



# selenium-webdriver

Free unaffiliated eBook created from **Stack Overflow contributors.** 

#selenium-

webdriver

	1
	J
1: selenium-webdriver	2
	2
	2
Exemples	- -
Examples	<u> </u>
······	2
Visual Studio[NuGet]	2
Selenium WebDriver	3
Java	4
2: HTML	7
	7
	-
Examples	ſ
	7
Allure	Э
Maven	)
	9
	S
	Ś
	1
	)
3: Java@FindBy13	3
	3
1:	R
Evenned	۔ م
Examples	2
13	3
4: JavaSelenium Webdriver	5
	5
Examples	5
JavaSelenium WebdriverURL	5
to 1	5
	` ~
5: PythonKubyJavascriptSelenium-webdriverCl	5
	3

Examples
Selenium PythonUnittest2CircleCI
6: Selenium e2e
Examples
TestNG
testng.xml
Maven
7: Selenium-WebDriver
Examples
Python
8: Selenium
Examples
Python
9: Web
Examples
C
Java
10: Web
Examples
WebDriver
ID

CSS	
XPath	29
JavaScript	29
[C]	
11:	
Examples	
C	
Java	
	34
C	34
10.	
12:	
Examples	
Java	
C	
Python	
13:	
Examples	
JAVA	

C
14:
Examples
Java41
Java
C42
C
15:
Examples
JAVA
16. 46
10. //
Examples
Python
java
1/:
Examples
18:

Examples	
Selenium GridJava	
Selenium Grid	
Json	
19:	
Examples	
APIJAVAKeypress	
APIJAVA	59
20:	61
Examples	
Navigate[C]	61
Navigate[Java]	62
WebDriver	
21: /	
Examples	
JAVA	65

Examples
C
Python
Python
23:
Examples
Java7 <sup>.</sup>
C72
24: Javascript
Examples
C74
Python
Java
Java.       74         74       74         25:       76         Examples.       76         PhantomJS [C].       76         SimpleBrowser [C].       76         Java.       77         HTMLUnitDriver.       77
Java
Java

EventFiringWebDriver
27: Selenium Webdriver
Examples
C
Python
Java
Java
28:
Examples
Selenium WebDriver
Ajax
C
29:
Examples

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: selenium-webdriver

It is an unofficial and free selenium-webdriver ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official selenium-webdriver.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

## 1: selenium-webdriverをいめる

このセクションでは、selenium-webdriverのと、なぜがそれをいたいのかをします。

また、セレン・ウェブ・ドライバーのきなテーマについてもし、するトピックにリンクするがあります。 selenium-webdriverのドキュメントはしいので、これらのトピックのバージョンをするがあります。

バージョン



## **Examples**

インストールまたはセットアップ

WebDriverのをするには、Seleniumサイト Selenium HQ Downloadsからするドライバをするがあ ります。ここから、WebDriverをしようとしているブラウザやプラットフォームにするドライバ をダウンロードするがあります。たとえば、Chromeでテストしているは、Seleniumサイトから のにします。

https://sites.google.com/a/chromium.org/chromedriver/

chromedriver.exeをダウンロードするには

に、WebDriverをするに、するバインディングをダウンロードするがあります。たとえば、Cをしているは、Selenium HQ Downloadsページからダウンロードして、な.dllファイルをするか、 Visual StudioでパッケージとしてダウンロードしますNuGetパッケージマネージャー。

WebDriverのいについては、なファイルをダウンロードするがあります。そのの $_{selenium}$ webdriverマニュアルをしてください。

## Visual Studio⑦[NuGet]

Selenium WebDriverをインストールするもなは、NuGetパッケージマネージャをすることです。

あなたのプロジェクトでは、 "References"をクリックして、 "Manage NuGet Packages"をクリックしてください



に、ボックス [webdriver]にします。のようなものがされます。

NuGet: CSharpTest 🤕 🗙 DataManagerTests.cs	Testing.cs	Program.cs*
Browse Installed Updates 2		
webdriver	× • 🙆	Include prerelease
Selenium.WebDriver by Selenium C .NET bindings for the Selenium WebDriver Selenium.WebDriver.ChromeDri	Committers, <b>1,4N</b> · API · <b>Ver</b> by jsakamo	<b>M</b> downloads oto, <b>220K</b> downloads
Selenium Google Chrome Driver (Win32) (	does not make y	your source repository to fat.)
Support classes for the .NET bindings of the	nmitters, <b>1,13M</b> c ne Selenium Web	downloads bDriver API

"Selenium.WebDriver "と "Selenium.Support " $\forall \pi$ - $\wedge \pi$ ,  $\forall \mu \nu$ ) ( $\forall \mu$ ) (( $\forall \mu$ )) ((\forall \mu)) (\forall \mu))(( $\forall \mu$ )) ((\forall \mu))(( $\forall \mu$ )) ((\forall \mu)) ((\forall \mu))(((\forall \mu))

に、したいWebDriversをインスト―ルできます。たとえば、のいずれかです。

- Selenium.WebDriver.ChromeDriverGoogle Chrome
- PhantomJS ヘッドレス
- •

Selenium WebDriverとはですか

Seleniumは、ブラウザをするためにされたのツールです。これは、のプラットフォームにわたる Webアプリケーションテストによくされます。 Selenium WebDriverSelenium RC、Selenium IDE、Selenium Gridなど、Seleniumののでできるツールがいくつかあります。

WebDriverは、WebページのDOMをしたり、ユーザーエージェントのをすることができるリモー トコントロールインターフェイスです。このインタフェースは、のようなさまざまなプラットフ ォームにされたのワイヤプロトコルをします。

- GeckoDriver Mozilla Firefox
- ChromeDriver Google Chrome
- SafariDriver Apple Safari
- InternetExplorerDriver MS InternetExplorer
- MicrosoftWebDriver、EdgeDriver MSエッジ
- OperaChromiumDriver Operaブラウザ

のとに

- EventFiringWebDriver
- HtmlUnitDriver
- PhantomJSDriver
- RemoteWebDriver

Selenium WebDriverは、さまざまなのオブジェクトAPIをするSeleniumツールの1つで、なソフトウェアのとをにします。ユーザーがWebアプリケーションとやりとりするをにシミュレートするため、「JavaScriptのイベントの」にして「ネイティブOSレベルのイベント」がされます。

するリンク

- http://www.seleniumhq.org/
- http://www.aosabook.org/en/selenium.html
- https://www.w3.org/TR/webdriver/

Javaのインスト―ルまたはセットアップ

Selenium WebdriverとJavaをしてプログラミングとしてテストをくには、Selenium Webdriverの JARファイルをSeleniumのWebサイトからダウンロードするがあります。

Selenium webdriverのJavaプロジェクトをするにはのがありますが、もなの1つはMavenをすることです。 MavenはすべてのをむSelenium webdriverになJavaバインディングをダウンロードします。もう1つのは、JARファイルをダウンロードしてプロジェクトにインポートすることです。

Maven  $\mathcal{E}$  C Selenium Webdriver  $\mathcal{I}$   $\mathcal{D}$   $\mathcal$ 

- 1. このドキュメントののウィンドウにmavenをインストールしてください https: //maven.apache.org/install.html
- 2. が<sub>selenium-learing</sub>あるフォルダをする
- 3. が<sub>pom.xml</sub>のテキストエディタをして、のフォルダにファイルをします。
- 4. コンテンツをpom.xmlコピーする

```
<?xml version="1.0" encoding="UTF-8"?>
    <project xmlns="http://maven.apache.org/POM/4.0.0"</pre>
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
            <modelVersion>4.0.0</modelVersion>
            <groupId>SeleniumLearning</groupId>
            <artifactId>SeleniumLearning</artifactId>
            <version>1.0</version>
            <dependencies>
                <dependency>
                    <groupId>org.seleniumhq.selenium</groupId>
                    <artifactId>selenium-learning</artifactId>
                    <version>3.0.0-beta1</version>
                </dependency>
            </dependencies>
    </project>
```

でしたバージョンがバージョンであることをしてください。バージョンは http://docs.seleniumhq.org/download/maven.jspからできます。

5. コマンドラインをして、のコマンドをプロジェクトディレクトリにします。

mvn clean install

のコマンドは、なをすべてダウンロ―ドし、プロジェクトにします。

6. のコマンドをいて、Eclipse IDEにインポートできるEclipseプロジェクトをします。

mvn eclipse:eclipse

7. Eclipse IDEにプロジェクトをインポートするには、のにいます

Elipse - >ファイル - >インポート - > - >のプロジェクトをワークスペースに - >ヘ - >ブ ラウズ - > pom.xmlをむフォルダをつける - > OK - >

プロジェクトをクリックし、Maven - > Enable Dependency Managementをして、m2eclipseプラ グインをインストールします。

JarファイルをしてSelenium Webdriverプロジェクトをする

1. のにって、Eclipseでプロジェクトをします。

Elipseをく - >ファイル - > - > Javaプロジェクト - >をするセレンの - >

2. http://www.seleniumhq.org/download/からjarファイルをダウンロードしてください。

**Selenium Standalone Server**と**Selenium ClientWebDriver**ののバインディングをダウンロードするがあります。このはJavaについてしているので、Javaのセクションからjarファイルだけをダウンロードするがあります。のスクリーンショットをてください。

#### Selenium Client & WebDriver Language Bindings

In order to create scripts that interact with the Selenium Server (Selenium RC, Selenium Remote WebDriver) or create local Selenium WebDriver scripts, you need to make use of language-specific client drivers. These languages include both 1.x and 2.x style clients.

