Using Playwright Basic Actions

Andrejs Doronins

Overview

Navigating and its options

A great variety of clicking

Fill, check, select, press

Verifying the result

```
navigate("url");


navigate("url", new Page.NavigateOptions()
                          .setTimeout(millis) // 0 to disable
                          .setWaitUntil( DOMCONTENTLOADED ));
                                         LOAD
                                         NETWORKIDLE
```

Navigation Life Cycle

Navigation:

- Ends when response headers parsed

Loading (shortened):

- document content is loaded over network and parsed
- Page.onDOMContentLoaded(handler) event is fired
- page executes some scripts and loads resources like stylesheets and images
- Page.onLoad(handler) event is fired
- page executes dynamically loaded scripts
- networkidle is fired when no new network requests are made for 500 ms

Fastest

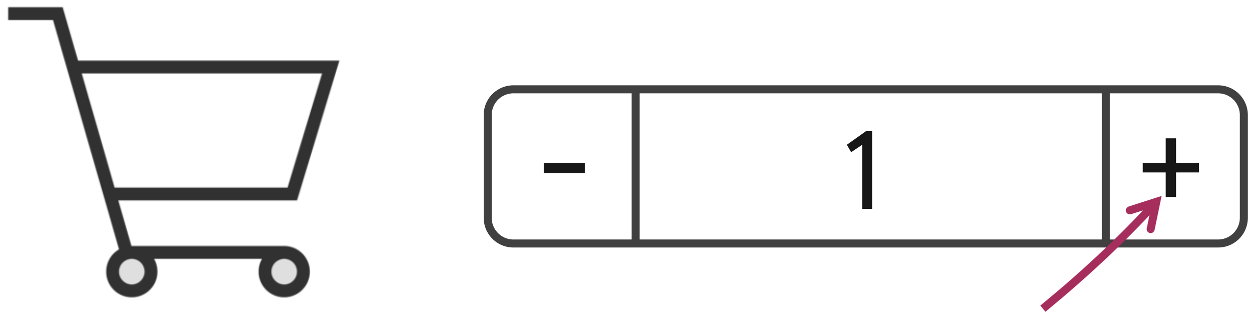
Balanced

Slowest

page.goBack() page.reload() page.goForward()

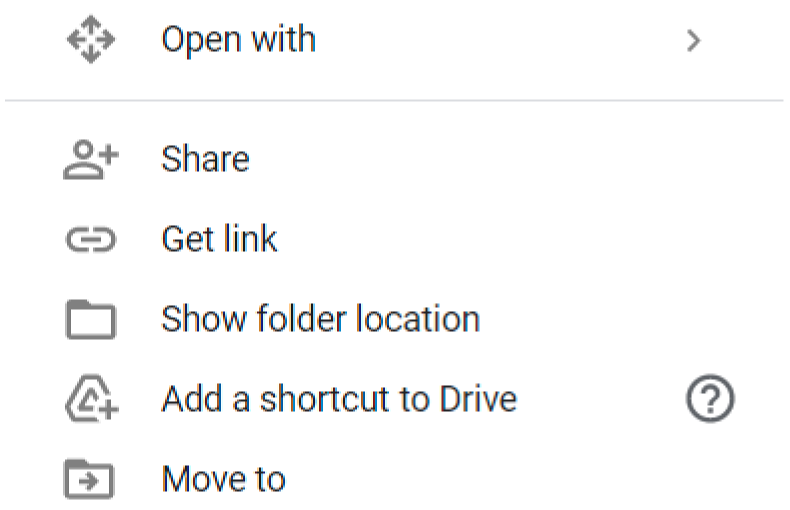
```
click("selector");


click("selector", new Page.ClickOptions()
                           .setClickCount(num));
```

Predictable and consistent patterns in a public API indicate a well-thought-out design



```
page.click("id=plus-sign");
page.click("id=plus-sign");
page.click("id=plus-sign");
[...]
```

```java
page.click("#selector", new Page.ClickOptions()
                                    .setButton(MouseButton.RIGHT));

page.click("#selector", new Page.ClickOptions()
                .setModifiers(Arrays.asList(KeyboardModifier.SHIFT)));

page.dblclick("#selector");
```



Open with ›

Share

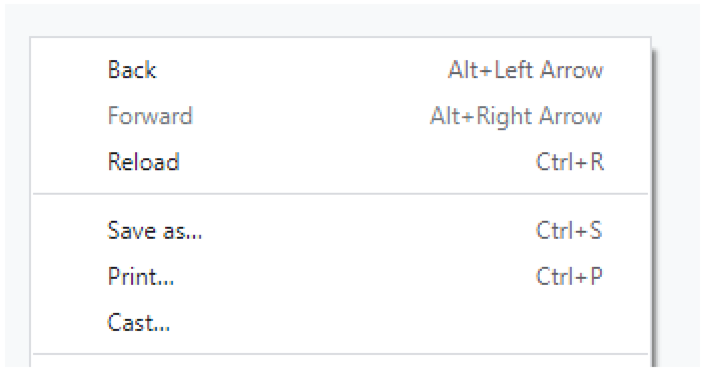
Get link

Show folder location

Add a shortcut to Drive ?

