
Firefox Puppeteer Documentation

Release 0.1

Mozilla

Sep 22, 2017

Contents

1	Installation	3
2	Versioning	5
3	Libraries	7
3.1	About Window	7
3.2	Deck	12
3.3	Menu	12
3.4	Page Info Window	13
3.5	Notifications	17
3.6	Tabbar	20
3.7	Toolbars	23
3.8	BrowserWindow	27
3.9	Update Wizard Dialog	30
3.10	Windows	37
3.11	AppInfo	40
3.12	Keys	40
3.13	Localization	42
3.14	Places	43
3.15	Security	44
3.16	SoftwareUpdate	44
3.17	Utils	46
4	Indices and tables	49

Firefox Puppeteer is a library built on top of the [Marionette Python client](#). It aims to make automation of Firefox's browser UI simpler. It does **not** make sense to use Firefox Puppeteer if:

- You are manipulating something other than Firefox (like Firefox OS)
- You are only manipulating elements in content scope (like a webpage)

Roughly speaking, Firefox Puppeteer provides a library to manipulate each visual section of Firefox's browser UI. For example, there are different libraries for the tab bar, the navigation bar, etc.

CHAPTER 1

Installation

For end-users Firefox Puppeteer can be easily installed as a [Python package](#) from PyPI. If you want to contribute to the project we propose that you clone the [mozilla-central](#) repository and run the following commands:

```
$ cd testing/puppeteer/firefox
$ python setup.py develop
```

In both cases all necessary files including all dependencies will be installed.

CHAPTER 2

Versioning

Puppeteer versions as regularly released from the Python source code, will follow a specific versioning schema. It means the major version number will always be identical with the supported Firefox version. Minor releases - the second part of the version number - are done throughout the life-cycle of a Firefox version when Puppeteer itself needs API changes for back-end and front-end modules. The last part of the version number is the patch level, and is only used for bugfix releases without any API changes.

Examples:

- firefox_puppeteer_45.0.0 - First release for Firefox 45.0 and Firefox 45.xESR
- firefox_puppeteer_46.2.0 - Second release for Firefox 46.0 caused by API changes
- firefox_puppeteer_47.0.1 - First bugfix release for the new Firefox 47.0 support

The following libraries are currently implemented. More will be added in the future. Each library is available from an instance of the `FirefoxTestCase` class.

About Window

AboutWindow

`class firefox_puppeteer.ui.about_window.window.AboutWindow` (*marionette*, *window_handle*)

Representation of the About window.

close (*callback=None*, *force=False*)

Closes the current chrome window.

If this is the last remaining window, the Marionette session is ended.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current `BaseWindow` as parameter. Defaults to `window.open()`.
- **force** – Optional, forces the closing of the window by using the Gecko API. Defaults to `False`.

closed

Returns closed state of the chrome window.

Returns True if the window has been closed.

deck

The `Deck` instance which represents the deck.

Returns Reference to the deck.

focus ()

Sets the focus to the current chrome window.

focused

Returns *True* if the chrome window is focused.

Returns True if the window is focused.

handle

Returns the *window handle* of the chrome window.

Returns *window handle*.

loaded

Checks if the window has been fully loaded.

Returns True, if the window is loaded.

localize_entity (entity_id)

Returns the localized string for the specified DTD entity id.

Parameters **entity_id** – The id to retrieve the value from.

Returns The localized string for the requested entity.

Raises **MarionetteException** – When entity id is not found.

localize_property (property_id)

Returns the localized string for the specified property id.

Parameters **property_id** – The id to retrieve the value from.

Returns The localized string for the requested property.

Raises **MarionetteException** – When property id is not found.

menubar

Provides access to the menu bar, for example, the **File** menu.

See the `MenuBar` reference.

open_window (callback=None, expected_window_class=None, focus=True)

Opens a new top-level chrome window.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current `BaseWindow` as parameter. Defaults to `window.open()`.
- **expected_class** – Optional, check for the correct window class.
- **focus** – Optional, if true, focus the new window. Defaults to *True*.

send_shortcut (command_key, **kwargs)

Sends a keyboard shortcut to the window.

Parameters

- **command_key** – The key (usually a letter) to be pressed.
- **accel** – Optional, If *True*, the *Accel* modifier key is pressed. This key differs between OS X (*Meta*) and Linux/Windows (*Ctrl*). Defaults to *False*.
- **alt** – Optional, If *True*, the *Alt* modifier key is pressed. Defaults to *False*.
- **ctrl** – Optional, If *True*, the *Ctrl* modifier key is pressed. Defaults to *False*.
- **meta** – Optional, If *True*, the *Meta* modifier key is pressed. Defaults to *False*.

- **shift** – Optional, If *True*, the *Shift* modifier key is pressed. Defaults to *False*.

switch_to (*focus=False*)

Switches the context to this chrome window.

By default it will not focus the window. If that behavior is wanted, the *focus* parameter can be used.

Parameters **focus** – If *True*, the chrome window will be focused.

Returns Current window as `BaseWindow` instance.

window_element

Returns the inner DOM window element.

Returns DOM window element.

Deck

class `firefox_puppeteer.ui.about_window.deck.Deck` (*marionette, window, element*)

apply

The `ApplyPanel` instance for the apply panel.

Returns `ApplyPanel` instance.

check_for_updates

The `CheckForUpdatesPanel` instance for the check for updates panel.

Returns `CheckForUpdatesPanel` instance.

checking_for_updates

The `CheckingForUpdatesPanel` instance for the checking for updates panel.

Returns `CheckingForUpdatesPanel` instance.

download_and_install

The `DownloadAndInstallPanel` instance for the download and install panel.

Returns `DownloadAndInstallPanel` instance.

download_failed

The `DownloadFailedPanel` instance for the download failed panel.

Returns `DownloadFailedPanel` instance.

downloading

The `DownloadingPanel` instance for the downloading panel.

Returns `DownloadingPanel` instance.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

no_updates_found

The `NoUpdatesFoundPanel` instance for the no updates found panel.

Returns `NoUpdatesFoundPanel` instance.

panels

List of all the `Panel` instances of the current deck.

Returns List of `Panel` instances.

selected_index

The index of the currently selected panel.

Returns Index of the selected panel.

selected_panel

A `Panel` instance of the currently selected panel.

Returns `Panel` instance.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

ApplyPanel

class `firefox_puppeteer.ui.about_window.deck.ApplyPanel` (*marionette, window, element*)

button

The DOM element which represents the Update button.

Returns Reference to the button element.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

CheckForUpdatesPanel

class `firefox_puppeteer.ui.about_window.deck.CheckForUpdatesPanel` (*marionette, window, element*)

button

The DOM element which represents the Check for Updates button.

Returns Reference to the button element.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

CheckingForUpdatesPanel

class `firefox_puppeteer.ui.about_window.deck.CheckingForUpdatesPanel` (*marionette*, *window*, *element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

DownloadAndInstallPanel

class `firefox_puppeteer.ui.about_window.deck.DownloadAndInstallPanel` (*marionette*, *window*, *element*)

button

The DOM element which represents the Download button.

Returns Reference to the button element.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

DownloadFailedPanel

class `firefox_puppeteer.ui.about_window.deck.DownloadFailedPanel` (*marionette*, *window*, *element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

DownloadingPanel

class `firefox_puppeteer.ui.about_window.deck.DownloadingPanel` (*marionette*, *window*, *element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

Deck

Panel

class `firefox_puppeteer.ui.deck.Panel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

Menu

Menu Bar

class `firefox_puppeteer.ui.menu.MenuBar` (*marionette*)

Wraps the menubar DOM element inside a browser window.

class `MenuElement` (*element*)

Wraps a menu element DOM element.

items

A list of menuitem DOM elements within this `MenuElement` instance.

Returns A list of items in the menu.

select (*label*)

Click on a menu item within this menu.

Parameters `label` – The label of the menu item, e.g., `New Tab`.

select_by_id (*menu_item_id*)

Click on a menu item within this menu.

Parameters `menu_item_id` – The ID of the menu item, e.g. `menu_newNavigatorTab`.

`MenuBar.get_menu` (*label*)

Get a `MenuElement` instance corresponding to the specified label.

Parameters `label` – The label of the menu, e.g., `File` or `View`.

Returns A `MenuElement` instance.

`MenuBar.get_menu_by_id` (*menu_id*)

Get a `MenuElement` instance corresponding to the specified ID.

Parameters `menu_id` – The ID of the menu, e.g., `file-menu` or `view-menu`.

Returns A `MenuElement` instance.

`MenuBar`.**menus**

A list of `MenuElement` instances corresponding to the top level menus in the menubar.

Returns A list of `MenuElement` instances.

`MenuBar`.**select** (*label*, *item*)

Select an item in a menu.

Parameters

- **label** – The label of the menu, e.g., `File` or `View`.
- **item** – The label of the item in the menu, e.g., `New Tab`.

`MenuBar`.**select_by_id** (*menu_id*, *item_id*)

Select an item in a menu.

Parameters

- **menu_id** – The ID of the menu, e.g. `file-menu` or `view-menu`.
- **item_id** – The ID of the item in the menu, e.g. `menu_newNavigatorTab`.

Page Info Window

PageInfoWindow

class `firefox_puppeteer.ui.pageinfo.window.PageInfoWindow` (*marionette*, *window_handle*)

Representation of a page info window.

close (*trigger*='shortcut', *force*=*False*)

Closes the current page info window by using the specified trigger.

Parameters

- **trigger** – Optional, method to close the current window. This can be a string with one of `menu` (OS X only) or `shortcut`, or a callback which gets triggered with the current `PageInfoWindow` as parameter. Defaults to `shortcut`.
- **force** – Optional, forces the closing of the window by using the Gecko API. Defaults to `False`.

closed

Returns closed state of the chrome window.

Returns True if the window has been closed.

deck

The `Deck` instance which represents the deck.

Returns Reference to the deck.

focus ()

Sets the focus to the current chrome window.

focused

Returns `True` if the chrome window is focused.

Returns True if the window is focused.

handle

Returns the *window handle* of the chrome window.

Returns *window handle*.

loaded

Checks if the window has been fully loaded.

Returns True, if the window is loaded.

localize_entity (*entity_id*)

Returns the localized string for the specified DTD entity id.

Parameters *entity_id* – The id to retrieve the value from.