While language bindings for <u>other languages exist</u>, these are the core ones that are supported by the main project hosted on google code.

Language	Client Version	Release Date	$\frown$		
Java	3.0.0-beta1	2016-07-28	Download	Change log	Javadoc
C#	2.53.1	2016-06-28	Download	Change log	API docs
Ruby	3.0.0.beta1	2016-07-28	Download	Change log	API docs
Python	2.53.6	2016-06-28	Download	Change log	API docs
Javascript (Node)	2.53.3	2016-06-28	<b>Download</b>	Change log	API docs

Selenium Standalone Serverは、リモートサーバーをしてテストをするにのみです。このはすべてプロジェクトをするにあるので、そのですべてをつがよい。

- 3. jarファイルはzipファイルにダウンロードされ、されます。.jarることができるはずです。
- 4. eclipseで、1でしたプロジェクトをクリックし、のにいます。

プロパティ -> Javaのビルドパス -> ライブラリのタブ -> Jarsの -> でダウンロードし たされたjarフォルダの -> 1ibフォルダからjarをすべて -> Okをクリック -> Add External Jars -> じclient-combined-3.0.0-beta1-nodeps.jar -> libフォルダのにあるjarを する client-combined-3.0.0-beta1-nodeps.jar -> Ok

 $\verb|C, OC \neg \tau_{\texttt{Selenium Standalone Server} \cup \texttt{Selenium Standalone Server}||$ 

5. これで、プロジェクトにセレンコードをきめることができます。

**PS**のドキュメントは、selenium-3.0.0ベータにづいているので、されたjarファイルのはバージョンによってわることがあります。

オンラインでselenium-webdriverをいめるをむ https://riptutorial.com/ja/seleniumwebdriver/topic/878/selenium-webdriverをいめる

2: HTMLレポート

### き

このトピックでは、セレンテストのHTMLレポートのについてします。レポートにはさまざまなのプラグインがあり、くされているのはAllure、ExtentReports、ReportNGです。

## **Examples**

エクステントレポート

このでは、TestNG、Java、MavenをしてSeleniumでExtentReportsをするについてします。

ExtentReportsには、コミュニティとの2つのバージョンがあります。わかりやすくなで、コミュ ニティをします。

#### 1.

エクステントレポートのをMavenのpom.xmlファイルにします。

```
<dependency>
<groupId>com.aventstack</groupId>
<artifactId>extentreports</artifactId>
<version>3.0.6</version>
</dependency>
```

```
2.プラグインの
```

#### pom.xmlにのようにmaven surefireプラグインをします

```
<build>
<defaultGoal>clean test</defaultGoal>
<plugins>
   <plugin>
        <proupId>org.apache.maven.plugins</proupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.6.1</version>
        <configuration>
            <source>${jdk.level}</source>
            <target>${jdk.level}</target>
        </configuration>
    </plugin>
    <plugin>
        <proupId>org.apache.maven.plugins</proupId>
        <artifactId>maven-surefire-plugin</artifactId>
        <version>2.19.1</version>
        <configuration>
            <suiteXmlFiles>
                <suiteXmlFile>testng.xml</suiteXmlFile>
            </suiteXmlFiles>
```

```
</configuration>
</plugin>
</plugins>
</build>
```

#### 3. ExtentReportsによるサンプルテスト

#### test.javaというのテストをします。

```
public class TestBase {
   WebDriver driver;
   ExtentReports extent;
   ExtentTest logger;
   ExtentHtmlReporter htmlReporter;
   String htmlReportPath = "C:\\Screenshots/MyOwnReport.html"; //Path for the HTML report to
be saved
    @BeforeTest
    public void setup() {
        htmlReporter = new ExtentHtmlReporter(htmlReportPath);
        extent = new ExtentReports();
       extent.attachReporter(htmlReporter);
        System.setProperty("webdriver.chrome.driver", "pathto/chromedriver.exe");
        driver = new ChromeDriver();
    }
    GTest
    public void test1() {
        driver.get("http://www.google.com/");
        logger.log(Status.INFO, "Opened site google.com");
        assertEquals(driver.getTitle()), "Google");
        logger.log(Status.PASS, "Google site loaded");
    }
    @AfterMethod
    public void getResult(ITestResult result) throws Exception {
        if (result.getStatus() == ITestResult.FAILURE)
            logger.log(Status.FAIL, MarkupHelper.createLabel(result.getName() + " Test case
FAILED due to below issues:", ExtentColor.RED));
           logger.fail(result.getThrowable());
        }
        else if (result.getStatus() == ITestResult.SUCCESS)
            logger.log(Status.PASS, MarkupHelper.createLabel(result.getName() + " Test Case
PASSED", ExtentColor.GREEN));
        }
        else if (result.getStatus() == ITestResult.SKIP)
            logger.log(Status.SKIP, MarkupHelper.createLabel(result.getName() + " Test Case
SKIPPED", ExtentColor.BLUE));
        }
     }
    @AfterTest
    public void testend() throws Exception {
```

```
extent.flush();
}
@AfterClass
public void tearDown() throws Exception {
    driver.close();
}
```

Allureレポート

このでは、TestNG、Java、およびMavenをしたSureniumでのAllure Reportsのについてします。

## **Maven**

リポジトリ

#### jcenterリポジトリをするのコードをしてください

#### pom.xmlにのをする

## シュアファイアプラグインの

## レポートのサンプルテスト

#### test.javaというのサンプル・テストをする

```
public class test{
    WebDriver driver;
    WebDriverWait wait;
    @BeforeMethod
    public void setup() {
       System.setProperty("webdriver.chrome.driver", "path to/chromedriver.exe");
        driver = new ChromeDriver();
        driver.get("https://www.google.com/");
        wait = new WebDriverWait(driver, 50);
    }
    @Title("Title check")
    @Description("Checking the title of the loaded page.")
    @Test
    public void searchTest() {
        String title = driver.getTitle();
       LogUtil.log("Title Fetched: "+title);
       assertEquals(title, "Google");
       LogUtil.log("Test Passed. Expected: Google | Actual: "+title);
        System.out.println("Page Loaded");
    }
    @AfterMethod
    public void teardown() {
       driver.close();
   }
}
```

のクラスでは、LogUtiilクラスをしました。これはにテストのステップをするためにわれます。は じもののためのコードです

#### LogUtil.java

```
public final class LogUtil {
    private LogUtil() {
    }
    @Step("{0}")
    public static void log(final String message){
        //intentionally empty
```

#### ここに

@Title ""はAllure Reportであなたのテストのタイトルをします

@Description ""はテストにをします

@Step ""はテストのレポートにステップをします

に、xmlファイルが "target / allure-results /"フォルダにされます

ジェンキンスとの

Allure ReportプラグインがインストールされたでJenkinsでしている、Jenkinsはジョブでレポートをにレンダリングします。

ジェンキンスのない

Jenkinsをっていないには、のコマンドラインをってhtmlレポートをしてください。 Allure CLIは Javaアプリケーションであるため、すべてのプラットフォームでできます。 Allure CLIをするに 、Java 1.7+をでインストールするがあります。

#### Debian

Debianベースのリポジトリの、PPAをしてインストールがですDebianのAllure CLIをインストー ルしてください

\$ sudo apt-add-repository ppa:yandex-qatools/allure-framework
\$ sudo apt-get update
\$ sudo apt-get install allure-commandline

サポートされているディストリビューションは、できるとです。インストール、なコマンドをできます。

マックOS

HomebrewでAllure CLIをインストールすることができます。

```
$ brew tap qatools/formulas
$ brew install allure-commandline
```

インスト―ル、なコマンドをできます。

WindowsおよびそののUnix

1. zipアーカイブとしてをhttps://github.com/allure-framework/allure-core/releases/latestからダ ウンロードしてください。 2. アーカイブをallure-commandlineディレクトリにします。 binディレクトリにします。 3. Windowsのはallure.bat、そののUnixプラットフォームのはallureをしてください。

コマンドライン/ターミナルでのをするだけで、レポートはallure-reportフォルダにされます

\$ allure ge	nerate directory-	-with-results/	
	llure	Suites name  duration  status	0
		Filter test cases by status: 0 0 2 0 0	
倄 Ove	rview	> Smoke : Test1	1
		✓ Sanity : Test1	1
Cate	egories	Title check 2s	6 061ms
📑 Suite	es		
<b></b> Gra	phs		
<li>O Time</li>	eline		
📕 Beh	aviors		
E Pacl	kages		
	En		
< Colla	apse		

オンラインでHTMLレポートをむ https://riptutorial.com/ja/selenium-webdriver/topic/10721/htmlレポート

## 3: Javaで@FindByアノテーションをする

- CLASS\_NAME@FindByclassName = "classname"
- CSS@FindBycss = "css"
- ID@FindByid = "id"
- ID\_OR\_NAME@FindByhow = ID\_OR\_NAME、 = "idOrName"を
- LINK\_TEXT@FindBylinkText = "text"
- NAME@FindByname = "name"
- PARTIAL\_LINK\_TEXT@FindBypartialLinkText = "text"
- TAG\_NAME@FindBytagName = "tagname"
- XPATH@FindByxpath = "xpath"