Move to

✓



Back	Alt+Left Arrow
Forward	Alt+Right Arrow
Reload	Ctrl+R
Save as...	Ctrl+S
Print...	Ctrl+P
Cast...	

✗

```
page.fill("#selector", "your text");
```

```java
// doesn’t bypass how browsers interpret captured events
page.click("selector", new Page.ClickOptions().setForce(true));
```

Filename.here

```java
 /**
   * Whether to bypass the actionability checks. Defaults to false.
   */
public Boolean force;


public ClickOptions setForce(boolean force) {
     this.force = force;
     return this;
}
```

Common methods:

setTimeout()
setForce()

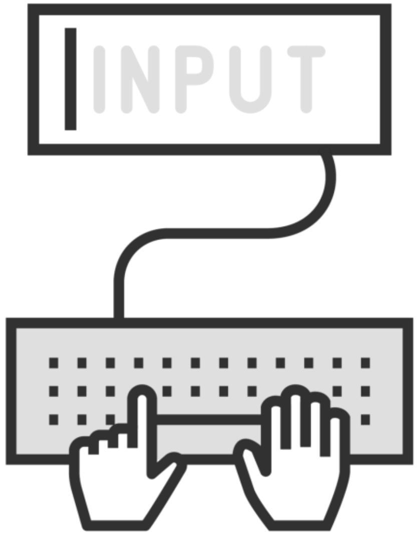
```
page.navigate("url", new Page.NavigateOptions());

page.click("#selector", new Page.ClickOptions());

page.fill("#selector", "your text“, new Page.FillOptions());

// etc.
```

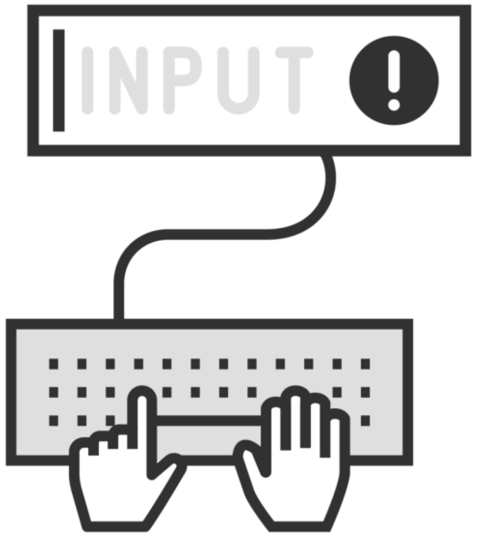
```
page.check("selector");

page.check("selector"); // does nothing


page.uncheck("selector");
```

Text

```
<option value="Bored">I'm just bored</option>
```

Attribute value 👍


INPUT


INPUT

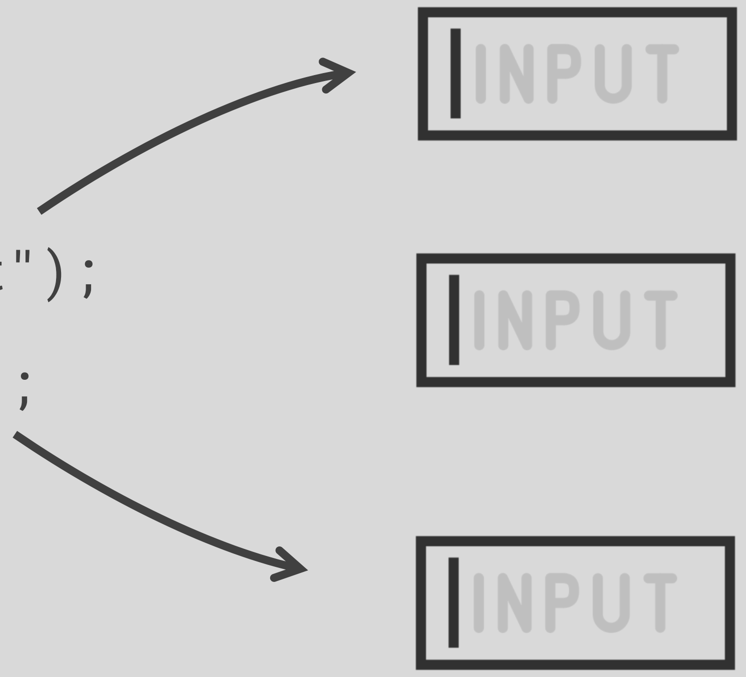
```java
Keyboard kb = page.keyboard();

kb.press("m");

kb.press("Backspace");

kb.press("ArrowDown");
```

```
String btn = "#selector";
```

```
page.click("#selector");

// some code

page.click("#selector");
```