Returns The localized string for the requested entity.

Raises **MarionetteException** – When entity id is not found.

localize_property (*property_id*)

Returns the localized string for the specified property id.

Parameters *property_id* – The id to retrieve the value from.

Returns The localized string for the requested property.

Raises **MarionetteException** – When property id is not found.

menubar

Provides access to the menu bar, for example, the **File** menu.

See the `MenuBar` reference.

open_window (*callback=None, expected_window_class=None, focus=True*)

Opens a new top-level chrome window.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current `BaseWindow` as parameter. Defaults to `window.open()`.
- **expected_class** – Optional, check for the correct window class.
- **focus** – Optional, if true, focus the new window. Defaults to `True`.

send_shortcut (*command_key, **kwargs*)

Sends a keyboard shortcut to the window.

Parameters

- **command_key** – The key (usually a letter) to be pressed.
- **accel** – Optional, If `True`, the *Accel* modifier key is pressed. This key differs between OS X (*Meta*) and Linux/Windows (*Ctrl*). Defaults to `False`.
- **alt** – Optional, If `True`, the *Alt* modifier key is pressed. Defaults to `False`.
- **ctrl** – Optional, If `True`, the *Ctrl* modifier key is pressed. Defaults to `False`.
- **meta** – Optional, If `True`, the *Meta* modifier key is pressed. Defaults to `False`.
- **shift** – Optional, If `True`, the *Shift* modifier key is pressed. Defaults to `False`.

switch_to (*focus=False*)

Switches the context to this chrome window.

By default it will not focus the window. If that behavior is wanted, the *focus* parameter can be used.

Parameters **focus** – If *True*, the chrome window will be focused.

Returns Current window as `BaseWindow` instance.

window_element

Returns the inner DOM window element.

Returns DOM window element.

Deck

class `firefox_puppeteer.ui.pageinfo.deck.Deck` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

feed

The `FeedPanel` instance for the feed panel.

Returns `FeedPanel` instance.

general

The `GeneralPanel` instance for the general panel.

Returns `GeneralPanel` instance.

media

The `MediaPanel` instance for the media panel.

Returns `MediaPanel` instance.

panels

List of all the `Panel` instances of the current deck.

Returns List of `Panel` instances.

permissions

The `PermissionsPanel` instance for the permissions panel.

Returns `PermissionsPanel` instance.

security

The `SecurityPanel` instance for the security panel.

Returns `SecurityPanel` instance.

select (*panel*)

Selects the specified panel via the tab element.

Parameters **panel** – The panel to select.

Returns `Panel` instance of the selected panel.

selected_index

The index of the currently selected panel.

Returns Index of the selected panel.

selected_panel

A `Panel` instance of the currently selected panel.

Returns Panel instance.

window

Returns the reference to the chrome window.

Returns BaseWindow instance of the chrome window.

PageInfoPanel

class `firefox_puppeteer.ui.pageinfo.deck.PageInfoPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

tab

The DOM element which represents the corresponding tab element at the top.

Returns Reference to the tab element.

window

Returns the reference to the chrome window.

Returns BaseWindow instance of the chrome window.

SecurityPanel

class `firefox_puppeteer.ui.pageinfo.deck.SecurityPanel` (*marionette, window, element*)

domain

The DOM element which represents the domain textbox.

Returns Reference to the textbox element.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

owner

The DOM element which represents the owner textbox.

Returns Reference to the textbox element.

tab

The DOM element which represents the corresponding tab element at the top.

Returns Reference to the tab element.

verifier

The DOM element which represents the verifier textbox.

Returns Reference to the textbox element.

view_certificate

The DOM element which represents the view certificate button.

Returns Reference to the button element.

view_cookies

The DOM element which represents the view cookies button.

Returns Reference to the button element.

view_passwords

The DOM element which represents the view passwords button.

Returns Reference to the button element.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

Notifications

AddOnInstallBlockedNotification

class `firefox_puppeteer.ui.browser.notifications.AddOnInstallBlockedNotification` (*marionette, window, element*)

Bases: `firefox_puppeteer.ui.browser.notifications.BaseNotification`

Add-on install blocked notification.

allow_button

Provide access to the allow button.

Returns The allow button.

close (*force=False*)

Close the notification.

Parameters **force** – Optional, if True force close the notification. Defaults to False.

close_button

Provide access to the close button.

Returns The close button.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

label

Provide access to the notification label.

Returns The notification label.

origin

Provide access to the notification origin.

Returns The notification origin.

window

Returns the reference to the chrome window.

Returns BaseWindow instance of the chrome window.

AddOnInstallConfirmationNotification

class `firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmationNotification` (*marionette*, *window*, *element*)

Bases: `firefox_puppeteer.ui.browser.notifications.BaseNotification`

Add-on install confirmation notification.

close (*force=False*)

Close the notification.

Parameters **force** – Optional, if True force close the notification. Defaults to False.

close_button

Provide access to the close button.

Returns The close button.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

label

Provide access to the notification label.

Returns The notification label.

origin

Provide access to the notification origin.

Returns The notification origin.

window

Returns the reference to the chrome window.

Returns BaseWindow instance of the chrome window.

AddOnInstallCompleteNotification

class `firefox_puppeteer.ui.browser.notifications.AddOnInstallCompleteNotification` (*marionette*, *window*, *element*)

Bases: `firefox_puppeteer.ui.browser.notifications.BaseNotification`

Add-on install complete notification.

close (*force=False*)

Close the notification.

Parameters **force** – Optional, if True force close the notification. Defaults to False.

close_button

Provide access to the close button.

Returns The close button.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

label

Provide access to the notification label.

Returns The notification label.

origin

Provide access to the notification origin.

Returns The notification origin.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

AddOnInstallFailedNotification

class `firefox_puppeteer.ui.browser.notifications.AddOnInstallFailedNotification` (*marionette, window, element*)

Bases: `firefox_puppeteer.ui.browser.notifications.BaseNotification`

Add-on install failed notification.

close (*force=False*)

Close the notification.

Parameters **force** – Optional, if True force close the notification. Defaults to False.

close_button

Provide access to the close button.

Returns The close button.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

label

Provide access to the notification label.

Returns The notification label.

origin

Provide access to the notification origin.

Returns The notification origin.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

AddOnProgressNotification

class `firefox_puppeteer.ui.browser.notifications.AddOnProgressNotification` (*marionette, window, element*)

Bases: `firefox_puppeteer.ui.browser.notifications.BaseNotification`

Add-on progress notification.

close (*force=False*)

Close the notification.

Parameters **force** – Optional, if True force close the notification. Defaults to False.

close_button

Provide access to the close button.

Returns The close button.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

label

Provide access to the notification label.

Returns The notification label.

origin

Provide access to the notification origin.

Returns The notification origin.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

Tabbar

TabBar

class `firefox_puppeteer.ui.browser.tabbar.TabBar` (*marionette, window, element*)
Wraps the tabs toolbar DOM element inside a browser window.

close_all_tabs (*exceptions=None*)

Forces closing of all open tabs.

There is an optional *exceptions* list, which can be used to exclude specific tabs from being closed.

Parameters **exceptions** – Optional, list of *Tab* instances not to close.

close_tab (*tab=None, trigger='menu', force=False*)

Closes the tab by using the specified trigger.

By default the currently selected tab will be closed. If another *Tab* is specified, that one will be closed instead. Also when the tab is closed, a *switch_to()* call is automatically performed, so that the new selected tab becomes active.

Parameters

- **tab** – Optional, the *Tab* instance to close. Defaults to the currently selected tab.
- **trigger** – Optional, method to close the current tab. This can be a string with one of *menu* or *shortcut*, or a callback which gets triggered with the *Tab* as parameter. Defaults to *menu*.
- **force** – Optional, forces the closing of the window by using the Gecko API. Defaults to *False*.

static get_handle_for_tab (*marionette, tab_element*)

Retrieves the marionette handle for the given *Tab* instance.

Parameters

- **marionette** – An instance of the Marionette client.
- **tab_element** – The DOM element corresponding to a tab inside the tabs toolbar.

Returns *handle* of the tab.

menupanel

A *MenuPanel* instance which represents the menu panel at the far right side of the tabs toolbar.

Returns *MenuPanel* instance.

newtab_button

The DOM element which represents the new tab button.

Returns Reference to the new tab button.

open_tab (*trigger='menu'*)

Opens a new tab in the current browser window.

If the tab opens in the foreground, a call to *switch_to()* will automatically be performed. But if it opens in the background, the current tab will keep its focus.

Parameters trigger – Optional, method to open the new tab. This can be a string with one of *menu*, *button* or *shortcut*, or a callback which gets triggered with the current *Tab* as parameter. Defaults to *menu*.

Returns *Tab* instance for the opened tab.

restore_tabs_button

The DOM element which represents the restore tabs button.

Returns Reference to the restore tabs button.

restore_tabs_button_wrapper

The DOM element which represents the restore tabs button wrapper.

Returns Reference to the restore tabs button wrapper.

selected_index

The index of the currently selected tab.

Returns Index of the selected tab.

selected_tab

A *Tab* instance of the currently selected tab.

Returns *Tab* instance.

switch_to (*target*)

Switches the context to the specified tab.

Parameters **target** – The tab to switch to. *target* can be an index, a *Tab* instance, or a callback that returns True in the context of the desired tab.

Returns Instance of the selected *Tab*.

tabs

List of all the *Tab* instances of the current browser window.

Returns List of *Tab* instances.

toolbar

The DOM element which represents the tab toolbar.

Returns Reference to the tabs toolbar.

Tab

class `firefox_puppeteer.ui.browser.tabbar.Tab` (*marionette, window, element*)

Wraps a tab DOM element.

certificate

The SSL certificate associated with the loaded web page.

Returns Certificate details as JSON blob.

close (*trigger='menu', force=False*)

Closes the tab by using the specified trigger.

When the tab is closed a `switch_to()` call is automatically performed, so that the new selected tab becomes active.

Parameters

- **trigger** – Optional, method in how to close the tab. This can be a string with one of *button*, *menu* or *shortcut*, or a callback which gets triggered with the current *Tab* as parameter. Defaults to *menu*.
- **force** – Optional, forces the closing of the window by using the Gecko API. Defaults to *False*.

close_button

The DOM element which represents the tab close button.

Returns Reference to the tab close button.

handle

The *handle* of the content window.