をするには2つのがあることにしてください。

@FindBy(id = "id")

#### そして

```
@FindBy(how = How.ID, using ="id")
```

```
がしく、ともそのIDによってをす。 ID_OR_NAMEは、
```

@FindBy(how = How.ID\_OR\_NAME, using ="idOrName")

 $_{\text{QFindBy}}$ アノテーションでマークされたをつけるには、  $_{\text{PageFactory.initElements()}}$ をページオブジェクトのインスタンスのにするがあります。

## **Examples**

な

### Pageオブジェクトモデルをしているとします。ページオブジェクトクラス

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
public class LoginPage {
    @FindBy(id="loginInput") //method used to find WebElement, in that case Id
    WebElement loginTextbox;
    @FindBy(id="passwordInput")
    WebElement passwordTextBox;
```

```
//xpath example:
@FindBy(xpath="//form[@id='loginForm']/button(contains(., 'Login')")
WebElement loginButton;
public void login(String username, String password){
    // login method prepared according to Page Object Pattern
    loginTextbox.sendKeys(username);
    passwordTextBox.sendKeys(password);
    loginButton.click();
    // because WebElements were declared with @FindBy, we can use them without
    // driver.find() method
}
```

#### そしてテストクラス

```
class Tests{
   public static void main(String[] args) {
     WebDriver driver = new FirefoxDriver();
     LoginPage loginPage = new LoginPage();
     //PageFactory is used to find elements with @FindBy specified
     PageFactory.initElements(driver, loginPage);
     loginPage.login("user", "pass");
   }
}
```

@FindBy - check セクションをして Web Elements をつけるはほとんどありません。

オンラインでJavaで@FindByアノテーションをするをむ https://riptutorial.com/ja/seleniumwebdriver/topic/6777/javaで-findbyアノテーションをする

## 4: JavaでSelenium Webdriverをする

### き

Selenium webdriverは、さまざまなWebブラウザにしてWebアプリケーションをテストできる Webフレームワークです。 Selenium IDEとはなり、webdriverでは、のテストケースをのプログ ラミングですることができます。 Java、.Net、PHP、Python、Perl、Rubyをサポートしています 。

## Examples

JavaでSelenium WebdriverをしてのURLのブラウザウィンドウをく

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class test_webdriver{
    public static void main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("http://stackoverflow.com/");
        driver.close();
    }
}
```

toメソッドをしてブラウザウィンドウをく

### toメソッドでブラウザをく

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
class navigateWithTo{
    public static void main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.navigate().to("http://www.example.com");
        driver.close();
    }
}
```

オンラインでJavaでSelenium Webdriverをするをむ https://riptutorial.com/ja/seleniumwebdriver/topic/9158/javaでselenium-webdriverをする

## **5: Python**、Ruby、Javascriptをした**Selenium**webdriverとCIツール

## き

これは、CircleCIでセレンテストをする1つのです

## **Examples**

Selenium Python  $\succeq$  Unittest2  $\succeq$   $\circ$  CircleCl

#### Circle.yml

```
machine:
    python:
    # Python version to use - Selenium requires python 3.0 and above
    version: pypy-3.6.0
dependencies:
    pre:
        # Install pip packages
        - pip install selenium
        - pip install selenium
        - pip install unittest
test:
    override:
    # Bash command to run main.py
    - python main.py
```

#### main.py

```
import unittest2
# Load and run all tests in testsuite matching regex provided
loader = unittest2.TestLoader()
# Finds all the tests in the same directory that have a filename that ends in test.py
testcases = loader.discover('.', pattern="*test.py")
test_runner = unittest2.runner.TextTestRunner()
# Checks that all tests ran
success = test_runner.run(testcases).wasSuccessful()
```

#### example\_test.py

```
class example_test(unittest.TestCase):
    def test_something(self):
        # Make a new webdriver instance
        self.driver = webdriver.Chrome()
        # Goes to www.gooogle.com
        self.driver.get("https://www.google.com")
```

オンラインでPython、Ruby、JavascriptをしたSelenium-webdriverとCIツールをむ

https://riptutorial.com/ja/selenium-webdriver/topic/10091/python-ruby-javascriptをしたselenium-webdriverとciツール

## 6: Selenium e2eセットアップ

## き

このトピックでは、Seleniumのエンドツーエンド、つまりSelenium Webdriver + TestNG + Maven + Jenkinsについてします。

レポートのについては、HTMLレポートのトピックをしてください。

## **Examples**

TestNGのセットアップ

TestNGは、junitにされたテストフレームワークです。テストスイートをびすためにtestng.xmlを するです。これは、にCIをするにちます。

## testng.xml

プロジェクトのルートフォルダにtestng.xmlというのxmlファイルをします。はなっていてもいませんが、どこでも "testng"としてされています。

は、testng.xmlファイルのなコードです。

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Smoke"> //name of the suite
<test name="Test1"> //name of the test
<classes>
<class name="test.SearchTest">
<methods>
<include name="searchTest"/>
</methods>
</class>
</class>
</test>
</suite>
```

Mavenセットアップ

TBD。 testng.xmlをびすためのpom.xmlの

ジェンキンスセットアップ

TBD。 git / bitbucketなどからコードをきすためのJenkinsのをカバーします。

オンラインでSelenium e2eセットアップをむ https://riptutorial.com/ja/seleniumwebdriver/topic/10724/selenium-e2eセットアップ

## 7: Selenium-WebDriver

き

Webdriverをしているにスローされるはいくつかあります。のは、それらのをするためのものです。

## Examples

**Python**𝔿

#### セレンドキュメント

ElementNotInteractableException DOMにがするが、そのとのやりとりがペイントのためにのに ヒットしたにスローされる

- ElementNotSelectableExceptionできないをしようとするとスローされます。できないの
   、スクリプト
- ElementNotVisibleExceptionがDOMにするがではないため、できないにスローされます。 されていないのテキストをクリックまたはみろうとするときによくします。
- ErrorInResponseExceptionサーバーでエラーがしたにスローされます。これは、firefoxまたはリモートドライバサーバとするときにするがあります。
- ImeActivationFailedException IMEエンジンのアクティブがしたときにスローされます。
- ImeNotAvailableException IMEサポートができないにスローされます。このは、IMEサポートがマシンででない、すべてのIMEメソッドびしでスローされます。
- InvalidArgumentExceptionコマンドにされたがかなです。
- InvalidCookieDomainExceptionのURLとなるドメインにCookieをしようとするとスローされます。
- InvalidElementStateExceptionアクションがにしてなになるにスローされます。サブクラス
  - ElementNotInteractableException
  - ElementNotSelectableException
  - ElementNotVisibleException
- InvalidSelectorExceptionのにされるセレクタがWebElementをさないにスローされます。 では、セレクタがxpathであり、にxpathではないか、でWebElementsたとえば "count// input"をしないとしません。
- InvalidSwitchToTargetExceptionりえるフレームまたはウィンドウのターゲットがしない にスローされます。
- MoveTargetOutOfBoundsException ActionsChains moveメソッドにされたターゲットが な、つまりドキュメントがしているにスローされます。
- NoAlertPresentExceptionされたアラートにりえるとスローされます。これは、アラートが まだにされていないときに、Alertクラスのをびすことによってするがあります。

- NoSuchAttributeException elementのがつからなかったにスローされます。テストしているのブラウザにがするかどうかをすることができます。のブラウザでは、じプロパティのプロパティがなるがあります。 IE8の.innerTextとFirefox .textContent
- NoSuchElementExceptionがつからなかったにスローされます。このがしたは、のをして ください。
  - 。あなたのfind\_byでされているセレクターをチェック...
  - にがまだにされていないがありますWebページがまだみまれています。がされるまで するラッパーをするについては、selenium.webdriver.support.wait.WebDriverWaitをし てください。
- NoSuchFrameExceptionりえるフレームターゲットがしないにスローされます。
- NoSuchWindowExceptionりえるウィンドウのターゲットがしないにスローされます。ア クティブなウィンドウハンドルののセットをつけるには、のでアクティブなウィンドウハン ドルのリストをします。
- print driver.window\_handles
   RemoteDriverServerException
- StaleElementReferenceExceptionへのが「くなった」ときにスローされます。とは、がページのDOMにされなくなったことをします。 StaleElementReferenceExceptionのとしては、のものがありますが、これにされません。
  - あなたはもはやじページにいないか、がされてからページがリフレッシュされている があります。
  - 。エレメントはされ、スクリーンにされているがあります。のなど。これは、、がされ、ノードがされるときに、javascriptフレームワークでします。
  - 。は、iframeやリフレッシュされたのコンテキストにあったがあります。
- TimeoutExceptionコマンドがなにしなかったにスローされます。
- UnableToSetCookieExceptionドライバがCookieのにしたときにスローされます。
- **UnexpectedAlertPresentException** bx r = -b b b b c c b b c c b b c c b b c c b b c c b b c c b c c b c c b c c b c c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b c b
- UnexpectedTagNameExceptionサポートクラスがしないWebをしなかったにスローされます。
- WebDriverExceptionWebdriverです。すべてのwebdriverは、WebDriverExceptionまたは InvalidStateExceptionをクラスとしてします。

