Capgemini

**Naming Convention**

**Information about this document**

|  |  |  |  |
| --- | --- | --- | --- |
| **Project name** |  | | |
| **Authors** | | | |
| **Name** | **Role** | **E-mail** | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
|  |  |  | |
| **History** | | | |
| **Name** | **Role** | **E-mail** | **Version** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Description 3](#_Toc453662714)

[2. Naming convention to create functions in page package 3](#_Toc453662715)

[3. Naming convention used in the code 3](#_Toc453662716)

[4. Name convention to create selector naming in page package 4](#_Toc453662717)

1. **Description**

A main idea is to write a specific function for a given element. Create names using template: actionElementNameElementType(), and for a verifying function: isElementNameElementTypePresent() or isElementNameElementTypeAction() To get a value of an another element attribute try using this schema: getElementNameElementTypeAttributeName();

We have a few basic web elements. There are items for a form: input, submit button, label. On a mage you can find button, checkbox, radio, dropdown list, tabs etc. On this element you can make an action: click, enter, submit, type, select, set, get, unset, expand, mouser over. You may also check a present element on a page.

**Important:** Visible and Present are not the same. First one means that a user sees this element. Present action means that element is in a HTML code but is not visible for a user (a css display attribute is set to none).

1. **Naming convention to create functions in page package**

Using this keyword you may create compilation of names. This method should be public. Every other method created onside this method should be private.

|  |  |  |
| --- | --- | --- |
| **Element** | **Action** | **Name (example)** |
| Form: Input text | enter | enterUsernameInput(username); |
| is (label) | isUsernameInputPresent(); |
| is (value) | isUsernameEmpty(); |
| get | getUsernameValue(); |
| Form: Label | get | getCashValue(); |
| is (value) | isCashValueEmpty(); |
| Is (label) | isCashLabelPresent(); |
| Form: Submit Button | submit | submitLoginForm(); |
| is | isLoginFormPresent(); |
| Page: Button | click | clickInfoButton(); |
| is | isInfoButtonPresent(); |
| Checkbox | set | setRememberMeCheckbox(); |
| unset | unsetRememberMeCheckbox(); |
| is (present) | isRememberMeCheckboxPresent(); |
| is (value) | isRememberMeCheckboxSet(); |
| Radio | set | setMaleRadioValue(“Woman”); |
| is (present) | isMaleRadioPresent(); |
| is (visible) | isMaleRadioVisible(); |
| get | getSelectedMaleValue(); |
| Elements( Tabs, Cards, Account etc. ) | click | clickPositionTab(); clickMyBilanceCard(); |
| is | isMyBilanceCardPresent(); |
| Dropdown List | select | selectAccountTypeValue(typeName); |
| unselect | unselectAccountTypeValue(typeName); |
| multiple select | SelectAccountTypesValues(List typeNames) |
| is (list) | isAccountTypeDropdownListPresent(); |
| is (element present) | isAccountTypeElementPresent(typeName); |
| is (element selected) | isAccountTypeSelected(typeName); |
| Link | click | clickMoreLink(); |
| is | isMoreLinkPresent(); |
| Combobox | select | selectSortComboboxPresent(); |
| is (present) | isSortComboboxContain(name); |
| is (contain) | isSortComboboxContain(name); |
| Element Attribute | get | getPositionTabCss(); |
| get | getMoreLinkHref(); getRememberMeCheckboxName(); |

1. **Naming convention used in the code**

Identifiers which are used in the code should named as follows:

* + packages: **com.<clientname>.<projectname>tests**
  + classes: **UpperCamelCase**
  + interfaces: **UpperCamelCase**
  + variables: **lowerCamelCase**
  + constants and final variables: **UPPERCASE\_SEPARATED\_\_BY\_UNDERSCORES**

Initialisms of three or more letters are CamelCase instead of UpperCase.

*Example:*

***parseDbmXmlFromIPAddress*** instead of ***parseDBMXMLFromIPAddress***

1. **Name convention to create selector naming in page package**

* selector variable must be **By** type
* selector variable must be **private final static**
* selector naming **selector<UserFieldDescription>**