Returns content window *handle*.

location

Returns the current URL

Returns Current URL

select ()

Selects the tab and sets the focus to it.

selected

Checks if the tab is selected.

Returns *True* if the tab is selected.

switch_to ()

Switches the context of Marionette to this tab.

Please keep in mind that calling this method will not select the tab. Use the *select ()* method instead.

tab_element

The inner tab DOM element.

Returns Tab DOM element.

MenuPanel

class `firefox_puppeteer.ui.browser.tabbar.MenuPanel` (*marionette, window, element*)

class `MenuPanelElement` (*element*)

Wraps the menu panel.

buttons

Returns A list of all the clickable buttons in the menu panel.

click (*target=None*)

Overrides `HTMLElement.click` to provide a target to click.

Parameters *target* – The label associated with the button to click on, e.g., *New Private Window*.

`MenuPanel.popup`

Returns The *MenuPanelElement*.

Toolbars

NavBar

class `firefox_puppeteer.ui.browser.toolbars.NavBar` (**args, **kwargs*)

Provides access to the DOM elements contained in the navigation bar as well as the location bar.

back_button

Provides access to the DOM element back button in the navbar.

Returns Reference to the back button.

forward_button

Provides access to the DOM element forward button in the navbar.

Returns Reference to the forward button.

home_button

Provides access to the DOM element home button in the navbar.

Returns Reference to the home button element

locationbar

Provides access to the DOM elements contained in the locationbar.

See the *LocationBar* reference.

menu_button

Provides access to the DOM element menu button in the navbar.

Returns Reference to the menu button element.

toolbar

The DOM element which represents the navigation toolbar.

Returns Reference to the navigation toolbar.

LocationBar

class `firefox_puppeteer.ui.browser.toolbars.LocationBar (*args, **kwargs)`

Provides access to and methods for the DOM elements contained in the locationbar (the text area of the ui that typically displays the current url).

autocomplete_results

Provides access to and methods for the location bar autocomplete results.

See the *AutocompleteResults* reference.

clear ()

Clears the contents of the url bar (via the DELETE shortcut).

close_context_menu ()

Closes the Location Bar context menu by a key event.

connection_icon

Provides access to the urlbar connection icon.

Returns Reference to the connection icon element.

contextmenu

Provides access to the urlbar context menu.

Returns Reference to the urlbar context menu.

focus (event='click')

Focus the location bar according to the provided event.

Parameters event – The event to synthesize in order to focus the urlbar (one of *click* or *shortcut*).

focused

Checks the focus state of the location bar.

Returns *True* if focused, otherwise *False*

get_contextmenu_entry (action)

Retrieves the urlbar context menu entry corresponding to the given action.

Parameters action – The action corresponding to the retrieved value.

Returns Reference to the urlbar contextmenu entry.

history_drop_marker

Provides access to the history drop marker.

Returns Reference to the history drop marker.

identity_box

The DOM element which represents the identity box.

Returns Reference to the identity box.

identity_country_label

The DOM element which represents the identity icon country label.

Returns Reference to the identity icon country label.

identity_icon

Provides access to the urlbar identity icon.

Returns Reference to the identity icon element.

identity_organization_label

The DOM element which represents the identity icon label.

Returns Reference to the identity icon label.

identity_popup

Provides utility members for accessing and manipulating the identity popup.

See the *IdentityPopup* reference.

load_url (url)

Load the specified url in the location bar by synthesized keystrokes.

Parameters **url** – The url to load.

notification_popup

Provides access to the DOM element notification popup.

Returns Reference to the notification popup.

open_identity_popup ()

Open the identity popup.

reload_button

Provides access to the DOM element reload button.

Returns Reference to the reload button.

reload_url (trigger='button', force=False)

Reload the currently open page.

Parameters

- **trigger** – The event type to use to cause the reload (one of *shortcut*, *shortcut2*, or *button*).
- **force** – Whether to cause a forced reload.

stop_button

Provides access to the DOM element stop button.

Returns Reference to the stop button.

urlbar

Provides access to the DOM element urlbar.

Returns Reference to the url bar.

urlbar_input

Provides access to the urlbar input element.

Returns Reference to the urlbar input.

value

Provides access to the currently displayed value of the urlbar.

Returns The urlbar value.

AutocompleteResults

class `firefox_puppeteer.ui.browser.toolbars.AutocompleteResults` (*marionette*, *window*, *element*)

Wraps DOM elements and methods for interacting with autocomplete results.

close (*force=False*)

Closes the urlbar autocomplete popup.

Parameters **force** – If true, the popup is closed by its own hide function, otherwise a key event is sent to close the popup.

get_matching_text (*result*, *match_type*)

Returns an array of strings of the matching text within an autocomplete result in the urlbar.

Parameters

- **result** – The result to inspect for matches.
- **match_type** – The type of match to search for (one of *title* or *url*).

is_complete

Returns when this popup is open and autocomplete results are complete.

Returns True, when autocomplete results have been populated.

is_open

Returns whether this popup is currently open.

Returns True when the popup is open, otherwise false.

results

Returns The autocomplete result container node.

selected_index

Provides the index of the selected item in the autocomplete list.

Returns The index.

visible_results

Supplies the list of visible autocomplete result nodes.

Returns The list of visible results.

IdentityPopup

class `firefox_puppeteer.ui.browser.toolbars.IdentityPopup` (**args*, ***kwargs*)

Wraps DOM elements and methods for interacting with the identity popup.

close (*force=False*)

Closes the identity popup by hitting the escape key.

Parameters **force** – Optional, If *True* force close the popup. Defaults to *False*

is_open

Returns whether this popup is currently open.

Returns True when the popup is open, otherwise false.

view

Provides utility members for accessing and manipulating the identity popup's multi view.

See the `IdentityPopupMultiView` reference.

BrowserWindow

BrowserWindow

class `firefox_puppeteer.ui.browser.window.BrowserWindow` (*args, **kwargs)

Representation of a browser window.

close (*trigger='menu', force=False*)

Closes the current browser window by using the specified trigger.

Parameters

- **trigger** – Optional, method to close the current browser window. This can be a string with one of *menu* or *shortcut*, or a callback which gets triggered with the current `BrowserWindow` as parameter. Defaults to *menu*.
- **force** – Optional, forces the closing of the window by using the Gecko API. Defaults to *False*.

closed

Returns closed state of the chrome window.

Returns True if the window has been closed.

default_homepage

The default homepage as used by the current locale.

Returns The default homepage for the current locale.

focus ()

Sets the focus to the current chrome window.

focused

Returns *True* if the chrome window is focused.

Returns True if the window is focused.

get_final_url (*url*)

Loads the page at *url* and returns the resulting url.

This function enables testing redirects.

Parameters *url* – The url to test.

Returns The resulting loaded url.

handle

Returns the *window handle* of the chrome window.

Returns *window handle*.

is_private

Returns True if this is a Private Browsing window.

loaded

Checks if the window has been fully loaded.

Returns True, if the window is loaded.

localize_entity (*entity_id*)

Returns the localized string for the specified DTD entity id.

Parameters **entity_id** – The id to retrieve the value from.

Returns The localized string for the requested entity.

Raises **MarionetteException** – When entity id is not found.

localize_property (*property_id*)

Returns the localized string for the specified property id.

Parameters **property_id** – The id to retrieve the value from.

Returns The localized string for the requested property.

Raises **MarionetteException** – When property id is not found.

menubar

Provides access to the menu bar, for example, the **File** menu.

See the `MenuBar` reference.

navbar

Provides access to the navigation bar. This is the toolbar containing the back, forward and home buttons. It also contains the location bar.

See the `NavBar` reference.

notification

Provides access to the currently displayed notification.

open_about_window (*trigger='menu'*)

Opens the about window by using the specified trigger.

Parameters **trigger** – Optional, method in how to open the new browser window. This can either be the string *menu* or a callback which gets triggered with the current `BrowserWindow` as parameter. Defaults to *menu*.

Returns `AboutWindow` instance of the opened window.

open_browser (*trigger='menu', is_private=False*)

Opens a new browser window by using the specified trigger.

Parameters

- **trigger** – Optional, method in how to open the new browser window. This can be a string with one of *menu* or *shortcut*, or a callback which gets triggered with the current `BrowserWindow` as parameter. Defaults to *menu*.
- **is_private** – Optional, if True the new window will be a private browsing one.

Returns `BrowserWindow` instance for the new browser window.

open_page_info_window (*trigger='menu'*)

Opens the page info window by using the specified trigger.

Parameters **trigger** – Optional, method in how to open the new browser window. This can be a string with one of *menu* or *shortcut*, or a callback which gets triggered with the current `BrowserWindow` as parameter. Defaults to *menu*.

Returns PageInfoWindow instance of the opened window.

open_window (*callback=None, expected_window_class=None, focus=True*)

Opens a new top-level chrome window.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current BaseWindow as parameter. Defaults to *window.open()*.
- **expected_class** – Optional, check for the correct window class.
- **focus** – Optional, if true, focus the new window. Defaults to *True*.

send_shortcut (*command_key, **kwargs*)

Sends a keyboard shortcut to the window.

Parameters

- **command_key** – The key (usually a letter) to be pressed.
- **accel** – Optional, If *True*, the *Accel* modifier key is pressed. This key differs between OS X (*Meta*) and Linux/Windows (*Ctrl*). Defaults to *False*.
- **alt** – Optional, If *True*, the *Alt* modifier key is pressed. Defaults to *False*.
- **ctrl** – Optional, If *True*, the *Ctrl* modifier key is pressed. Defaults to *False*.
- **meta** – Optional, If *True*, the *Meta* modifier key is pressed. Defaults to *False*.
- **shift** – Optional, If *True*, the *Shift* modifier key is pressed. Defaults to *False*.

switch_to (*focus=False*)

Switches the context to this chrome window.

By default it will not focus the window. If that behavior is wanted, the *focus* parameter can be used.

Parameters **focus** – If *True*, the chrome window will be focused.

Returns Current window as BaseWindow instance.

tabbar

Provides access to the tab bar.

See the TabBar reference.

wait_for_notification (*notification_class=<class 'firefox_puppeteer.ui.browser.notifications.BaseNotification'>, timeout=5*)

Waits for the specified notification to be displayed.