オンラインでSelenium-WebDriverのをむ https://riptutorial.com/ja/seleniumwebdriver/topic/10546/selenium-webdriverの

## 8: Seleniumをしたオートメーションのエラー

## Examples

### Python

WebDriverException、ベースである<sub>Selenium-WebDriver</sub>キャッチするためにすることができのすべて Selenium-WebDriverを

をキャッチするには、まずインポートするがあります。

from selenium.common.exceptions import  ${\tt WebDriverException}$  as  ${\tt WDE}$ 

#### その

```
try:
    element = driver.find_element_by_id('ID')
except WDE:
    print("Not able to find element")
```

#### じように、よりなをインポートすることもできます。

```
from selenium.common.exceptions import ElementNotVisibleException
from selenium.common.exceptions import NoAlertPresentException
...
```

#### メッセ―ジのみをする

 $\label{eq:common_exceptions} \text{ import } \texttt{UnexpectedAlertPresentException}$ 

```
try:
    driver.find_element_by_tag_name('a').click()
except UnexpectedAlertPresentException as e:
    print(e.__dict__["msg"])
```

オンラインでSeleniumをしたオートメーションのエラーをむ https://riptutorial.com/ja/seleniumwebdriver/topic/9548/seleniumをしたオートメーションのエラー



## **Examples**

#### С

のをクリアするにテキストボックス

interactionWebElement.Clear();

#### へのデ---タにテキストボックス

interactionWebElement.SendKeys("Text");

#### のをする。

string valueinTextBox = interactionWebElement.GetAttribute("value");

#### のテキストの。

string textOfElement = interactionWebElement.Text;

#### をクリックする

interactionWebElement.Click();

#### フォームの

interactionWebElement.Submit();

#### ページののをする

bool isDisplayed=interactionWebElement.Displayed;

#### ページののをする

bool isEnabled = interactionWebElement.Enabled;

bool isSelected=interactionWebElement.Selected;

#### interactionWebElement $\mathcal{O}\mathcal{O}$

IWebElement childElement = interactionWebElement.FindElement(By.Id("childElementId"));

#### interactionWebElementのの

Ilist<IWebElement> childElements =
interactionWebElement.FindElements(By.TagName("childElementsTagName"));

Java

Webのをクリアする - テストでのユ―ザアクションをシミュレ―トするときは、バックスペ―ス をするがい、のアクションを

interactionWebElement.clear();

データの-キーストロークのをシミュレートする

interactionWebElement.sendKeys("Text"); interactionWebElement.sendKeys(Keys.CONTROL + "c"); // copy to clipboard.

#### ののをする

```
interactionWebElement.getAttribute("value");
interactionWebElement.getAttribute("style");
```

#### のテキストをする

String elementsText = interactionWebElement.getText();

#### ドロップダウンから

```
Select dropDown = new Select(webElement);
dropDown.selectByValue(value);
```

```
interactionWebElement.click();
interactionWebElement.submit(); //for forms
interactionWebElement.isDisplayed();
interactionWebElement.isEnabled(); // for exampale - is clickable.
interactionWebElement.isSelected(); // for radio buttons.
```

org.openqa.selenium.interactions.Actions をしたアクション

#### ドラッグドロップ

```
Action dragAndDrop = builder.clickAndHold(someElement)
.moveToElement(otherElement)
.release(otherElement)
.build();
```

```
dragAndDrop.perform();
```

```
Action selectMultiple = builder.keyDown(Keys.CONTROL)
   .click(someElement)
   .click(someOtherElement)
   .keyUp(Keys.CONTROL);
```

dragAndDrop.perform();

#### ビルダーを

```
builder.doubleClick(webElement).perform();
builder.moveToElement(webElement).perform(); //hovering
```

#### なアクションのとなリストについては、 ここをしてください。

#### Javascriptをう

```
// Scroll to view element:
((JavascriptExecutor) driver).executeJavaScript("arguments[0].scrollIntoView(true);",
webElement);
```

オンラインでWebとのをむ https://riptutorial.com/ja/selenium-webdriver/topic/4280/webとの

**10: Web**∽

• ByChainedparams By [] bys

は、 ロケータと<sub>By</sub>クラスをしてSeleniumでされます。 Seleniumでなプロジェクトをするには、 Web Elementsのロケータをスマートにするがあります。 ロケータはで、ユニークで、するはいは ずです。たとえば、テストでがないようにしてください。はのものをすることです。

- 1. **ID** それはユニークなので、あなたがむをにることができます。
- 2. クラス それはであり、えられたにおいてユニークであることができる。
- 3. CSS xpathよりもれたパフォーマンス よりなセレクタ。
- 4. **XPATH CSS**ができない **XPATH**の。えば<sub>div::parent</sub>。

りのロケータはやレンダリングがしやすく、けてください。

あなたのコードがのをつけることができない、あなたのコードがダウンロードするすべてのDOM をっていないことがの1つのがあります。のをするに、あなたのプログラムに ""35してから、に じてゆっくりとするようにしてください。は、 このからしたPythonのです

```
from selenium import webdriver
import time
browser = webdriver.Firefox()
browser.get("https://app.website.com")
reports_element = browser.find_element_by_xpath("//button[contains(text(), 'Reports')]")
# Element not found! Try giving time for the browser to download all DOM elements:
time.sleep(10)
```

reports\_element = browser.find\_element\_by\_xpath("//button[contains(text(), 'Reports')]")
# This returns correct value!

### **Examples**

WebDriverをしたページの

WebページでWebElementsとやりとりするには、まずのをするがあります。

Byはセレンでなキーワードです。

あなたは、By .. をつけることができます

```
1. ID により、
2. クラス
3. TagName により、
4.
```

で 5. リンク・テキスト が 6. なリンクテキスト で 7. CSSセレクタ で 8. XPathの ことで 9. JavaScript をする

のスクリプトのをえてみましょう

```
<form name="loginForm">
Login Username: <input id="username" name="login" type="text" />
Password: <input id="password" name="password" type="password" />
<input name="login" type="submit" value="Login" />
```

のコードでは、ユーザーとパスワードはIDをしてされています。はidをつをしようとしています

driver.findElement(By.id(username));

driver.findElement(By.id(password));

セレンは7つのなるをサポートしていますので、このではすべてのでをつけるアイディアをしてい ます。

### IDで

IDをってをつけるの

<div id="coolestWidgetEvah">...</div>
Java - WebElement element = driver.findElement(By.id("coolestWidgetEvah"));
C# - IWebElement element = driver.FindElement(By.Id("coolestWidgetEvah"));
Python - element = driver.find\_element\_by\_id("coolestWidgetEvah")
Ruby - element = driver.find\_element(::id, "coolestWidgetEvah")
JavaScript/Protractor - var elm = element(by.id("coolestWidgetEvah"));

### クラスで

#### クラスをしてをつけるの

```
<div class="cheese"><span>Cheddar</span></div>
Java - WebElement element = driver.findElement(By.className("cheese"));
C# - IWebElement element = driver.FindElement(By.ClassName("cheese"));
Python - element = driver.find_element_by_class_name("cheese")
Ruby - cheeses = driver.find_elements(:class, "cheese")
JavaScript/Protractor - var elm = element(by.className("cheese"));
```

## タグによる

#### タグをってをつけるの

```
<iframe src="..."></iframe>
Java - WebElement element = driver.findElement(By.tagName("iframe"));
C# - IWebElement element = driver.FindElement(By.TagName("iframe"));
Python - element = driver.find_element_by_tag_name("iframe")
Ruby - frame = driver.find_element(:tag_name, "iframe")
JavaScript/Protractor - var elm = element(by.tagName("iframe"));
```

で

#### nameをってをつけるの

```
<input name="cheese" type="text"/>
Java - WebElement element = driver.findElement(By.name("cheese"));
C# - IWebElement element = driver.FindElement(By.Name("cheese"));
Python - element = driver.find_element_by_name("cheese")
Ruby - cheese = driver.find_element(:name, "cheese")
JavaScript/Protractor - var elm = element(by.name("cheese"));
```

```
リンクテキスト
```

#### リンクテキストをってをつけるの

<a href="http://www.google.com/search?q=cheese">cheese</a>>

Java	-	<pre>WebElement element = driver.findElement(By.linkText("cheese"));</pre>
C#	-	<pre>IWebElement element = driver.FindElement(By.LinkText("cheese"));</pre>
Python	-	<pre>element = driver.find_element_by_link_text("cheese")</pre>
Ruby	-	<pre>cheese = driver.find_element(:link, "cheese")</pre>
JavaScript	/Pr	<pre>otractor - var elm = element(by.linkText("cheese"));</pre>

## リンクテキスト

#### リンクテキストをしてをつけるの

```
<a href="http://www.google.com/search?q=cheese">search for cheese</a>>
```

```
Java - WebElement element = driver.findElement(By.partialLinkText("cheese"));
C# - IWebElement element = driver.FindElement(By.PartialLinkText("cheese"));
Python - element = driver.find_element_by_partial_link_text("cheese")
Ruby - cheese = driver.find_element(:partial_link_text, "cheese")
JavaScript/Protractor - var elm = element(by.partialLinkText("cheese"));
```
# CSSセレクタによる

