get_filename()` (spy.AbstractDecorator method), 3

`get_filename()` (spy.AbstractProxy method), 3

`get_filename()` (spy.Site method), 5

`get_instance()` (spy.Mailer class method), 4

`get_location()` (spy.AbstractDecorator method), 3

`get_location()` (spy.AbstractProxy method), 4

`get_location()` (spy.Site method), 5

`get_name()` (spy.AbstractDecorator method), 3

`get_name()` (spy.AbstractProxy method), 4

`get_name()` (spy.Site method), 5

`get_new_content()` (spy.AbstractDecorator method), 3

`get_new_content()` (spy.AbstractProxy method), 4

`get_new_content()` (spy.Site method), 5

`get_old_content()` (spy.AbstractDecorator method), 3

`get_old_content()` (spy.AbstractProxy method), 4

`get_old_content()` (spy.Site method), 5

`get_site()` (spy.SiteFactory method), 6

`get_sites()` (spy.SPy method), 5

## H

`has_changed()` (spy.AbstractDecorator method), 3

`has_changed()` (spy.AbstractProxy method), 4

`has_changed()` (spy.Site method), 5

`HTMLSite` (class in spy), 4

## I

`ImproperlyConfigured`, 4

`initialize_mailer()` (spy.SPy method), 5

## M

`Mailer` (class in spy), 4

`Mailer.NotSingle`, 4

`main()` (in module spy), 6

`make_email()` (spy.Mailer method), 4

`make_messages()` (spy.Mailer method), 4

## N

`NewSiteDecorator` (class in spy), 4

`notify()` (spy.AbstractDecorator method), 3

`notify()` (spy.AbstractProxy method), 4

`notify()` (spy.SPy method), 5

## O

`OfflineSiteDecorator` (class in spy), 4

`OnlineSiteDecorator` (class in spy), 5

## P

`parse_new_content()` (`spy.AbstractDecorator` method), 3  
`parse_new_content()` (`spy.AbstractProxy` method), 4  
`parse_new_content()` (`spy.BinarySite` method), 4  
`parse_new_content()` (`spy.Site` method), 5  
`prepare_notification()` (`spy.AbstractDecorator` method), 3  
`prepare_notification()` (`spy.AbstractProxy` method), 4  
`prepare_notification()` (`spy.Mailer` method), 4  
`prepare_notification()` (`spy.Site` method), 5

## R

`run()` (`spy.SPy` method), 5

## S

`save_new_content()` (`spy.AbstractDecorator` method), 3  
`save_new_content()` (`spy.AbstractProxy` method), 4  
`save_new_content()` (`spy.Site` method), 5  
`send_messages()` (`spy.Mailer` method), 4  
`set_diff()` (`spy.AbstractDecorator` method), 3  
`set_diff()` (`spy.AbstractProxy` method), 4  
`set_diff()` (`spy.Site` method), 5  
`set_new_content()` (`spy.AbstractDecorator` method), 3  
`set_new_content()` (`spy.AbstractProxy` method), 4  
`set_new_content()` (`spy.Site` method), 5  
`set_old_content()` (`spy.AbstractDecorator` method), 3  
`set_old_content()` (`spy.AbstractProxy` method), 4  
`set_old_content()` (`spy.Site` method), 6  
`Site` (class in `spy`), 5  
`SiteFactory` (class in `spy`), 6  
`slugify()` (in module `spy`), 6  
`SPy` (class in `spy`), 5  
`spy` (module), 3  
`spy()` (`spy.SPy` method), 5

## T

`TextSite` (class in `spy`), 6