

Extensibility

Custom selector engines

Playwright supports custom selector engines, registered with `selectors.register()`.

Selector engine should have the following properties:

- `create` function to create a relative selector from `root` (root is either a `Document`, `ShadowRoot` or `Element`) to a `target` element.
- `query` function to query first element matching `selector` relative to the `root`.
- `queryAll` function to query all elements matching `selector` relative to the `root`.

By default the engine is run directly in the frame's JavaScript context and, for example, can call an application-defined function. To isolate the engine from any JavaScript in the frame, but leave access to the DOM, register the engine with `{contentScript: true}` option. Content script engine is safer because it is protected from any tampering with the global objects, for example altering `Node.prototype` methods. All built-in selector engines run as content scripts. Note that running as a content script is not guaranteed when the engine is used together with other custom engines.

Selectors must be registered before creating the page.

An example of registering selector engine that queries elements based on a tag name:

Sync **Async**

```
tag_selector = ""
// Must evaluate to a selector engine instance.
{
  // Returns the first element matching given selector in the root's subtree.
  query(root, selector) {
    return root.querySelector(selector);
  },

  // Returns all elements matching given selector in the root's subtree.
  queryAll(root, selector) {
```

```
        return Array.from(root.querySelectorAll(selector));
    }
}"""
```

```
# register the engine. selectors will be prefixed with "tag=".
playwright.selectors.register("tag", tag_selector)
```

```
# now we can use "tag=" selectors.
button = page.locator("tag=button")
button.click()
```

```
# we can combine it with built-in locators.
page.locator("tag=div").get_by_text("click me").click()
```

```
# we can use it in any methods supporting selectors.
button_count = page.locator("tag=button").count()
```