### CSSセレクタをってをつけるの

```
<div id="food" class="dairy">milk</span>
Java - WebElement element = driver.findElement(By.cssSelector("#food.dairy")); //# is
used to indicate id and . is used for classname.
C# - IWebElement element = driver.FindElement(By.CssSelector("#food.dairy"));
Python - element = driver.find_element_by_css_selector("#food.dairy")
Ruby - cheese = driver.find_element(:css, "#food span.dairy.aged")
JavaScript/Protractor - var elm = element(by.css("#food.dairy"));
```

CSSセレクタのにするはのとおりです。http://www.w3schools.com/cssref/css\_selectors.asp

# XPathによる

#### XPathをしてをつけるの

<input name="example" type="text"/>				
Java	-	<pre>WebElement element = driver.findElement(By.xpath("//input"));</pre>		
C#	-	<pre>IWebElement element = driver.FindElement(By.XPath("//input"))</pre>		
Python	-	<pre>element = driver.find_element_by_xpath("//input")</pre>		
Ruby	-	<pre>inputs = driver.find_elements(:xpath, "//input")</pre>		
<pre>JavaScript/Protractor - var elm = element(by.xpath("//input"));</pre>				

XPathにするはのとおりです。http://www.w3schools.com/xsl/xpath\_intro.asp

# JavaScriptをする

のjavascriptをしてをつけることができます。また、DOMをすり、にWebElementオブジェクトに されます。

jQueryがみまれたページのな

```
Java - WebElement element = (WebElement)
((JavascriptExecutor)driver).executeScript("return $('.cheese')[0]");
C# - IWebElement element = (IWebElement)
((IJavaScriptExecutor)driver).ExecuteScript("return $('.cheese')[0]");
Python - element = driver.execute_script("return $('.cheese')[0]");
Ruby - element = driver.execute_script("return $('.cheese')[0]")
JavaScript/Protractor -
```

のWebDriverがSimpleBrowserなどのJavaScriptをサポートしていない、このはしません。

### のによる[C]

セレクタをにすることもできます。これは、 OpenQA.Selenium.Support.PageObjects.ByChainedオブジ エクトをしてOpenQA.Selenium.Support.PageObjects.ByChainedます。

element = driver.FindElement(new ByChained(By.TagName("input"), By.ClassName("class"));

### ののByさせることができ、ANDのとしてされますつまり、 Byはすべてします

ページのみみがするにをする

driver.Navigate().GoToUrl(url); びすときdriver.Navigate().GoToUrl(url); 、 ページがにロードされるまでコードのがします。これは、データをするだけのにはなことがあります。

のコードサンプルはハックとなすことができます。これをう "の"はありません。

## しいスレッドをする

Webページをみむスレッドをしてし、 Waitをします。

#### С

```
using (var driver = new ChromeDriver())
{
    new Thread(() =>
    {
        driver.Navigate().GoToUrl("http://stackoverflow.com");
    }).Start();
    new WebDriverWait(driver, TimeSpan.FromSeconds(10))
        .Until(ExpectedConditions.ElementIsVisible(By.XPath("//div[@class='summary']/h3/a")));
}
```

# タイムアウトをする

WebDriverTimeoutをすると、ページをみむことができます。がするとがスローされ、ページのみ みがします。 catchブロックでは、 Waitをできます。

```
С
```

```
using (var driver = new ChromeDriver())
{
    driver.Manage().Timeouts().SetPageLoadTimeout(TimeSpan.FromSeconds(5));
    try
    {
        driver.Navigate().GoToUrl("http://stackoverflow.com");
    }
}
```

```
}
catch (WebDriverTimeoutException)
{
    new WebDriverWait(driver, TimeSpan.FromSeconds(10))
    .Until(ExpectedConditions.ElementIsVisible
        (By.XPath("//div[@class='summary']/h3/a")));
}
```

タイムアウトをすぎるにすると、のがするかどうかになく、ページのみみがします。タイムアウトをすぎるにすると、パフォーマンスのメリットをにします。

オンラインでWebのをむ https://riptutorial.com/ja/selenium-webdriver/topic/3991/webの

# 11: アクションなユーザーのジェスチャーをエミ ュレートする

き

Actionsクラスは、ユーザーがWebページ/とどのようにするかをにエミュレートするをします。このクラスのインスタンスをすると、クリック、ダブルクリック、ドラッグ、キーのなどののアクションをできます。これらのアクションがされると、アクションをするためにアクションをビルドするがあります .Build() をし、するようにします .Perform() 。だから々は、し、し、するがあります。のはこれにづいてされます。

- dragAndDropWebElementソース、WebElementターゲット
- dragAndDropByWebElement source int xOffset int yOffset
- perform

パラメ—タ—

パラメ—タ—	
ソース	ボタンをエミュレートする。
ターゲット	マウスをしてマウスをす。
xOffset	xにする。
yオフセット	yにする。

このセクションでは、Selenium WebDriverのActionsクラスについてします。 Actionsクラスは、 ドラッグアンドドロップ、やクリックなどのなユーザジェスチャをするためのなメソッドをしま す。

# **Examples**

ドラッグアンドドロップ

# С

using OpenQA.Selenium; using OpenQA.Selenium.Firefox; using OpenQA.Selenium.Interactions;

```
namespace WebDriverActions
{
    class WebDriverTest
    {
        static void Main()
        {
            IWebDriver driver = new FirefoxDriver();
            driver.Navigate().GoToUrl("");
            IWebElement source = driver.FindElement(By.CssSelector(""));
            IWebElement target = driver.FindElement(By.CssSelector(""));
            Actions action = new Actions(driver);
            action.DragAndDrop(source, target).Perform();
        }
    }
}
```

のでは、 IWebElement source をつけてドラッグし、2の IWebElement target ドロップします。

# Java

ソースとターゲットのWebをしてドラッグアンドドロップします。

ソースのでクリックホールドをし、ターゲットのにしてからマウスをしていくなメソッドです。

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.interactions.Actions;
/**
 * Drag and Drop test using source and target webelement
*/
public class DragAndDropClass {
   public static void main(String[] args) {
        WebDriver driver = new FirefoxDriver();
        driver.get("");
       WebElement source = driver.findElement(By.cssSelector(""));
        WebElement target = driver.findElement(By.cssSelector(""));
        Actions action = new Actions(driver);
        action.build();
       action.dragAndDrop(source, target).perform();
   }
}
```

### をドラッグして、されたオフセットにドロップします。

ソースのでクリックアンドホールドをし、されたオフセットxとy、のだけしてからマウスをして いくメソッドです。

```
WebElement source = driver.findElement(By.cssSelector(""));
Actions action = new Actions(driver);
```

#### に

# С

のにマウスをくと、ドロップリストがされることをテストしたいとします。このリストのをした り、リストからオプションをすることができます。

に、のにマウスをするアクションをしますたとえば、myにリンクテキスト "Admin"があります。

Actions mouseHover = new Actions(driver); mouseHover.MoveToElement(driver.FindElement(By.LinkText("Admin"))).Perform();

#### のでは、

- アクションmouseHoverをしましたmouseHover
- あなたは<sub>driver</sub>にのにするようにいました
- ここから、  $_{mouseHover}$ オブジェクトでの $_{Actions}$ をするか、  $_{driver}$ オブジェクトでテストをけることができます

このは、をクリックすると、そのにカ-ソルをくのとはなるをするときににされます。

#### な

Actions mouseHover = new Actions(driver); mouseHover.MoveToElement(driver.FindElement(By.LinkText("Admin"))).Perform();

Assert.IsTrue(driver.FindElement(By.LinkText("Edit Record")).Displayed); Assert.IsTrue(driver.FindElement(By.LinkText("Delete Record")).Displayed);

### オンラインでアクションなユーザーのジェスチャーをエミュレートするをむ

https://riptutorial.com/ja/selenium-webdriver/topic/4849/アクション-なユーザーのジェスチャーを エミュレートする-

12: アラートをする

## **Examples**

Javaでのセレン

SeleniumをってJavaでポップアップアラートをするはのとおりです。

ポップアップは3あります。

1.シンプルアラート アラート「これはなアラート」。

2. アラート var popuResult = confirm "OKとキャンセルボタンでポップアップをする";

3.  $\neg$  ロンプトの var person = prompt "あなたはstackoverflowがきですか"、 "はい/いいえ";

ポップアップのをテストケースでするがあるのは、そのユーザーだけです。

どちらか

- 1. acceptアラートをけれる
- 2. dismissアラートをじる
- 3. getTextアラートのテキストをする
- 4. sendKeysにテキストをきむ

なアラートの

```
Alert simpleAlert = driver.switchTo().alert();
String alertText = simpleAlert.getText();
System.out.println("Alert text is " + alertText);
simpleAlert.accept();
```

アラートの

```
Alert confirmationAlert = driver.switchTo().alert();
String alertText = confirmationAlert.getText();
System.out.println("Alert text is " + alertText);
confirmationAlert.dismiss();
```

プロンプトの

```
Alert promptAlert = driver.switchTo().alert();
String alertText = promptAlert .getText();
System.out.println("Alert text is " + alertText);
//Send some text to the alert
promptAlert .sendKeys("Accepting the alert");
Thread.sleep(4000); //This sleep is not necessary, just for demonstration
```

#### あなたのにじて。

これをうのは、コードをtry-catchのにラップすることです。

