
PRAWdditions

Release 0.2.0

Apr 05, 2020

Contents:

1	prawdditions.patch	3
1.1	prawdditions.patch.patch	3
1.2	prawdditions.patch.unpatch	3
1.3	Patched Functions	3
2	References	5
	Index	7

PRAWdditions is a high-end library wrapper for PRAW that provides several functions that are missing from the main package. The only dependency is PRAW.

prawdditions.patch

This submodule contains functions that directly add attributes to PRAW models. To prevent conflicts with future PRAW additions, the models will not be modified without a call to `patch()`. Furthermore, the additions can be removed with a call to `unpatch()`.

1.1 prawdditions.patch.patch

`prawdditions.patch.patch()`
Apply new methods to classes.

1.2 prawdditions.patch.unpatch

`prawdditions.patch.unpatch()`
Remove added methods.

1.3 Patched Functions

1.3.1 prawdditions.patch.message.message

`prawdditions.patch.message.message` (*self*, *to*: `Union[praw.models.reddit.redditor.Redditor, praw.models.reddit.subreddit.Subreddit, str]`, *title*: `str`, *body*: `str`, *from_sr*: `Union[praw.models.reddit.subreddit.Subreddit, str, None]` = `None`) → `praw.models.reddit.message.Message`
Abstract function for sending out a message via string.

Note: Adds attribute `message` to `praw.Reddit`.

Parameters

- **to** – Destination of the message.
- **title** – The subject line of the message.
- **body** – The body of the message.
- **from_sr** – A Subreddit instance of string to send the message from. By default the message originates from the user.

Returns An instance of `praw.models.Message`.

Example code:

```
import prawdditions.patch
reddit = praw.Reddit(client_id='CLIENT_ID',
                    client_secret="CLIENT_SECRET",
                    password='PASSWORD',
                    user_agent='USERAGENT', username='USERNAME')
prawdditions.patch.patch()
reddit.message('username', 'title', 'body')
```

1.3.2 prawdditions.patch.update.update

`prawdditions.patch.update.update` (*self*, *transformation: Callable[str, str]*, *reason: Optional[str]* = *None*) → *None*

Safely update a page based on its current content.

Note: Adds attribute `update` to `praw.models.WikiPage`.

Parameters

- **transformation** – A function taking the previous content as its sole parameter and returning the new content.
- **reason** – (Optional) The reason for the revision.

Example code:

```
import prawdditions.patch
reddit = praw.Reddit(client_id='CLIENT_ID',
                    client_secret="CLIENT_SECRET",
                    password='PASSWORD',
                    user_agent='USERAGENT', username='USERNAME')
prawdditions.patch.patch()
def transform(content: str) -> str:
    return content + " test"
page = next(iter(reddit.subreddit('test').wiki))
page.update(transform)
```

CHAPTER 2

References

- `genindex`
- `search`

M

`message()` (*in module `prawdditions.patch.message`*), 3

P

`patch()` (*in module `prawdditions.patch`*), 3

U

`unpatch()` (*in module `prawdditions.patch`*), 3

`update()` (*in module `prawdditions.patch.update`*), 4