

# kotlenide (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-08-31 23:22:20 UTC

# Table of Contents

1. Usage (TODO).....	2
2. How to (work in progress) .....	3
3. Links .....	5

# KOTLENIDE



*Idiomatic UI tests*

# Chapter 1. Usage (TODO)

*At the moment, my goal is implement idiomatic kotlin DSL for UI ent-to-end testing, which under the hood is using famous good known [selenide](#) functionality:*

```
class AwesomeKotlenideTests {
    @Test fun `selenide with idiomatic kotlin should be awesome`() {
        open("http://127.0.0.1:8080") {
            find("form") {
                it.find("input[type='text']") {
                    it sendKeys "hola!"
                }
                it.find("button") {
                    it press enter
                }
            }
        }
    }
}
```

## Chapter 2. How to (work in progress)

*java:* `Selenide.open(String relativeOrAbsoluteUrl)`

```
@Test
@DisplayName("should open method with String argument")
void test() {
    open("https://google.com")
    // ...
    close()
}
```

*kotlin:* `String.open { ... }`

```
@Test fun `should open String extension function`() {
    "https://google.com".open {
        // ...
        close()
    }
}

// or:

@Test fun `should open Pair extension function with free first key name contract`()
{
    ("url" to "https://google.com").open {
        // ...
        close()
    }
}
```

*java:* `Selenide.open(URL absoluteUrl)`

```
@Test
@DisplayName("should open method with URL argument")
void test() {
    open(new URL("https://ya.ru"))
    // ...
    close()
}
```

*kotlin:* `URL.open { ... }`

```
@Test fun `should open URL extension function`() {
    URL("https://ya.ru").open {
        // ...
        close()
    }
}
```

*java:* `Selenide.open(String relativeOrAbsoluteUrl, String domain, String login, String password)`

```
@Test
@DisplayName("should open using string arguments by some contract")
void test() {
    open("https://blog.example.com?q=java", "example.com", "login", "password")
    // ...
    close()
}
```

*kotlin:* `Map<String, String>.open { ... }`

```
@Test fun `should open Map extension function minimal contract`() {
    mapOf("relativeOrAbsoluteUrl" to "https://google.com").open {
        // ...
        close()
    }
}

// or:

@Test fun `should open Map extension function full contract`() {
    mapOf(
        "relativeOrAbsoluteUrl" to "https://blog.example.com?q=java",
        "domain" to "example.com",
        "login" to "login",
        "password" to "password"
    ) {
        // ...
        close()
    }
}
```

TODO: implement documentation...

# Chapter 3. Links

- [GitHub repo](#)
- [GitHub pages](#)