```
try{
   // Your logic here.
} catch(UnhandledAlertException e){
   Alert alert = driver.switchTo().alert();
   alert.accept();
}
// Continue.
```

### С

CでSeleniumのポップアップアラートをじるはのとおりです。

```
IAlert alert = driver.SwitchTo().Alert();
// Prints text and closes alert
System.out.println(alert.Text);
alert.Accept();
or
alert.Dismiss();
```

#### あなたのにじて。

これをうのは、コードをtry-catchのにラップすることです。

```
try{
   // Your logic here.
} catch(UnhandledAlertException e){
   var alert = driver.SwitchTo().Alert();
   alert.Accept();
}
// Continue.
```

### **Python**

Pythonアラートポップアップにりえるには、のがあります。

#### 1. されない

alert = driver.switch\_to\_alert()

### 2. switch\_toをする

alert = driver.switch\_to.alert

3. ExplicitWaitをう

from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected\_conditions as EC

alert = WebDriverWait(driver, TIMEOUT\_IN\_SECONDS).until(EC.alert\_is\_present())

### 4. Alertクラスのインスタンスをすることによって

from selenium.webdriver.common.alert import Alert

```
alert = Alert(driver)
```

JavaScript prompt () トリガされたフィールドをポップアップでするには

alert.send\_keys('Some text to send')

#### ダイアログポップアップをする\*

alert.accept()

#### するには

alert.dismiss()

テキストをポップアップからするには

alert.text

\* **PS**  $_{alert.dismiss()}$ は、  $_{JavaScript alert()}$ と $_{alert.confirm()}$ によってトリガーされたポップアップをするためにできます。

オンラインでアラートをするをむ https://riptutorial.com/ja/selenium-webdriver/topic/6048/アラートをする

# **13:** クラスを

- Java
- すべてのを
- deselectByIndexint index
- deselectByValuejava.lang.String value
- deselectByVisibleTextjava.lang.String text
- getAllSelectedOptions
- getFirstSelectedOption
- getOptions
- isMultiple
- selectByIndexint index
- selectByValuejava.lang.String value
- selectByVisibleTextjava.lang.String text

パラメ—タ—

パラメ <b>―タ</b> ―	
	このインデックスのオプションがされます
	する
テキスト	するテキスト

 $s_{elect}$ クラスのSelenium WebDriverは、  $s_{elect}$ オプションとするためのなメソッドを $s_{elect}$ ます。 ユーザーは、したドロップダウンでをし、のをしてをすることもできます。

**C**では、Selectクラスはには<sub>SelectElement</sub>

# **Examples**

ドロップダウンリストからするさまざまな

### はHTMLページです

```
<html>
<head>
<title>Select Example by Index value</title>
</head>
<body>
<select name="Travel"><option value="0" selected> Please select</option>
<option value="1">Car</option>
<option value="2">Bike</option>
<option value="3">Cycle</option>
```



インデックスに

### Javaをしてでオプションをするには

```
public class selectByIndexExample {
    WebDriver driver;
    @Test
    public void selectSamples()
    {
        driver = new FirefoxDriver();
        driver.get("URL GOES HERE");
        WebElement element=driver.findElement(By.name("Travel")); //This is the 'Select'
element locator
        Select sel=new Select(element);
        sel.selectByIndex(1); //This will select the first 'Option' from 'Select' List i.e.
Car
    }
}
```

# で

```
public class selectByValueExample {
    WebDriver driver;
    @Test
    public void selectSamples()
    {
        driver = new FirefoxDriver();
        driver.get("URL GOES HERE");
        WebElement element=driver.findElement(By.name("Travel")); //This is the 'Select'
element locator
        Select sel=new Select(element);
        sel.selectByValue("Bike"); //This will select the 'Option' from 'Select' List which
has value as "Bike".
        //NOTE: This will be case sensitive
    }
}
```

# テキストで

```
public class selectByVisibilityTextExample {
    WebDriver driver;
    @Test
    public void selectSamples()
    {
```

```
driver = new FirefoxDriver();
driver.get("URL GOES HERE");
WebElement element=driver.findElement(By.name("Travel")); //This is the 'Select'
element locator
Select sel=new Select(element);
sel.selectByVisibleText("Cycle"); //This will select the 'Option' from 'Select' List
who's visibility text is "Cycle".
//NOTE: This will be case sensitive
}
```

# С

### のすべてのは、な<sub>IWebDriver</sub>インターフェイスにづいています

インデックスに

```
IWebElement element=driver.FindElement(By.name("Travel"));
SelectElement selectElement = new SelectElement(title);
selectElement.SelectByIndex(0);
```

## で

```
IWebElement element=driver.FindElement(By.name("Travel"));
SelectElement selectElement = new SelectElement(title);
selectElement.SelectByIndex("1");
//NOTE: This will be case sensitive
```

# テキストで

```
IWebElement element=driver.FindElement(By.name("Travel"));
SelectElement selectElement = new SelectElement(title);
selectElement.SelectByText("Walk");
```

オンラインでクラスををむ https://riptutorial.com/ja/selenium-webdriver/topic/6426/クラスを

14: スイッチングフレーム

- Java
- driver.switchTo。frameストリング;
- driver.switchTo。frameストリングid;
- driver.switchTo. frameint index;
- フレームワークWebElement frameElement。
- driver.switchTo. defaultContent;
- C
- ドライバ.SwitchTo。フレームint frameIndex;
- driver.SwitchTo。フレームIWebElementのframeElement。
- driver.SwitchTo<sub>□</sub> フレームstring frameName;
- デフォルトのコンテンツ。
- Python
- driver.switch\_to\_framenameOrld
- driver.switch\_to.framenameOrld
- driver.switch\_to\_frameindex
- driver.switch\_to.frameindex
- driver.switch\_to\_frameframeElement
- driver.switch\_to.frameframeElement
- driver.switch\_to\_default\_content
- driver.switch\_to.default\_content
- JavaScript
- driver.switchTo<sub>°</sub> framenameOrld
- driver.switchTo<sub>°</sub> frameindex
- driver.switchTo defaultContent

パラメ-タ-

パラメ <b>―タ</b>	
nameOrld	idのでフレームをします。
	ゼロベースのインデックスでフレームをします。
frameElement	にされたWebElementをしてフレ―ムをする

## **Examples**

Javaをしてフレームにりえるには

インスタンスの、htmlビューまたはのhtmlソースコードがのようにiframeでラップされているとし

```
<iframe src="../images/eightball.gif" name="imgboxName" id="imgboxId">
    iframes example
    <a href="../images/redball.gif" target="imgbox">Red Ball</a>
</iframe><br />
```

... iframeのWebでアクションをするには、まず、のいずれかのをしてフォーカスをiframeにりえるがあります。

フレームIDをする iframeのIDをっているにのみしてください。

driver.switchTo().frame("imgboxId"); //imgboxId - Id of the frame

フレームをする iframeのがわかっているにのみしてください。

driver.switchTo().frame("imgboxName"); //imgboxName - Name of the frame

フレームインデックスをする iframeのIDまたはがないにのみするがあります。インデックスでは 、すべてのフレームのiframeのがされます。

driver.switchTo().frame(0); //0 - Index of the frame

ページに3つのフレームがある、のフレームはインデックス0に、2のフレームはインデックス1に、3のフレームはインデックス2になります。

にされたwebelementをする にフレームをつけて $_{WebElement}$ としてしたのみする

driver.switchTo().frame(frameElement); //frameElement - webelement that is the frame

Red Ballアンカーをクリックするには

driver.switchTo().frame("imgboxId"); driver.findElement(By.linkText("Red Ball")).Click();

Javaをしてフレームからけるには

メインまたはページののフレームのいずれかにフォーカスをりえます。のをするがあります。

driver.switchTo().defaultContent();

Cをしてフレームにりえる

1.インデックスでフレームにりえます。

ここでは、インデックス1にりわります。インデックスは、ページのフレームのをします。フレー

ムIDやがはるかにできるため、これはのとしてするがあります。

driver.SwitchTo().Frame(1);

2.でフレームにりえます。

driver.SwitchTo().Frame("Name\_Of\_Frame");

#### 3. IWebElementをして、タイトル、IDなどでフレームにりえます。

IDまたはタイトルでフレームにりえるは、Webをパラメータとしてすがあります。

driver.SwitchTo().Frame(driver.FindElement(By.Id("ID\_OF\_FRAME"))); driver.SwitchTo().Frame(driver.FindElement(By.CssSelector("iframe[title='Title\_of\_Frame']")));

また、フレームがされるまでかかるは、 をするがあります。

```
new WebDriverWait(driver, TimeSpan.FromSeconds(10))
.Until(ExpectedConditions.ElementIsVisible(By.Id("Id_Of_Frame")));
```

フレームからる

driver.SwitchTo().DefaultContent()

Cをしてフレームからけすには

メインまたはページののフレームのいずれかにフォーカスをりえます。のをするがあります。

webDriver.SwitchTo. DefaultContent;

フレームのフレームのりえ。

フレームフレーム - があるとします。 2つのフレームFrame\_Son、Frame\_Daughter。さまざまなやりいをてみましょう。

1.からまたはから