Parameters

- **notification_class** – Optional, the notification class to wait for. If *None* is specified it will wait for any notification to be closed. Defaults to *BaseNotification*.
- **timeout** – Optional, how long to wait for the expected notification. Defaults to 5 seconds.

window_element

Returns the inner DOM window element.

Returns DOM window element.

Update Wizard Dialog

UpdateWizardDialog

`class` `firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog` (*marionette*,
window_handle)

Representation of the old Software Update Wizard Dialog.

close (*callback=None, force=False*)
Closes the current chrome window.

If this is the last remaining window, the Marionette session is ended.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current `BaseWindow` as parameter. Defaults to `window.open()`.
- **force** – Optional, forces the closing of the window by using the Gecko API. Defaults to `False`.

closed

Returns closed state of the chrome window.

Returns True if the window has been closed.

focus ()

Sets the focus to the current chrome window.

focused

Returns `True` if the chrome window is focused.

Returns True if the window is focused.

handle

Returns the *window handle* of the chrome window.

Returns *window handle*.

loaded

Checks if the window has been fully loaded.

Returns True, if the window is loaded.

localize_entity (*entity_id*)

Returns the localized string for the specified DTD entity id.

Parameters **entity_id** – The id to retrieve the value from.

Returns The localized string for the requested entity.

Raises **MarionetteException** – When entity id is not found.

localize_property (*property_id*)

Returns the localized string for the specified property id.

Parameters **property_id** – The id to retrieve the value from.

Returns The localized string for the requested property.

Raises **MarionetteException** – When property id is not found.

menubar

Provides access to the menu bar, for example, the **File** menu.

See the `MenuBar` reference.

open_window (*callback=None, expected_window_class=None, focus=True*)

Opens a new top-level chrome window.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current `BaseWindow` as parameter. Defaults to `window.open()`.
- **expected_class** – Optional, check for the correct window class.
- **focus** – Optional, if true, focus the new window. Defaults to `True`.

select_next_page ()

Clicks on “Next” button, and waits for the next page to show up.

send_shortcut (*command_key, **kwargs*)

Sends a keyboard shortcut to the window.

Parameters

- **command_key** – The key (usually a letter) to be pressed.
- **accel** – Optional, If `True`, the *Accel* modifier key is pressed. This key differs between OS X (*Meta*) and Linux/Windows (*Ctrl*). Defaults to `False`.
- **alt** – Optional, If `True`, the *Alt* modifier key is pressed. Defaults to `False`.
- **ctrl** – Optional, If `True`, the *Ctrl* modifier key is pressed. Defaults to `False`.
- **meta** – Optional, If `True`, the *Meta* modifier key is pressed. Defaults to `False`.
- **shift** – Optional, If `True`, the *Shift* modifier key is pressed. Defaults to `False`.

switch_to (*focus=False*)

Switches the context to this chrome window.

By default it will not focus the window. If that behavior is wanted, the *focus* parameter can be used.

Parameters **focus** – If `True`, the chrome window will be focused.

Returns Current window as `BaseWindow` instance.

window_element

Returns the inner DOM window element.

Returns DOM window element.

wizard

The `Wizard` instance which represents the wizard.

Returns Reference to the wizard.

Wizard

`class firefox_puppeteer.ui.update_wizard.wizard.Wizard (*args, **kwargs)`

checking

The checking for updates panel.

Returns `CheckingPanel` instance.

downloading

The downloading panel.

Returns `DownloadingPanel` instance.

dummy

The dummy panel.

Returns `DummyPanel` instance.

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

error

The errors panel.

Returns `ErrorPanel` instance.

error_extra

The error extra panel.

Returns `ErrorExtraPanel` instance.

error_patching

The error patching panel.

Returns `ErrorPatchingPanel` instance.

finished

The finished panel.

Returns `FinishedPanel` instance.

finished_background

The finished background panel.

Returns `FinishedBackgroundPanel` instance.

incompatible_check

The incompatible check panel.

Returns `IncompatibleCheckPanel` instance.

incompatible_list

The incompatible list panel.

Returns `IncompatibleListPanel` instance.

manual_update

The manual update panel.

Returns `ManualUpdatePanel` instance.

no_updates_found

The no updates found panel.

Returns `NoUpdatesFoundPanel` instance.

panels

List of all the available `Panel` instances.

Returns List of `Panel` instances.

selected_index

The index of the currently selected panel.

Returns Index of the selected panel.

selected_panel

A `Panel` instance of the currently selected panel.

Returns `Panel` instance.

updates_found_basic

The updates found panel.

Returns `UpdatesFoundPanel` instance.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

CheckingPanel

class `firefox_puppeteer.ui.update_wizard.wizard.CheckingPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

progress

The DOM element which represents the progress meter.

Returns Reference to the progress element.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

DownloadingPanel

class `firefox_puppeteer.ui.update_wizard.wizard.DownloadingPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

progress

The DOM element which represents the progress meter.

Returns Reference to the progress element.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

DummyPanel

class `firefox_puppeteer.ui.update_wizard.wizard.DummyPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

ErrorPatchingPanel

class `firefox_puppeteer.ui.update_wizard.wizard.ErrorPatchingPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

ErrorPanel

class `firefox_puppeteer.ui.update_wizard.wizard.ErrorPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

ErrorExtraPanel

class `firefox_puppeteer.ui.update_wizard.wizard.ErrorExtraPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

FinishedPanel

class `firefox_puppeteer.ui.update_wizard.wizard.FinishedPanel` (*marionette*, *window*, *element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

FinishedBackgroundPanel

class `firefox_puppeteer.ui.update_wizard.wizard.FinishedBackgroundPanel` (*marionette*, *window*, *element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

IncompatibleCheckPanel

class `firefox_puppeteer.ui.update_wizard.wizard.IncompatibleCheckPanel` (*marionette*, *window*, *element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

progress

The DOM element which represents the progress meter.

Returns Reference to the progress element.

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

IncompatibleListPanel

`class firefox_puppeteer.ui.update_wizard.wizard.IncompatibleListPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

InstalledPanel

LicensePanel

ManualUpdatePanel

`class firefox_puppeteer.ui.update_wizard.wizard.ManualUpdatePanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

NoUpdatesFoundPanel

`class firefox_puppeteer.ui.update_wizard.wizard.NoUpdatesFoundPanel` (*marionette, window, element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

PluginUpdatesFoundPanel

UpdatesFoundBasicPanel

`class firefox_puppeteer.ui.update_wizard.wizard.UpdatesFoundBasicPanel` (*marionette*, *window*, *element*)

element

Returns the reference to the underlying DOM element.

Returns Reference to the DOM element

window

Returns the reference to the chrome window.

Returns `BaseWindow` instance of the chrome window.

Windows

Windows

`class firefox_puppeteer.ui.windows.Windows` (*marionette*)

all

Retrieves a list of all open chrome windows.

Returns List of `BaseWindow` instances corresponding to the windows in *marionette.chrome_window_handles*.

close (*handle*)

Closes the chrome window with the given handle.

Parameters **handle** – The handle of the chrome window.

close_all (*exceptions=None*)

Closes all open chrome windows.

There is an optional *exceptions* list, which can be used to exclude specific chrome windows from being closed.

Parameters **exceptions** – Optional, list of `BaseWindow` instances not to close.

create_window_instance (*handle*, *expected_class=None*)

Creates a `BaseWindow` instance for the given chrome window.

Parameters

- **handle** – The handle of the chrome window.
- **expected_class** – Optional, check for the correct window class.

current

Retrieves the currently selected chrome window.

Returns The `BaseWindow` for the currently active window.

focus (*handle*)

Focuses the chrome window with the given handle.

Parameters **handle** – The handle of the chrome window.

focused_chrome_window_handle

Returns the currently focused chrome window handle.

Returns The *window handle* of the focused chrome window.

loaded (*handle*)

Check if the chrome window with the given handle has been completed loading.

Parameters **handle** – The handle of the chrome window.

Returns True, if the chrome window has been loaded.

classmethod register_window (*window_type*, *window_class*)

Registers a chrome window with this class so that this class may in turn create the appropriate window instance later on.

Parameters

- **window_type** – The type of window.
- **window_class** – The constructor of the window

switch_to (*target*)

Switches context to the specified chrome window.

Parameters **target** – The window to switch to. *target* can be a *handle* or a callback that returns True in the context of the desired window.

Returns Instance of the selected *BaseWindow*.

BaseWindow

class `firefox_puppeteer.ui.windows.BaseWindow` (*marionette*, *window_handle*)

Base class for any kind of chrome window.

close (*callback=None*, *force=False*)

Closes the current chrome window.

If this is the last remaining window, the Marionette session is ended.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current *BaseWindow* as parameter. Defaults to *window.open()*.
- **force** – Optional, forces the closing of the window by using the Gecko API. Defaults to *False*.

closed

Returns closed state of the chrome window.

Returns True if the window has been closed.

focus ()

Sets the focus to the current chrome window.

focused

Returns *True* if the chrome window is focused.

Returns True if the window is focused.

handle

Returns the *window handle* of the chrome window.

Returns *window handle*.

loaded

Checks if the window has been fully loaded.

Returns `True`, if the window is loaded.

localize_entity (*entity_id*)

Returns the localized string for the specified DTD entity id.

Parameters **entity_id** – The id to retrieve the value from.

Returns The localized string for the requested entity.

Raises **MarionetteException** – When entity id is not found.

localize_property (*property_id*)

Returns the localized string for the specified property id.

Parameters **property_id** – The id to retrieve the value from.

Returns The localized string for the requested property.

Raises **MarionetteException** – When property id is not found.

menubar

Provides access to the menu bar, for example, the **File** menu.

See the `MenuBar` reference.

open_window (*callback=None, expected_window_class=None, focus=True*)

Opens a new top-level chrome window.

Parameters

- **callback** – Optional, function to trigger the window to open. It is triggered with the current `BaseWindow` as parameter. Defaults to `window.open()`.
- **expected_class** – Optional, check for the correct window class.
- **focus** – Optional, if true, focus the new window. Defaults to `True`.

send_shortcut (*command_key, **kwargs*)

Sends a keyboard shortcut to the window.

