

Selector Conventions (data-id-xxx)

Data-id-xxx Naming Conventions

- The data-id-xxx needs to be properly 'scoped'
- Use data-id-`scope`

Scopes

- center
- profile
- user
- group
- etc

Selector Naming Conventions

- Be as brief and descriptive as possible while avoiding name collision
- Selectors will still be contained in separate areas, which will mitigate name collisions
- Some selectors will still require a passed in parameter => ex: `nodeName`

`[elemType abbrev][name]` => ex: `chkProfileItemsRequired`

`data-id='[elemName]'` => ex: `data-id='chkProfileItemsRequired'`

Element Type Abbreviations

- `btn` - Button
- `chk` - Checkbox
- `rad` - Radio Button
- `txt` - TextBox
- `sel` - Select (ComboBox, Dropdown)
- `lab` - Label
- `lnk` - Link
- `img` - Image

Style guide

All Selectors

- Declare as static (unless there is a reason not to)
- Use `=>` notation instead of an assignment (i.e. `=`)
- Use `By.XPath()`, not `By.Css()`
- Use double quotes outside, single quotes inside (instead of escaped double quotes)

Parameterized Selectors

- Use string interpolation (i.e. `$"My name is {firstName}."` instead of `"My name is " + firstName + "."`)
- Use a normal function definition (i.e. `By Selector(firstName) => ...` instead of `Func<string, By> Selector => (firstName) => ...`)

Examples

Do

```
private static By SignInButtonSelector => By.XPath("//*[@contains(@data-id-login, 'btnSignIn')]");
```

```
public static By ContextMenuOptionSelector(string option) => By.XPath($"//*[@contains(@data-id-workflow, 'btnContextMenuOption')][contains(text(), '{option}')]");
```

Don't

```
private By SignInButtonSelector = By.Css("[data-id-login=\"btnSignIn\"]");
```

```
public Func<string, By> ContextMenuOptionSelector => (string option) => By.XPath("//*[@contains(@data-id-workflow, 'btnContextMenuOption')][contains(text(), '\" + option + '\"')]");
```

Revision #2

Created 18 July 2022 22:30:17 by Bryce Holloway

Updated 20 July 2022 13:54:29 by Bryce Holloway