```
driver.switchTo().frame("Frame_Son");
driver.switchTo().frame("Frame_Daughter");
```

からへがデフォルトフレームのは、デフォルトフレームにりえ、それのはデフォルトフレームか らフレームにりえます。しかし、あなたはからにりえることはできません。

```
driver.switchTo().defaultContent();
driver.switchTo().frame("Frame_Parent");
```

3.からまであなたのがかいをしたは、のびをかずに、あなたのにをしべてください。に、フレームとフレームにをえます。

```
driver.switchTo().defaultContent();
driver.switchTo().frame("Frame_Parent");
driver.switchTo().frame("Frame_Daughter");
```

フレームがみまれるまでちます

かなりの、フレームがすぐにされず、スイッチがロードされるまでつがあるかもしれません。それのは、NoSuchFrameExceptionがします。

あなたがりえるにつことはにいです。は、フレームがロードされるまでつなです。

try{

until(ExpectedConditions.visibilityOf((driver.findElement(By.id("cpmInteractionDivFrame"))));}
 catch{

//あなたのフレームがあなたのち300でされないにのみをスローする}

オンラインでスイッチングフレームをむ https://riptutorial.com/ja/selenium-webdriver/topic/4589/ スイッチングフレーム

# 15: スクリーンショットをる

# き

スクリーンショットをってのパスにする

- ファイルsrc =TakesScreenshotドライバ.getScreenshotAsOutputType.FILE;
- FileUtils.copyFilesrc、しいファイル "D\ screenshot.png";

# Examples

### JAVA

### スクリーンショットをってするコード

```
public class Sample
{
   public static void main (String[] args)
    {
        *//Initialize Browser*
       System.setProperty("webdriver.gecko.driver", "**E:\\path\\to\\geckodriver.exe**");
       WebDriver driver = new FirefoxDriver();
       driver.manage().window().maximize();
        driver.get("https://www.google.com/");
        //Take Screesnshot
       File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
        try {
            //Save Screenshot in destination file
           FileUtils.copyFile(src, new File("D:\\screenshot.png"));
        } catch (IOException e) {
            e.printStackTrace();
        }
   }
}
```

## スクリ―ンショットをる

File src = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);

### ソースからまでのスクリーンショット

FileUtils.copyFile(src, new File("D:\\screenshot.png"));

オンラインでスクリーンショットをるをむ https://riptutorial.com/ja/seleniumwebdriver/topic/10094/スクリーンショットをる

# **16:** スクロール

# き

このトピックでは、seleniumスクロールをするのいくつかのをします

# **Examples**

Pythonをったスクロール

1. Actionsターゲットへのスクロールページのにある「BROWSE TEMPLATES」ボタン

```
from selenium import webdriver
from selenium.webdriver.common.action_chains import ActionChains
driver = webdriver.Chrome()
driver.get('http://www.w3schools.com/')
target = driver.find_element_by_link_text('BROWSE TEMPLATES')
actions = ActionChains(driver)
actions.move_to_element(target)
actions.perform()
```

### 2. JavaScriptをJavaScriptしたターゲットへのスクロールページのにある「BROWSE TEMPLATES」ボタン

from selenium import webdriver
driver = webdriver.Chrome()
driver.get('http://www.w3schools.com/')
target = driver.find\_element\_by\_link\_text('BROWSE TEMPLATES')
driver.execute\_script('arguments[0].scrollIntoView(true);', target)

3.メソッドをみんだターゲットへのスクロールページのにある「BROWSE TEMPLATES」ボタン

```
from selenium import webdriver
driver = webdriver.Chrome()
driver.get('http://www.w3schools.com/')
target = driver.find_element_by_link_text('BROWSE TEMPLATES')
target.location_once_scrolled_into_view
```

 $location\_once\_scrolled\_into\_view$ は、スクロールののx、 yもすことにしてください

4. Keysをしてページのにスクロールする

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
```

driver = webdriver.Chrome()
driver.get('http://www.w3schools.com/')
driver.find\_element\_by\_tag\_name('body').send\_keys(Keys.END) # Use send\_keys(Keys.HOME) to
scroll up to the top of page

send\_keys (Keys.DOWN) / send\_keys (Keys.UP) と send\_keys (Keys.PAGE\_DOWN) / send\_keys (Keys.PAGE\_UP) もで きることにしてください

さまざまなでjavaをしてなるスクロール

のソリューションは、サポートされているのプログラミングでもできます

1. カスタムスクロールバーブラウザスクロールではありませんがあるに、Webページのページ /セクション/ディビジョンをにスクロールします。 ここをクリックデモとチェックスクロー ルバーにはしたがあります。

のコードでは、スクロールバーのをし、スクロールポイントがです。

```
public static boolean scroll_Page (WebElement webelement, int scrollPoints)
   {
   try
   {
       System.out.println("------ Started - scroll_Page -----");
       driver = ExecutionSetup.getDriver();
       dragger = new Actions(driver);
       // drag downwards
       int numberOfPixelsToDragTheScrollbarDown = 10;
       for (int i = 10; i < scrollPoints; i = i + numberOfPixelsToDragTheScrollbarDown)
           dragger.moveToElement(webelement).clickAndHold().moveByOffset(0,
numberOfPixelsToDragTheScrollbarDown).release(webelement).build().perform();
       Thread.sleep(500);
       System.out.println("----- Ending - scroll_Page -----");
       return true;
   }
   catch (Exception e)
       System.out.println("----- scroll is unsucessfully done in scroll_Page -----
       ----");
       e.printStackTrace();
       return false;
   }
  }
```

 カスタムスクロールバーブラウザスクロールではありませんがあるに、Webページのスクロ ールアップページ/セクション/ディビジョンをいます。 ここをクリックデモとチェックスク ロールバーにはしたがあります。

のコードでは、スクロールバーのをし、スクロールポイントがです。

```
public static boolean scroll_Page_Up(WebElement webelement, int scrollPoints)
{
   try
   {
       System.out.println("------ Started - scroll_Page_Up ------");
       driver = ExecutionSetup.getDriver();
       dragger = new Actions(driver);
       // drag upwards
       int numberOfPixelsToDragTheScrollbarUp = -10;
       for (int i = scrollPoints; i > 10; i = i + numberOfPixelsToDragTheScrollbarUp)
           dragger.moveToElement(webelement).clickAndHold().moveByOffset(0,
numberOfPixelsToDragTheScrollbarUp).release(webelement).build().perform();
       System.out.println("----- Ending - scroll_Page_Up -----");
       return true;
   }
   catch (Exception e)
   {
       System.out.println("----- scroll is unsucessfully done in scroll_Page_Up---
   ----");
       e.printStackTrace();
      return false;
   }
}
```

3. のブラウザのスクロール ブラウザをい、 Page Downキーをしてにスクロールしたいときに スクロールします。 デモにここをクリック

のコードでは、<sub><div></sub>ようなスクロールエリアをし、ページダウンキーがです。

```
public static boolean pageDown_New(WebElement webeScrollArea, int iLoopCount)
{
   try
   {
      System.out.println("----- Started - pageDown_New -----");
      driver = ExecutionSetup.getDriver();
      dragger = new Actions(driver);
      for (int i = 0; i <= iLoopCount; i++)</pre>
dragger.moveToElement(webeScrollArea).click().sendKeys(Keys.PAGE_DOWN).build().perform();
      }
      System.out.println"----- Ending - pageDown_New ------");
      return true;
   }
   catch (Exception e)
   {
      System.out.println("-----");
      return false;
   }
}
```

4. のブラウザスクロ─ル ブラウザをしていて、 **Page** Up キ─でにスクロ─ルしたいは、にス クロ─ルします。 デモにここをクリック

```
のコードでは、<sub><div></sub>ようなスクロールエリアをし、ページアップキーがです。
```

```
public static boolean pageUp_New(WebElement webeScrollArea, int iLoopCount)
{
   try
   {
       System.out.println("------ Started - pageUp_New ------");
       driver = ExecutionSetup.getDriver();
       dragger = new Actions(driver);
       for (int i = 0; i <= iLoopCount; i++)</pre>
       {
dragger.moveToElement(webeScrollArea).click().sendKeys(Keys.PAGE_UP).build().perform();
       System.out.println("----- Ending - pageUp_New -----");
       return true;
   }
   catch (Exception e)
   {
      System.out.println("-----"Not able to do page up -----");
      return false;
   }
}
```

5. のブラウザスクロール ブラウザときキーのみでスクロールしたいは、にスクロールします 。デモにここをクリック

のコードでは、<br/>
くdiv>ようなスクロールエリアをし、ダウンキーがです。

```
public static boolean scrollDown_Keys (WebElement webeScrollArea, int iLoopCount)
{
   try
   {
       System.out.println("------ Started - scrollDown_Keys ------");
       driver = ExecutionSetup.getDriver();
       dragger = new Actions(driver);
       for (int i = 0; i <= iLoopCount; i++)</pre>
dragger.moveToElement(webeScrollArea).click().sendKeys(Keys.DOWN).build().perform();
       }
       System.out.println("----- Ending - scrollDown_Keys -----");
       return true;
   }
   catch (Exception e)
   {
       System.out.println("----- Not able to do scroll down with keys------
---");
       return false;
   }
```