Parameters

- **command_key** – The key (usually a letter) to be pressed.
- **accel** – Optional, If `True`, the *Accel* modifier key is pressed. This key differs between OS X (*Meta*) and Linux/Windows (*Ctrl*). Defaults to `False`.
- **alt** – Optional, If `True`, the *Alt* modifier key is pressed. Defaults to `False`.
- **ctrl** – Optional, If `True`, the *Ctrl* modifier key is pressed. Defaults to `False`.
- **meta** – Optional, If `True`, the *Meta* modifier key is pressed. Defaults to `False`.
- **shift** – Optional, If `True`, the *Shift* modifier key is pressed. Defaults to `False`.

switch_to (*focus=False*)

Switches the context to this chrome window.

By default it will not focus the window. If that behavior is wanted, the *focus* parameter can be used.

Parameters `focus` – If *True*, the chrome window will be focused.

Returns Current window as *BaseWindow* instance.

window_element

Returns the inner DOM window element.

Returns DOM window element.

AppInfo

The appinfo class is a wrapper around the `nsIXULAppInfo` interface in Firefox and provides access to a subset of its members.

AppInfo

class `firefox_puppeteer.api.appinfo.AppInfo` (*marionette*)

This class provides access to various attributes of AppInfo.

For more details on AppInfo, visit: https://developer.mozilla.org/en-US/docs/Mozilla/QA/Mozmill_tests/Shared_Modules/UtilsAPI/appInfo

Keys

Keys

class `firefox_puppeteer.api.keys.Keys` (*marionette*)

Proxy to marionette's keys with an "accel" provided for convenience testing across platforms.

ACCEL

ADD = `u'\ue025'`

ALT = `u'\ue00a'`

ARROW_DOWN = `u'\ue015'`

ARROW_LEFT = `u'\ue012'`

ARROW_RIGHT = `u'\ue014'`

ARROW_UP = `u'\ue013'`

BACK_SPACE = `u'\ue003'`

CANCEL = `u'\ue001'`

CLEAR = `u'\ue005'`

COMMAND = `u'\ue03d'`

CONTROL = `u'\ue009'`

DECIMAL = `u'\ue028'`

DELETE = `u'\ue017'`

DIVIDE = `u'\ue029'`

DOWN = u'\ue015'
END = u'\ue010'
ENTER = u'\ue007'
EQUALS = u'\ue019'
ESCAPE = u'\ue00c'
F1 = u'\ue031'
F10 = u'\ue03a'
F11 = u'\ue03b'
F12 = u'\ue03c'
F2 = u'\ue032'
F3 = u'\ue033'
F4 = u'\ue034'
F5 = u'\ue035'
F6 = u'\ue036'
F7 = u'\ue037'
F8 = u'\ue038'
F9 = u'\ue039'
HELP = u'\ue002'
HOME = u'\ue011'
INSERT = u'\ue016'
LEFT = u'\ue012'
LEFT_ALT = u'\ue00a'
LEFT_CONTROL = u'\ue009'
LEFT_SHIFT = u'\ue008'
META = u'\ue03d'
MULTIPLY = u'\ue024'
NULL = u'\ue000'
NUMPAD0 = u'\ue01a'
NUMPAD1 = u'\ue01b'
NUMPAD2 = u'\ue01c'
NUMPAD3 = u'\ue01d'
NUMPAD4 = u'\ue01e'
NUMPAD5 = u'\ue01f'
NUMPAD6 = u'\ue020'
NUMPAD7 = u'\ue021'
NUMPAD8 = u'\ue022'

```
NUMPAD9 = u'\ue023'  
PAGE_DOWN = u'\ue00f'  
PAGE_UP = u'\ue00e'  
PAUSE = u'\ue00b'  
RETURN = u'\ue006'  
RIGHT = u'\ue014'  
SEMICOLON = u'\ue018'  
SEPARATOR = u'\ue026'  
SHIFT = u'\ue008'  
SPACE = u'\ue00d'  
SUBTRACT = u'\ue027'  
TAB = u'\ue004'  
UP = u'\ue013'
```

Localization

Localization

class `firefox_puppeteer.api.l10n.L10n` (*marionette*)
An API which allows Marionette to handle localized content.

Deprecated since version 52.2.0: Use the localization module from `marionette_driver` instead.

The [localization](#) of UI elements in Gecko based applications is done via entities and properties. For static values entities are used, which are located in `.dtd` files. Whereby for dynamically updated content the values come from `.property` files. Both types of elements can be identified via a unique id, and the translated content retrieved.

localize_entity (*dtd_urls, entity_id*)

Returns the localized string for the specified DTD entity id.

To find the entity all given DTD files will be searched for the id.

Parameters

- **dtd_urls** – A list of dtd files to search.
- **entity_id** – The id to retrieve the value from.

Returns The localized string for the requested entity.

Raises `NoSuchElementException` – When entity id is not found in `dtd_urls`.

localize_property (*property_urls, property_id*)

Returns the localized string for the specified property id.

To find the property all given property files will be searched for the id.

Parameters

- **property_urls** – A list of property files to search.
- **property_id** – The id to retrieve the value from.

Returns The localized string for the requested entity.

Raises `NoSuchElementException` – When property id is not found in `property_urls`.

Places

The Places class provides low-level access for several bookmark and history related methods.

Places

class `firefox_puppeteer.api.places.Places` (*marionette*)

Low-level access to several bookmark and history related actions.

BookmarkFolders

alias of `bookmark_folders`

clear_plugin_data ()

Clear any kind of locally stored data from plugins.

get_all_urls_in_history ()

Retrieve any URLs which have been stored in the history.

get_folder_ids_for_url (*url*)

Retrieve the folder ids where the given URL has been bookmarked in.

Parameters `url` – URL of the bookmark

Returns List of folder ids

is_bookmark_star_button_ready ()

Check if the status of the star-button is not updating.

Returns True, if the button is ready

is_bookmarked (*url*)

Check if the given URL is bookmarked.

Parameters `url` – The URL to Check

Returns True, if the URL is a bookmark

remove_all_history ()

Remove all history items.

restore_default_bookmarks ()

Restore the default bookmarks for the current profile.

wait_for_visited (*urls, callback*)

Wait until all passed-in urls have been visited.

Parameters

- **urls** – List of URLs which need to be visited and indexed
- **callback** – Method to execute which triggers loading of the URLs

Security

The Security class gives access to various helper methods, which assist in working with certificates or accessing specific security related information.

Security

class `firefox_puppeteer.api.security.Security` (*marionette*)

Low-level access to security (SSL, TLS) related information.

get_address_from_certificate (*certificate*)

Retrieves the address of the organization from the certificate information.

The returned address may be *None* in case of no address found, or a dictionary with the following entries: *city*, *country*, *postal_code*, *state*, *street*.

Parameters *certificate* – A JSON object representing the certificate, which can usually be retrieved via the current tab: *self.browser.tabbar.selected_tab.certificate*.

Returns Address details as dictionary

get_certificate_for_page (*tab_element*)

The SSL certificate associated with the loaded web page in the given tab.

Parameters *tab_element* – The inner tab DOM element.

Returns Certificate details as JSON object.

get_domain_from_common_name (*common_name*)

Retrieves the domain associated with a page's security certificate from the common name.

Parameters *certificate* – A string containing the certificate's common name, which can usually be retrieved like so: *certificate['commonName']*.

Returns Domain as string

SoftwareUpdate

The SoftwareUpdate class provides helpers for update tests.

SoftwareUpdate

class `firefox_puppeteer.api.software_update.SoftwareUpdate` (*marionette*)

The SoftwareUpdate API adds support for an easy access to the update process.

ABI

Get the customized ABI for the update service.

Returns ABI version

active_update

Holds a reference to an *ActiveUpdate* object.

allowed

Check if the user has permissions to run the software update

Returns Status if the user has the permissions

build_info

Return information of the current build version

Returns A dictionary of build information

force_fallback()

Update the update.status file and set the status to 'failed:6'

get_formatted_update_url (*force=False*)

Retrieve the formatted AUS update URL the update snippet is retrieved from.

Parameters **force** – Boolean flag to force an update check

Returns The URL of the update snippet

get_update_snippet (*update_url*)

Retrieve contents of the update snippet.

Parameters **update_url** – URL to the update snippet

is_complete_update

Return true if the offered update is a complete update

Returns True if the offered update is a complete update

mar_channels

Holds a reference to a *MARChannels* object.

os_version

Returns information about the OS version

Returns The OS version

patch_info

Returns information of the active update in the queue.

staging_directory

Returns the path to the updates staging directory.

update_channel

Return the currently used update channel.

update_type

Returns the type of the active update.

update_url

Return the update URL used for update checks.

ActiveUpdate

class `firefox_puppeteer.api.software_update.ActiveUpdate` (*marionette*)

exists

Checks if there is an active update.

Returns True if there is an active update

get_patch_at (*patch_index*)

Use `nsIUpdate.getPatchAt` to return a patch from an update.

Returns JSON data for an `nsIUpdatePatch` object

patch_count

Get the patchCount from the active update.

Returns The patch count

selected_patch

Get the selected patch for the active update.

Returns JSON data for the selected patch

MARChannels

class `firefox_puppeteer.api.software_update.MARChannels` (*marionette*)

Class to handle the allowed MAR channels as listed in update-settings.ini.

class `MARConfigParser` (*defaults=None, dict_type=<class 'collections.OrderedDict'>, allow_no_value=False*)

INI parser which allows to read and write MAR config files.

Virtually identical to the original method, but delimit keys and values with '=' instead of '='

write (*fp*)

Write an .ini-format representation of the configuration state.

`MARChannels.add_channels` (*channels*)

Add additional MAR channels.

Parameters `channels` – A set of channel names to add

`MARChannels.channels`

The currently accepted MAR channels.

Returns A set of channel names

`MARChannels.remove_channels` (*channels*)

Remove MAR channels.

Parameters `channels` – A set of channel names to remove

UpdateChannel

Utils

The Utils class gives access to various helper methods.

Utils

class `firefox_puppeteer.api.utils.Utils` (**args, **kwargs*)

Low-level access to utility actions.

compare_version (*a, b*)

Compare two version strings.

Parameters

- **a** – The first version.
- **b** – The second version.

Returns -1 if a is smaller than b, 0 if equal, and 1 if larger.

permissions

Handling of various permissions for hosts.

Returns Instance of the Permissions class.

sanitize (*data_type*)

Sanitize user data, including cache, cookies, offlineApps, history, formdata, downloads, passwords, sessions, siteSettings.

