

Command Usage Guide

A comprehensive guide to using all custom commands and assertion helpers in the Playwright POM Framework.

Click Element

```
await cmdClick(this.page, '#submit-button');
```

Fill Input Field

```
await cmdFill(this.page, '#email', 'user@example.com');
```

Press a Key

```
await cmdPress(this.page, '#input-box', 'Enter');
```

Check a Checkbox

```
await cmdCheck(this.page, '#accept-terms');
```

☐ Uncheck a Checkbox

```
await cmdUncheck(this.page, '#accept-terms');
```

Select Option from Dropdown

```
await cmdSelectOption(this.page, '#country-select', 'LK');  
await cmdSelectOption(this.page, '#multi-select', ['A', 'B']);
```

Hover Over Element

```
await cmdHover(this.page, '#profile-icon');
```



Scroll Into View

```
await cmdScrollIntoView(this.page, '#footer');
```



Get Text Content

```
const text = await cmdGetText(this.page, '#message');
```



Get Input Value

```
const value = await cmdGetValue(this.page, '#username');
```

Wait for Element to be Visible

```
await cmdWaitForVisible(this.page, '#modal');
```



Wait for Element to be Hidden

```
await cmdWaitForHidden(this.page, '#loading-spinner');
```



Check if Element is Visible

```
const isVisible = await cmdIsVisible(this.page, '#banner');
```



Check if Checkbox is Checked

```
const isChecked = await cmdIsChecked(this.page, '#terms');
```



Upload File

```
await cmdUploadFile(this.page, '#upload', 'path/to/file.pdf');
```



Click First Match

```
await cmdClickFirst(this.page, '.item');
```



Set Orientation (Mobile Testing)

```
await cmdSetOrientation(this.page, 'portrait');  
await cmdSetOrientation(this.page, 'landscape');
```



Capture Screenshot

```
await cmdCaptureScreenshot(this.page, 'Homepage Screenshot');
```



Set Viewport Size

```
await cmdSetViewportSize(this.page, 1440, 900);
```



Reset to Default Viewport

```
await cmdResetToDefaultViewport(this.page);
```



Capture & Check Network Errors

```
await cmdCaptureAndCheckNetworkErrors(this.page);
```



Log Local/Session Storage

```
await cmdLogStorage(this.page, 'localStorage');  
await cmdLogStorage(this.page, 'sessionStorage');
```



Is Element Disabled

```
const isDisabled = await cmdIsDisabled(this.page, '#submit');
```

Keyboard Typing (character by character)

```
await cmdKeyboardType(this.page, 'Hello World');
```

Keyboard Press (e.g., Enter, Tab)

```
await cmdKeyboardPress(this.page, 'Enter');
```

Keyboard Shortcut (e.g., Ctrl+A)

```
await cmdKeyboardShortcut(this.page, ['Control', 'A']);
```

Pause Execution

```
await cmdPause(this.page, 'Waiting for backend sync', 5000);
```

Write to Report

```
await writeToReport('Validation Step', 'Checked cart item count');
```

Select Mat Option (Custom Dropdown)

```
await cmdSelectMatOption(this.page, '#mat-select-status', 'mat-option span',  
'Active'); // specific  
await cmdSelectMatOption(this.page, '#mat-select-status', 'mat-option span',  
undefined, true); // random  
await cmdSelectMatOption(this.page, '#mat-select-status', 'mat-option span',  
undefined, true, ['None']); // random excluding 'None'
```

Get All Mat Option Values

```
const options = await getAllMatOptionValues(this.page, '#mat-select-status',  
'mat-option span');
```

Select Multiple Mat Options (Multi-Select)

```
await cmdSelectMatMultiOptions(this.page, '#mat-select-status', 'mat-option span', ['Value 1', 'Value 2']);
await cmdSelectMatMultiOptions(this.page, '#mat-select-status', 'mat-option span', [], true, 2); // random 2 options
await cmdSelectMatMultiOptions(this.page, '#mat-select-status', 'mat-option span', [], true, 2, ['None']); // random 2 options excluding 'None'
```

Utility: Random Data Generator (generateAndStore.ts)

```
import {
  generateAutoMixed,
  generateAutoLetterSpecial,
  generateOnlySpecial,
  generateOnlyNumbers
} from '@utils/generateAndStore';
```

Mixed Characters with Prefix

```
const value = await generateAutoMixed('username', 10); // "AutoA1b2C3!@"
```

Letters + Special Characters with Prefix


```
const value = await generateAutoLetterSpecial('name', 8); // "AutoAb#_C$"
```

Only Special Characters

```
const value = await generateOnlySpecial('symbolKey', 5); // "!@#_~"
```

Only Numbers

```
const value = await generateOnlyNumbers('numberKey', 6); // "948301"
```

 All values are automatically saved into `data/admin/generate.json` with their key for future reuse.

Assertion Helpers

Assert Visible

```
await assertVisible(this.page, '#element');
```

Assert Hidden

```
await assertHidden(this.page, '#element');
```

Assert Text Equals

```
await assertTextEquals(this.page, '#title', 'Dashboard');
```

Assert Text Contains

```
await assertTextContains(this.page, '#summary', 'completed');
```

Assert Value Equals

```
await assertValueEquals(this.page, '#username', 'admin');
```

Assert Value Contains

```
await assertValueContains(this.page, '#description', 'error');
```

Assert Checked

```
await assertChecked(this.page, '#remember');
```

Assert Enabled

```
await assertEnabled(this.page, '#submit');
```

Assert Disabled

```
await assertDisabled(this.page, '#locked');
```

Assert URL Contains

```
await assertUrlContains(this.page, '/dashboard');
```

Assert Title Equals

```
await assertTitleEquals(this.page, 'Welcome Page');
```

Assert Text Equals Directly

```
await assertTextEqualsDirectly(this.page, actualText, 'Expected Text');
```

Use this guide to speed up development and ensure consistency when writing automation scripts.