Captures the logic to retrieve the element

```
Locator input = page.locator("#selector");
input.fill ("hi!");


input.first().fill("first");
input.last().fill("last");
input.nth(2).fill("second");
input.first().fill("first");

input.last().fill("last");
```



```
 <div>
    Text here
    <p>And here</p>
    <span style="display:none;">hidden</span>
    <img src="img.jpb" alt="image">
</div>
```

$\longrightarrow$ .innerHTML()

```
 <div>
   Text here
   <p>And here</p>
   <span style="display:none;">hidden</span>
   <img src="img.jpb" alt="image">
</div>
```

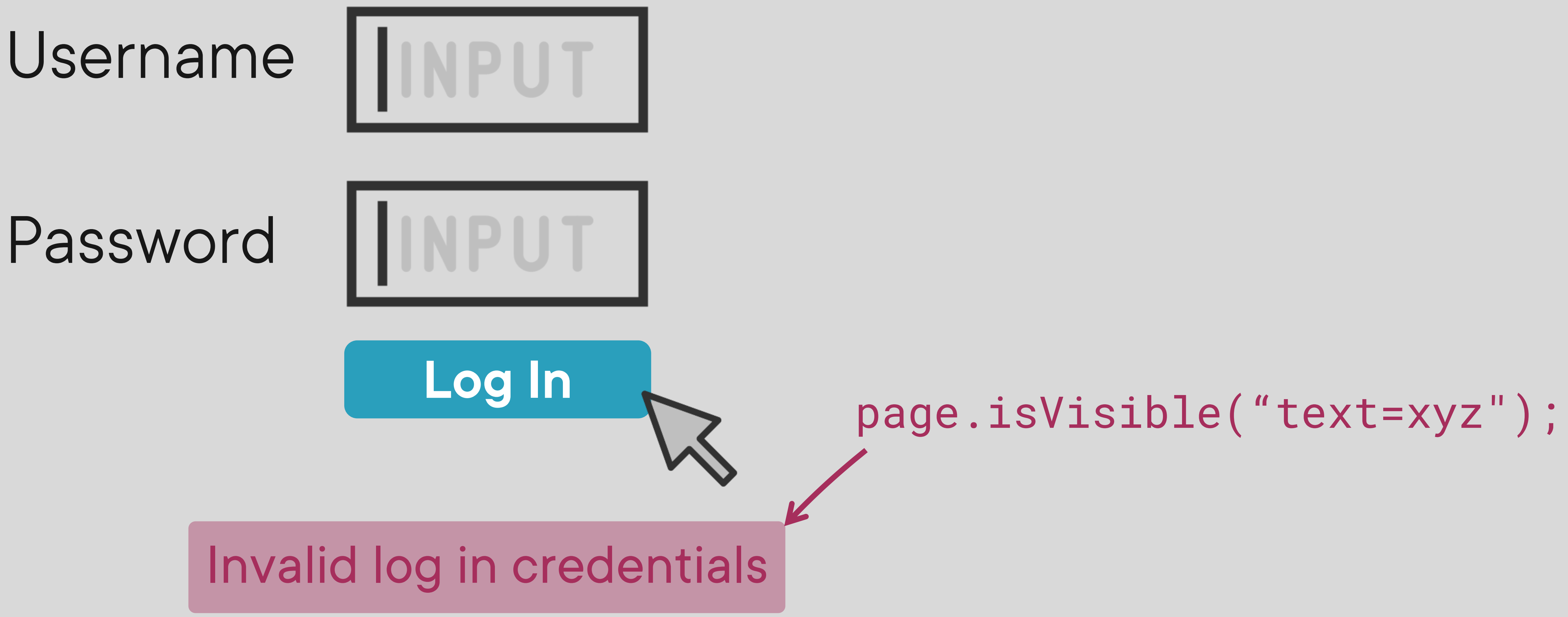
.innerText()

```
<div>
  Text here
  <p>And here</p>
  <span style="display:none;">hidden</span>
  <img src="img.jpb" alt="image">
</div>
```

$\longrightarrow$ .textContent()

```
 <div>
    Text here
    <p>And here</p>
    <span style="display:none;">hidden</span>
    <img src="img.jpb" alt="image">
</div>
```

$\xrightarrow{}$ `.getAttribute("img", "alt")`



```
page.isVisible("text=xyz");
```

```
assertTrue(   page.isVisible();
              page.isChecked();
              page.isDisabled();      )
              page.isEnabled();
              page.isHidden();
              page.isEditable();
```

```java
page.isVisible(new Page.IsVisibleOptions());

page.isChecked(new Page.IsCheckedOptions());

page.isDisabled(new Page.IsDisabledOptions());

page.isEnabled(new Page.IsEnabledOptions());

page.isHidden(new Page.IsHiddenOptions());

page.isEditable(new Page.IsEditableOptions());
```

```java
setStrict(true);
setTimeout(2_000);
```

Summary

Navigate, reload, goBack, goForward

Clicking with options, double-clicking

Filling, checking, selecting

Configurable

page.keyboard()

Locator

Text getters and element state checks

Up Next:

Learning Advanced Actions