Usage: `sanitize()`: Clears all user data. `sanitize({ "sessions": True })`: Clears only session user data.

more: <https://dxr.mozilla.org/mozilla-central/source/browser/base/content/sanitize.js>

Parameters `data_type` – optional, Information specifying data to be sanitized

CHAPTER 4

Indices and tables

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

A

- ABI (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 44
- AboutWindow (class in firefox_puppeteer.ui.about_window.window), 7
- ACCEL (firefox_puppeteer.api.keys.Keys attribute), 40
- active_update (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 44
- ActiveUpdate (class in firefox_puppeteer.api.software_update), 45
- ADD (firefox_puppeteer.api.keys.Keys attribute), 40
- add_channels() (firefox_puppeteer.api.software_update.MARChannels method), 46
- AddOnInstallBlockedNotification (class in firefox_puppeteer.ui.browser.notifications), 17
- AddOnInstallCompleteNotification (class in firefox_puppeteer.ui.browser.notifications), 18
- AddOnInstallConfirmationNotification (class in firefox_puppeteer.ui.browser.notifications), 18
- AddOnInstallFailedNotification (class in firefox_puppeteer.ui.browser.notifications), 19
- AddOnProgressNotification (class in firefox_puppeteer.ui.browser.notifications), 20
- all (firefox_puppeteer.ui.windows.Windows attribute), 37
- allow_button (firefox_puppeteer.ui.browser.notifications.AddOnInstallBlockedNotification attribute), 17
- allowed (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 44
- ALT (firefox_puppeteer.api.keys.Keys attribute), 40
- AppInfo (class in firefox_puppeteer.api.appinfo), 40
- apply (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
- ApplyPanel (class in firefox_puppeteer.ui.about_window.deck), 10
- ARROW_DOWN (firefox_puppeteer.api.keys.Keys attribute), 40
- ARROW_LEFT (firefox_puppeteer.api.keys.Keys attribute), 40
- ARROW_RIGHT (firefox_puppeteer.api.keys.Keys attribute), 40
- ARROW_UP (firefox_puppeteer.api.keys.Keys attribute), 40
- autocomplete_results (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 24
- AutocompleteResults (class in firefox_puppeteer.ui.browser.toolbars), 26

B

- back_button (firefox_puppeteer.ui.browser.toolbars.NavBar attribute), 23
- BACK_SPACE (firefox_puppeteer.api.keys.Keys attribute), 40
- BaseWindow (class in firefox_puppeteer.ui.windows), 38
- BookmarkFolders (firefox_puppeteer.api.places.Places attribute), 43
- BrowserWindow (class in firefox_puppeteer.ui.browser.window), 27
- build_info (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 44
- button (firefox_puppeteer.ui.about_window.deck.ApplyPanel attribute), 10
- button (firefox_puppeteer.ui.about_window.deck.CheckForUpdatesPanel attribute), 10
- button (firefox_puppeteer.ui.about_window.deck.DownloadAndInstallPanel attribute), 11
- buttons (firefox_puppeteer.ui.browser.tabbar.MenuPanel.MenuPanelElement attribute), 23

C

- CANCEL (firefox_puppeteer.api.keys.Keys attribute), 40
- certificate (firefox_puppeteer.ui.browser.tabbar.Tab attribute), 22
- channels (firefox_puppeteer.api.software_update.MARChannels attribute), 46
- check_for_updates (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9

CheckForUpdatesPanel (class in firefox_puppeteer.ui.about_window.deck), 10
 checking (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 31
 checking_for_updates (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
 CheckingForUpdatesPanel (class in firefox_puppeteer.ui.about_window.deck), 11
 CheckingPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 33
 CLEAR (firefox_puppeteer.api.keys.Keys attribute), 40
 clear() (firefox_puppeteer.ui.browser.toolbars.LocationBar method), 24
 clear_plugin_data() (firefox_puppeteer.api.places.Places method), 43
 click() (firefox_puppeteer.ui.browser.tabbar.MenuPanel.MenuPanelElement attribute), 27
 click() (firefox_puppeteer.ui.browser.tabbar.MenuPanel.MenuPanelElement method), 23
 close() (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 7
 close() (firefox_puppeteer.ui.about_window.window.AboutWindow method), 7
 close() (firefox_puppeteer.ui.browser.notifications.AddOnInstallBlockedNotification method), 17
 close() (firefox_puppeteer.ui.browser.notifications.AddOnInstallCompleteNotification method), 18
 close() (firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmationNotification method), 18
 close() (firefox_puppeteer.ui.browser.notifications.AddOnInstallFailedNotification method), 19
 close() (firefox_puppeteer.ui.browser.notifications.AddOnProgressNotification method), 20
 close() (firefox_puppeteer.ui.browser.tabbar.Tab method), 22
 close() (firefox_puppeteer.ui.browser.toolbars.AutoCompleteResults method), 26
 close() (firefox_puppeteer.ui.browser.toolbars.IdentityPopup method), 26
 close() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 27
 close() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow method), 13
 close() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog method), 30
 close() (firefox_puppeteer.ui.windows.BaseWindow method), 38
 close() (firefox_puppeteer.ui.windows.Windows method), 37
 close_all() (firefox_puppeteer.ui.windows.Windows method), 37
 close_all_tabs() (firefox_puppeteer.ui.browser.tabbar.TabBar method), 20
 close_button (firefox_puppeteer.ui.browser.notifications.AddOnInstallBlockedNotification attribute), 17
 close_button (firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmationNotification attribute), 18
 close_button (firefox_puppeteer.ui.browser.notifications.AddOnInstallFailedNotification attribute), 19
 close_button (firefox_puppeteer.ui.browser.notifications.AddOnProgressNotification attribute), 20
 close_button (firefox_puppeteer.ui.browser.tabbar.Tab attribute), 22
 close_context_menu() (firefox_puppeteer.ui.browser.toolbars.LocationBar method), 24
 close_tab() (firefox_puppeteer.ui.browser.tabbar.TabBar method), 20
 closed (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 7
 closed (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 27
 closed (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 13
 closed (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 30
 closed (firefox_puppeteer.ui.windows.BaseWindow attribute), 38
 COMMAND (firefox_puppeteer.api.keys.Keys attribute), 40
 compare_version() (firefox_puppeteer.api.utils.Utils method), 43
 connection_icon (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 24
 contextmenu (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 24
 CONTROL (firefox_puppeteer.api.keys.Keys attribute), 40
 create_window_instance() (firefox_puppeteer.ui.windows.Windows method), 37
 current (firefox_puppeteer.ui.windows.Windows attribute), 37
 Deck (class in firefox_puppeteer.ui.about_window.deck), 9
 Deck (class in firefox_puppeteer.ui.pageinfo.deck), 15
 deck (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 7
 deck (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 13
 default_homepage (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 27
 DELETE (firefox_puppeteer.api.keys.Keys attribute), 40

D

DIVIDE (firefox_puppeteer.api.keys.Keys attribute), 40
 domain (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 16
 DOWN (firefox_puppeteer.api.keys.Keys attribute), 40
 download_and_install (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
 download_failed (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
 DownloadAndInstallPanel (class in firefox_puppeteer.ui.about_window.deck), 11
 DownloadFailedPanel (class in firefox_puppeteer.ui.about_window.deck), 11
 downloading (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
 downloading (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 DownloadingPanel (class in firefox_puppeteer.ui.about_window.deck), 11
 DownloadingPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 33
 dummy (firefox_puppeteer.ui.update_wizard.wizard.Wizard element attribute), 32
 DummyPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 34

E

element (firefox_puppeteer.ui.about_window.deck.ApplyPanel attribute), 10
 element (firefox_puppeteer.ui.about_window.deck.CheckForUpdatesPanel attribute), 10
 element (firefox_puppeteer.ui.about_window.deck.CheckingForUpdatesPanel attribute), 11
 element (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
 element (firefox_puppeteer.ui.about_window.deck.DownloadAndInstallPanel attribute), 11
 element (firefox_puppeteer.ui.about_window.deck.DownloadFailedPanel attribute), 11
 element (firefox_puppeteer.ui.about_window.deck.DownloadInProgressPanel attribute), 11
 element (firefox_puppeteer.ui.browser.notifications.AddOnInstallCompleteNotification), 17
 element (firefox_puppeteer.ui.browser.notifications.AddOnInstallCompleteNotification attribute), 19
 element (firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmationNotification), 18
 element (firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmationNotification attribute), 19
 element (firefox_puppeteer.ui.browser.notifications.AddOnProgressNotification attribute), 20
 element (firefox_puppeteer.ui.deck.Panel attribute), 12
 element (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15
 element (firefox_puppeteer.ui.pageinfo.deck.PageInfoPanel attribute), 16
 element (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 16
 element (firefox_puppeteer.ui.update_wizard.wizard.CheckingPanel attribute), 33
 element (firefox_puppeteer.ui.update_wizard.wizard.DownloadingPanel attribute), 33
 element (firefox_puppeteer.ui.update_wizard.wizard.DummyPanel attribute), 34
 element (firefox_puppeteer.ui.update_wizard.wizard.ErrorExtraPanel attribute), 34
 element (firefox_puppeteer.ui.update_wizard.wizard.ErrorPanel attribute), 34
 element (firefox_puppeteer.ui.update_wizard.wizard.ErrorPatchingPanel attribute), 34
 element (firefox_puppeteer.ui.update_wizard.wizard.FinishedBackgroundPanel attribute), 35
 element (firefox_puppeteer.ui.update_wizard.wizard.FinishedPanel attribute), 35
 element (firefox_puppeteer.ui.update_wizard.wizard.IncompatibleCheckPanel attribute), 35
 element (firefox_puppeteer.ui.update_wizard.wizard.IncompatibleListPanel attribute), 36
 element (firefox_puppeteer.ui.update_wizard.wizard.ManualUpdatePanel attribute), 36
 element (firefox_puppeteer.ui.update_wizard.wizard.NoUpdatesFoundPanel attribute), 36
 element (firefox_puppeteer.ui.update_wizard.wizard.UpdatesFoundBasicPanel attribute), 37
 element (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 END (firefox_puppeteer.api.keys.Keys attribute), 41
 ENTER (firefox_puppeteer.api.keys.Keys attribute), 41
 EQUALS (firefox_puppeteer.api.keys.Keys attribute), 41
 DownloadAndInstallPanel (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 DownloadFailedPanel (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 DownloadInProgressPanel (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 ErrorExtraPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 34
 ErrorPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 34
 ErrorPatchingPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 34
 FinishedBackgroundPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 35
 FinishedPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 35
 IncompatibleCheckPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 35
 IncompatibleListPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 36
 ManualUpdatePanel (class in firefox_puppeteer.ui.update_wizard.wizard), 36
 NoUpdatesFoundPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 36
 UpdatesFoundBasicPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 37
 Wizard (class in firefox_puppeteer.ui.update_wizard.wizard), 32
 ESCAPE (firefox_puppeteer.api.keys.Keys attribute), 41
 exists (firefox_puppeteer.api.software_update.ActiveUpdate

