

# kotlenide (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-09-10 01:48:53 UTC

# Table of Contents

1. Introduction . . . . .	2
2. How to (work in progress) . . . . .	3
3. Links . . . . .	6
4. License . . . . .	7

# KOTLENIDE



*Idiomatic UI tests*

# Chapter 1. Introduction

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

```
class AwesomeKotlenideTests {
    @Test fun `selenide using idiomatic kotlin should be awesome`() {
        "https://google.com".open {
            find("body") {
                this contains "G"
                this has "mail"
            }
            close()
        }
    }

    @Test fun `should open given URL in chrome using String extension function and
    assert by text condition`() {
        "https://google.com".openIn("chrome") {
            find("body") {
                this.find(".ololo") shouldNotBe exist
                find(".trololo") shouldBeNot visible
                this shouldHaveText "Gmail"
            }
            close()
        }
    }
}
```

# Chapter 2. How to (work in progress)

## open url (String)

*java*

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

`open(String relativeOrAbsoluteUrl)`

*kotlin*

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

`String.open { ... }`

*kotlin*

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

`Pair.open { ... }`

## open absolute url (URL)

*java*

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

`open(URL absoluteUrl)`

*kotlin*

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

`URL.open { ... }`

**open by ...**

*java*

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

`open(String relativeOrAbsoluteUrl, String domain, String login, String password)`

```
@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"
    ).open {
        // ...
        close()
    }
}
```

`Map<String, String>.open { ... }`

# Chapter 3. Links

- [Kotlin](#)
- [Kotlin DSL guide](#)
- [Kotlin infix functions](#)
- [Selenide](#)
- [Awesome free online images editor LunaPic](#)
- [GitHub repo](#)
- [GitHub pages](#)



# Chapter 4. License

MIT