attribute), 45

F

F1 (firefox_puppeteer.api.keys.Keys attribute), 41

F10 (firefox_puppeteer.api.keys.Keys attribute), 41

F11 (firefox_puppeteer.api.keys.Keys attribute), 41

F12 (firefox_puppeteer.api.keys.Keys attribute), 41

F2 (firefox_puppeteer.api.keys.Keys attribute), 41

F3 (firefox_puppeteer.api.keys.Keys attribute), 41

F4 (firefox_puppeteer.api.keys.Keys attribute), 41

F5 (firefox_puppeteer.api.keys.Keys attribute), 41

F6 (firefox_puppeteer.api.keys.Keys attribute), 41

F7 (firefox_puppeteer.api.keys.Keys attribute), 41

F8 (firefox_puppeteer.api.keys.Keys attribute), 41

F9 (firefox_puppeteer.api.keys.Keys attribute), 41

feed (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15

finished (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32

finished_background (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32

FinishedBackgroundPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 35

FinishedPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 35

focus() (firefox_puppeteer.ui.about_window.window.AboutWindow method), 7

focus() (firefox_puppeteer.ui.browser.toolbars.LocationBar method), 24

focus() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 27

focus() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow method), 13

focus() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog method), 30

focus() (firefox_puppeteer.ui.windows.BaseWindow method), 38

focus() (firefox_puppeteer.ui.windows.Windows method), 37

focused (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 8

focused (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 24

focused (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 27

focused (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 13

focused (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 30

focused (firefox_puppeteer.ui.windows.BaseWindow attribute), 38

focused_chrome_window_handle (firefox_puppeteer.ui.windows.Windows attribute), 38

force_fallback() (firefox_puppeteer.api.software_update.SoftwareUpdate method), 45

forward_button (firefox_puppeteer.ui.browser.toolbars.NavBar attribute), 23

G

general (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15

get_address_from_certificate() (firefox_puppeteer.api.security.Security method), 44

get_all_urls_in_history() (firefox_puppeteer.api.places.Places method), 43

get_certificate_for_page() (firefox_puppeteer.api.security.Security method), 44

get_contextmenu_entry() (firefox_puppeteer.ui.browser.toolbars.LocationBar method), 24

get_domain_from_common_name() (firefox_puppeteer.api.security.Security method), 44

get_final_url() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 27

get_folder_ids_for_url() (firefox_puppeteer.api.places.Places method), 43

get_formatted_update_url() (firefox_puppeteer.api.software_update.SoftwareUpdate method), 45

get_handle_for_tab() (firefox_puppeteer.ui.browser.tabbar.TabBar static method), 21

get_matching_text() (firefox_puppeteer.ui.browser.toolbars.AutocompleteResults method), 26

get_menu() (firefox_puppeteer.ui.menu.MenuBar method), 12

get_menu_by_id() (firefox_puppeteer.ui.menu.MenuBar method), 12

get_patch_at() (firefox_puppeteer.api.software_update.ActiveUpdate method), 45

get_update_snippet() (firefox_puppeteer.api.software_update.SoftwareUpdate method), 45

H

handle (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 8

handle (firefox_puppeteer.ui.browser.tabbar.Tab attribute), 22

handle (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 27

handle (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 14

handle (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 30

handle (firefox_puppeteer.ui.windows.BaseWindow attribute), 38

HELP (firefox_puppeteer.api.keys.Keys attribute), 41

history_drop_marker (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 24

HOME (firefox_puppeteer.api.keys.Keys attribute), 41

home_button (firefox_puppeteer.ui.browser.toolbars.NavBar attribute), 23

I

identity_box (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 24

identity_country_label (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25

identity_icon (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25

identity_organization_label (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25

identity_popup (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25

IdentityPopup (class in firefox_puppeteer.ui.browser.toolbars), 26

incompatible_check (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32

incompatible_list (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32

IncompatibleCheckPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 35

IncompatibleListPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 36

INSERT (firefox_puppeteer.api.keys.Keys attribute), 41

is_bookmark_star_button_ready() (firefox_puppeteer.api.places.Places method), 43

is_bookmarked() (firefox_puppeteer.api.places.Places method), 43

is_complete (firefox_puppeteer.ui.browser.toolbars.AutocompleteResults attribute), 26

is_complete_update (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 22

is_open (firefox_puppeteer.ui.browser.toolbars.AutocompleteResults attribute), 26

is_open (firefox_puppeteer.ui.browser.toolbars.IdentityPopup attribute), 26

is_private (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 27

items (firefox_puppeteer.ui.menu.MenuBar.MenuElement attribute), 12

K

Keys (class in firefox_puppeteer.api.keys), 40

L

L10n (class in firefox_puppeteer.api.l10n), 42

label (firefox_puppeteer.ui.browser.notifications.AddOnInstallBlockedNotification attribute), 17

label (firefox_puppeteer.ui.browser.notifications.AddOnInstallCompleteNotification attribute), 19

label (firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmationNotification attribute), 18

label (firefox_puppeteer.ui.browser.notifications.AddOnInstallFailedNotification attribute), 19

label (firefox_puppeteer.ui.browser.notifications.AddOnProgressNotification attribute), 20

LEFT (firefox_puppeteer.api.keys.Keys attribute), 41

LEFT_ALT (firefox_puppeteer.api.keys.Keys attribute), 41

LEFT_CONTROL (firefox_puppeteer.api.keys.Keys attribute), 41

LEFT_SHIFT (firefox_puppeteer.api.keys.Keys attribute), 41

load_url() (firefox_puppeteer.ui.browser.toolbars.LocationBar method), 25

loaded (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 8

loaded (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 27

loaded (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 14

loaded (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 30

loaded (firefox_puppeteer.ui.windows.BaseWindow attribute), 39

loaded() (firefox_puppeteer.ui.windows.Windows method), 38

localize_entity() (firefox_puppeteer.api.l10n.L10n method), 42

localize_entity() (firefox_puppeteer.ui.about_window.window.AboutWindow method), 8

localize_entity() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 28

localize_entity() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow method), 14

localize_entity() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 30
 localize_entity() (firefox_puppeteer.ui.windows.BaseWindow attribute), 39
 localize_entity() (firefox_puppeteer.ui.windows.BaseWindow.MenuBar.MenuElement (class in firefox_puppeteer.ui.menu)), 12
 localize_property() (firefox_puppeteer.api.110n.L10n.MenuPanel (class in firefox_puppeteer.ui.browser.tabbar)), 23
 localize_property() (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 21
 localize_property() (firefox_puppeteer.ui.about_window.window.AboutWindow.MenuPanel.MenuPanelElement (class in firefox_puppeteer.ui.browser.tabbar)), 23
 localize_property() (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 13
 localize_property() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 41
 localize_property() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow.MULTIPLY (firefox_puppeteer.api.keys.Keys attribute)), 41
 localize_property() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 30
 localize_property() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog.NavBar (class in firefox_puppeteer.ui.browser.toolbars)), 23
 localize_property() (firefox_puppeteer.ui.windows.BaseWindow attribute), 28
 location (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 22
 LocationBar (class in firefox_puppeteer.ui.browser.toolbars), 24
 locationbar (firefox_puppeteer.ui.browser.toolbars.NavBar attribute), 23
 locationbar (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 28
 newtab_button (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 21
 no_updates_found (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
 no_updates_found (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 manual_update (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 ManualUpdatePanel (class in firefox_puppeteer.ui.update_wizard.wizard), 36
 ManualUpdatePanel (class in firefox_puppeteer.ui.update_wizard.wizard.NoUpdatesFoundPanel (class in firefox_puppeteer.ui.update_wizard.wizard)), 36
 mar_channels (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 45
 MARChannels (class in firefox_puppeteer.api.software_update), 46
 MARChannels.MARConfigParser (class in firefox_puppeteer.api.software_update), 46
 media (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15
 menu_button (firefox_puppeteer.ui.browser.toolbars.NavBar attribute), 24
 MenuBar (class in firefox_puppeteer.ui.menu), 12
 menubar (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 8
 menubar (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 28
 menubar (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 14
 menubar (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 30
 notification_popup (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 28
 notification_popup (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25
 NoUpdatesFoundPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 36
 NULL (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD0 (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD1 (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD2 (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD3 (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD4 (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD5 (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD6 (firefox_puppeteer.api.keys.Keys attribute), 41
 NUMPAD7 (firefox_puppeteer.api.keys.Keys attribute), 41

- NUMPAD8 (firefox_puppeteer.api.keys.Keys attribute), 41
- NUMPAD9 (firefox_puppeteer.api.keys.Keys attribute), 41
- ## O
- open_about_window() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 28
- open_browser() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 28
- open_identity_popup() (firefox_puppeteer.ui.browser.toolbars.LocationBar method), 25
- open_page_info_window() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 28
- open_tab() (firefox_puppeteer.ui.browser.tabbar.TabBar method), 21
- open_window() (firefox_puppeteer.ui.about_window.window.AboutWindow method), 8
- open_window() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 29
- open_window() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow method), 14
- open_window() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog method), 31
- open_window() (firefox_puppeteer.ui.windows.BaseWindow method), 39
- origin (firefox_puppeteer.ui.browser.notifications.AddOnInstallBlockedNotification attribute), 17
- origin (firefox_puppeteer.ui.browser.notifications.AddOnInstallCompleteNotification attribute), 19
- origin (firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmationNotification attribute), 18
- origin (firefox_puppeteer.ui.browser.notifications.AddOnInstallFailedNotification attribute), 19
- origin (firefox_puppeteer.ui.browser.notifications.AddOnProgressNotification attribute), 20
- os_version (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 45
- owner (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 16
- ## P
- PAGE_DOWN (firefox_puppeteer.api.keys.Keys attribute), 42
- PAGE_UP (firefox_puppeteer.api.keys.Keys attribute), 42
- PageInfoPanel (class in firefox_puppeteer.ui.pageinfo.deck), 16
- PageInfoWindow (class in firefox_puppeteer.ui.pageinfo.window), 13
- Panel (class in firefox_puppeteer.ui.deck), 12
- panels (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
- panels (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15
- panels (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
- patch_count (firefox_puppeteer.api.software_update.ActiveUpdate attribute), 45
- patch_info (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 45
- PAUSE (firefox_puppeteer.api.keys.Keys attribute), 42
- permissions (firefox_puppeteer.api.utils.Utils attribute), 47
- permissions (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15
- Places (class in firefox_puppeteer.api.places), 43
- popup (firefox_puppeteer.ui.browser.tabbar.MenuPanel attribute), 23
- progress (firefox_puppeteer.ui.update_wizard.wizard.CheckingPanel attribute), 33
- progress (firefox_puppeteer.ui.update_wizard.wizard.DownloadingPanel attribute), 33
- progress (firefox_puppeteer.ui.update_wizard.wizard.IncompatibleCheckPanel attribute), 35
- ## R
- register_window() (firefox_puppeteer.ui.windows.Windows class method), 38
- remove_channels() (firefox_puppeteer.api.software_update.MARChannels method), 46
- restore_default_bookmarks() (firefox_puppeteer.api.places.Places method), 43
- restore_tabs_button (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 21
- restore_tabs_button_wrapper (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 21
- results (firefox_puppeteer.ui.browser.toolbars.AutocompleteResults attribute), 26
- RETURN (firefox_puppeteer.api.keys.Keys attribute), 42
- RIGHT (firefox_puppeteer.api.keys.Keys attribute), 42

S

- sanitize() (firefox_puppeteer.api.utils.Utils method), 47
 - Security (class in firefox_puppeteer.api.security), 44
 - security (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15
 - SecurityPanel (class in firefox_puppeteer.ui.pageinfo.deck), 16
 - select() (firefox_puppeteer.ui.browser.tabbar.Tab method), 22
 - select() (firefox_puppeteer.ui.menu.MenuBar method), 13
 - select() (firefox_puppeteer.ui.menu.MenuBar.MenuElement method), 12
 - select() (firefox_puppeteer.ui.pageinfo.deck.Deck method), 15
 - select_by_id() (firefox_puppeteer.ui.menu.MenuBar method), 13
 - select_by_id() (firefox_puppeteer.ui.menu.MenuBar.MenuElement method), 12
 - select_next_page() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog method), 31
 - selected (firefox_puppeteer.ui.browser.tabbar.Tab attribute), 23
 - selected_index (firefox_puppeteer.ui.about_window.deck.Deck attribute), 9
 - selected_index (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 21
 - selected_index (firefox_puppeteer.ui.browser.toolbars.AutoCompleteResults attribute), 26
 - selected_index (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15
 - selected_index (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 32
 - selected_panel (firefox_puppeteer.ui.about_window.deck.Deck attribute), 10
 - selected_panel (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 15
 - selected_panel (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 33
 - selected_patch (firefox_puppeteer.api.software_update.ActiveUpdate attribute), 46
 - selected_tab (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 21
 - SEMICOLON (firefox_puppeteer.api.keys.Keys attribute), 42
 - send_shortcut() (firefox_puppeteer.ui.about_window.window.AboutWindow method), 8
 - send_shortcut() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 29
 - send_shortcut() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow method), 14
 - send_shortcut() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog method), 31
 - send_shortcut() (firefox_puppeteer.ui.windows.BaseWindow method), 39
 - SEPARATOR (firefox_puppeteer.api.keys.Keys attribute), 42
 - SHIFT (firefox_puppeteer.api.keys.Keys attribute), 42
 - SoftwareUpdate (class in firefox_puppeteer.api.software_update), 44
 - SPACE (firefox_puppeteer.api.keys.Keys attribute), 42
 - staging_directory (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 45
 - stop_button (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25
 - SUBTRACT (firefox_puppeteer.api.keys.Keys attribute), 42
 - switch_to() (firefox_puppeteer.ui.about_window.window.AboutWindow method), 9
 - switch_to() (firefox_puppeteer.ui.browser.tabbar.Tab method), 23
 - switch_to() (firefox_puppeteer.ui.browser.tabbar.TabBar method), 22
 - switch_to() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 29
 - switch_to() (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow method), 14
 - switch_to() (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog method), 31
 - switch_to() (firefox_puppeteer.ui.windows.BaseWindow method), 39
 - switch_to() (firefox_puppeteer.ui.windows.Windows method), 38
- T**
- Tab (class in firefox_puppeteer.ui.browser.tabbar), 22
 - TAB (firefox_puppeteer.api.keys.Keys attribute), 42
 - tab (firefox_puppeteer.ui.pageinfo.deck.PageInfoPanel attribute), 16
 - tab (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 16
 - tab_element (firefox_puppeteer.ui.browser.tabbar.Tab attribute), 23
 - TabBar (class in firefox_puppeteer.ui.browser.tabbar), 20
 - tabbar (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 29
 - tabs (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 22
 - toolbar (firefox_puppeteer.ui.browser.tabbar.TabBar attribute), 22
 - toolbar (firefox_puppeteer.ui.browser.toolbars.NavBar attribute), 24
- U**
- UP (firefox_puppeteer.api.keys.Keys attribute), 42
 - UpdateWizardDialog (class in firefox_puppeteer.api.software_update), 45

update_type (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 45

update_url (firefox_puppeteer.api.software_update.SoftwareUpdate attribute), 45

updates_found_basic (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 33

UpdatesFoundBasicPanel (class in firefox_puppeteer.ui.update_wizard.wizard), 37

UpdateWizardDialog (class in firefox_puppeteer.ui.update_wizard.dialog), 30

urlbar (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25

urlbar_input (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25

Utils (class in firefox_puppeteer.api.utils), 46

V

value (firefox_puppeteer.ui.browser.toolbars.LocationBar attribute), 25

verifier (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 16

view (firefox_puppeteer.ui.browser.toolbars.IdentityPopup attribute), 27

view_certificate (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 16

view_cookies (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 16

view_passwords (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 17

visible_results (firefox_puppeteer.ui.browser.toolbars.AutocompleteResults attribute), 26

W

wait_for_notification() (firefox_puppeteer.ui.browser.window.BrowserWindow method), 29

wait_for_visited() (firefox_puppeteer.api.places.Places method), 43

window (firefox_puppeteer.ui.about_window.deck.ApplyPanel attribute), 10

window (firefox_puppeteer.ui.about_window.deck.CheckForUpdatesPanel attribute), 10

window (firefox_puppeteer.ui.about_window.deck.CheckingForUpdatesPanel attribute), 11

window (firefox_puppeteer.ui.about_window.deck.Deck attribute), 10

window (firefox_puppeteer.ui.about_window.deck.DownloadAndInstallPanel attribute), 11

window (firefox_puppeteer.ui.about_window.deck.DownloadFailedPanel attribute), 11

window (firefox_puppeteer.ui.about_window.deck.DownloadingPanel attribute), 12

window (firefox_puppeteer.ui.browser.notifications.AddOnInstallBlockedNotification attribute), 17

window (firefox_puppeteer.ui.browser.notifications.AddOnInstallCompletedNotification attribute), 19

window (firefox_puppeteer.ui.browser.notifications.AddOnInstallConfirmation attribute), 18

window (firefox_puppeteer.ui.browser.notifications.AddOnInstallFailedNotification attribute), 19

window (firefox_puppeteer.ui.browser.notifications.AddOnProgressNotification attribute), 20

window (firefox_puppeteer.ui.deck.Panel attribute), 12

window (firefox_puppeteer.ui.pageinfo.deck.Deck attribute), 16

window (firefox_puppeteer.ui.pageinfo.deck.PageInfoPanel attribute), 16

window (firefox_puppeteer.ui.pageinfo.deck.SecurityPanel attribute), 17

window (firefox_puppeteer.ui.update_wizard.wizard.CheckingPanel attribute), 33

window (firefox_puppeteer.ui.update_wizard.wizard.DownloadingPanel attribute), 33

window (firefox_puppeteer.ui.update_wizard.wizard.DummyPanel attribute), 34

window (firefox_puppeteer.ui.update_wizard.wizard.ErrorExtraPanel attribute), 34

window (firefox_puppeteer.ui.update_wizard.wizard.ErrorPanel attribute), 34

window (firefox_puppeteer.ui.update_wizard.wizard.ErrorPatchingPanel attribute), 34

window (firefox_puppeteer.ui.update_wizard.wizard.FinishedBackgroundPanel attribute), 35

window (firefox_puppeteer.ui.update_wizard.wizard.FinishedPanel attribute), 35

window (firefox_puppeteer.ui.update_wizard.wizard.IncompatibleCheckPanel attribute), 35

window (firefox_puppeteer.ui.update_wizard.wizard.IncompatibleListPanel attribute), 36

window (firefox_puppeteer.ui.update_wizard.wizard.ManualUpdatePanel attribute), 36

window (firefox_puppeteer.ui.update_wizard.wizard.NoUpdatesFoundPanel attribute), 36

window (firefox_puppeteer.ui.update_wizard.wizard.UpdatesFoundBasicPanel attribute), 37

window (firefox_puppeteer.ui.update_wizard.wizard.Wizard attribute), 33

window_element (firefox_puppeteer.ui.about_window.window.AboutWindow attribute), 9

window_element (firefox_puppeteer.ui.browser.window.BrowserWindow attribute), 29

window_element (firefox_puppeteer.ui.pageinfo.window.PageInfoWindow attribute), 15

window_element (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizard attribute), 30

attribute), 31

window_element (firefox_puppeteer.ui.windows.BaseWindow attribute), 40

Windows (class in firefox_puppeteer.ui.windows), 37

Wizard (class in firefox_puppeteer.ui.update_wizard.wizard), 31

wizard (firefox_puppeteer.ui.update_wizard.dialog.UpdateWizardDialog attribute), 31

write() (firefox_puppeteer.api.software_update.MARChannels.MARConfigParser method), 46