6. のブラウザスクロール ブラウザときキーでにスクロールするは、にスクロールします。 デ

#### モにここをクリック

のコードでは、<div>ようなスクロールエリアをし、キーをとします。

```
public static boolean scrollUp_Keys(WebElement webeScrollArea, int iLoopCount)
{
   try
   {
       System.out.println("------ Started - scrollUp_Keys -----");
       driver = ExecutionSetup.getDriver();
       dragger = new Actions(driver);
       for (int i = 0; i <= iLoopCount; i++)</pre>
       {
          dragger.moveToElement(webeScrollArea).click().sendKeys(Keys.UP).build().perform();
       System.out.println("----- Ending - scrollUp_Keys -----");
       return true;
   }
   catch (Exception e)
   {
       System.out.println("----- Not able to do scroll up with keys------
-");
       return false;
   }
}
```

7. ブラウザのスクロール ブラウザとアップ/ダウンスクロールをのみでスクロールするは/をス クロールします。 デモにここをクリック

のコードでは、スクロールポイントをします。ポジティブはダウン、ネガティブはス クロールアップします。

```
public static boolean scroll_without_WebE(int scrollPoint)
{
   JavascriptExecutor jse;
   try
   {
       System.out.println("------ Started - scroll_without_WebE ------");
       driver = ExecutionSetup.getDriver();
       jse = (JavascriptExecutor) driver;
       jse.executeScript("window.scrollBy(0," + scrollPoint + ")", "");
       System.out.println("----- Ending - scroll_without_WebE -----");
       return true;
   }
   catch (Exception e)
   {
       System.out.println("----- scroll is unsucessful in scroll_without_WebE ----
     ----");
       e.printStackTrace();
       return false;
   }
}
```

8. ブラウザのスクロール In-Builtブラウザと/にスクロールするにスクロールアップ/ダウンする にはまたはダイナミックスクロールのをします。 デモにここをクリック

のコードでは、あなたのをします。

```
public static boolean scroll_to_WebE(WebElement webe)
{
   try
   {
      System.out.println("------ Started - scroll_to_WebE ------");
       driver = ExecutionSetup.getDriver();
       ((JavascriptExecutor) driver).executeScript("arguments[0].scrollIntoView();", webe);
      System.out.println("----- Ending - scroll_to_WebE -----");
      return true;
   }
   catch (Exception e)
   {
      System.out.println("------ scroll is unsucessful in scroll_to_WebE ------
  ----");
      e.printStackTrace();
      return false;
   }
}
```

ケースをし、をしてください。いずれかのケースがつからないは、にらせてください。

オンラインでスクロールをむ https://riptutorial.com/ja/selenium-webdriver/topic/9063/スクロール

17: セレングリッド

## **Examples**

ノード

Selenium Grid Nodeはノードにあり、ネットワークとノードにするをしています。はさまざまな でできます。

- デフォルト
- JSONの
- コマンドラインの

#### JSONの

JSONファイルのノードは2つのセクションにされています。

•

は、サポートされているブラウザのとバージョン、ブラウザバイナリの、ブラウザののインスタンスなどのをします。

は、ハブやノードのアドレスやポートなどのをします。

はJSONファイルのです

```
{
  "capabilities": [
   {
     "browserName": "firefox",
     "acceptSslCerts": true,
     "javascriptEnabled": true,
     "takesScreenshot": false,
     "firefox_profile": "",
     "browser-version": "27",
      "platform": "WINDOWS",
      "maxInstances": 5,
      "firefox_binary": "",
      "cleanSession": true
   },
    {
     "browserName": "chrome",
      "maxInstances": 5,
      "platform": "WINDOWS",
      "webdriver.chrome.driver": "C:/Program Files (x86)/Google/Chrome/Application/chrome.exe"
   },
    {
      "browserName": "internet explorer",
      "maxInstances": 1,
      "platform": "WINDOWS",
```

```
"webdriver.ie.driver": "C:/Program Files (x86)/Internet Explorer/iexplore.exe"
}
],
"configuration": {
    "_comment" : "Configuration for Node",
    "cleanUpCycle": 2000,
    "timeout": 30000,
    "proxy": "org.openqa.grid.selenium.proxy.WebDriverRemoteProxy",
    "port": 5555,
    "host": ip,
    "register": true,
    "hubPort": 4444,
    "maxSessions": 5
}
```

ノードをする

ノードをするには、まずハブをするがあります。ハブがないは、のようにできます。

java -jar selenium-server-standalone-<version>.jar -role hub

に、ノードをすることができます。

java -jar selenium-server-standalone-<version>.jar -role node -hub http://localhost:4444/grid/register // default port is 4444

はこちら https://github.com/SeleniumHQ/selenium/wiki/Grid2

オンラインでセレングリッドをむ https://riptutorial.com/ja/selenium-webdriver/topic/1359/セレン グリッド

18: セレングリッド

## き

Selenium Gridは、さまざまなテストデバイスにしてテストをするためのフレームワークです。これは、Webアプリケーションのテストにされます。 C、Groovy、Java、Perl、PHP、Python、 Rubyなどのなプログラミングでテストをくことがです。このテストは、Windows、Linux、OS X などのプラットフォームのさまざまなウェブブラウザーにしてできます。

これはApache 2.0のライセンスでされているオープンソースのソフトウェアです。Webはでダウンロードしてできます。

• jarファイルをするために、はすべてのjarファイルのです

• java -jar <jar-file-full-name>.jar -<your parameters if any>

パラメーター

パラメ <b>ーター</b>	
	それは、それが <sub>hub</sub> か <sub>node</sub> であったかをセレンにえますか
	これは、 <sub>hub</sub> または <sub>node</sub> がリスニングするポートをするためです。
ハブ	このパラメータは、 $_{ m node}$ ハブURLをするためにされ $_{ m node}$
browserName	firefox、chrome、Internet Explorerのようなブラウザをするために <sub>node</sub> され ています
maxInstances	そのブラウザのインスタンスがどこでされているか。 5は、ユ―ザ―がした ブラウザのインスタンスが5つすることをします。
nodeConfig	ノードのJsonファイル。、ポートなどをここでできます
hubConfig	ノードのJsonファイル。ここでは、、ポート、インスタンスなどをできます

# **Examples**

Selenium Grid<sup></sup> Java ⊐−<sup>ド</sup>

```
String hubUrl = "http://localhost:4444/wd/hub"
DesiredCapabilities capability = DesiredCapabilities.firefox(); //or which browser you want
RemoteWebDriver driver = new RemoteWebDriver(hubUrl, capability);
```

Selenium Gridハブとノードの

# ハブをする

セレングリッドのハブとノードのセットアップのためのな。は、 Grid 2のドキュメントをしてく ださい。

グリッドハブをするには、フロ—がです。

• Selenium-server-standalone-.jar

ハブの

ハブをするには、セレンサ—バ—をするがあります。

- 1. Selenium-server-standalone-.jarのダウンロード
- 2. ターミナルをき、Selenium-server-standalone-.jarがあるフォルダにします。
- 3. following  $\neg$   $\neg$   $\vee$   $\lor$   $\delta$   $\lor$   $\delta$   $\lor$   $\delta$ 
  - 1.  $\vec{\tau}$ 7J $\nu$   $\wedge$   $\mathcal{O}_{java}$  -jar selenium-server-standalone-<Version>.jar -role hub
  - 2. Json  $\mathcal{O}_{java}$  -jar selenium-server-standalone-<Version>.jar -role hub -hubConfig hubConfig.json
- 4. http:// localhost4444をく/のメッセージがされます。

(+) (i) localhost:4444

You are using grid 2.52.0 Find help on the official selenium wiki : <u>more help here</u> default monitoring page : <u>console</u>

console - > View configをクリックして、ハブののをします。

# ノードの

グリッドハブをするには、フロ—がです。

- Selenium-server-standalone-.jar
- Webdrivers
  - Chromeのドライバー
  - FireFoxドライバ
  - Microsoft Edgeドライバ
- ブラウザ
  - 。クロム
  - FireFox

• Microsoft Edge Windows 10

、ハブのノードをするには

1. Selenium-server-standalone-.jarのダウンロード

2. テストしたいブラウザをダウンロードする

- 3. テストしたいブラウザのドライバをダウンロードする
- 4. しいをき、selenium server jarファイルのにします。
- - 1. デフォルトの<sub>java</sub> -jar selenium-server-standalone-<VERSION NUMBER>.jar -role node
  - 2. Json  $\mathcal{D}_{java jar}$  selenium-server-standalone-<Version>.jar -role node -nodeConfig nodeConfig.json
- 6. に、 http:// localhost4444 / grid / consoleにアクセスして、ノードのをします

Jsonのコンフィグレーション

#### ハブの

java -jar selenium-server-standalone-<version>.jar -role hub -hubConfig hubConfig.json

```
{
    "_comment" : "Configuration for Hub - hubConfig.json",
    "host": ip,
    "maxSessions": 5,
    "port": 4444,
    "cleanupCycle": 5000,
    "timeout": 300000,
    "newSessionWaitTimeout": -1,
    "servlets": [],
    "prioritizer": null,
    "capabilityMatcher": "org.openqa.grid.internal.utils.DefaultCapabilityMatcher",
    "throwOnCapabilityNotPresent": true,
    "nodePolling": 180000,
    "platform": "WINDOWS"
}
```

#### ノードの

java -jar selenium-server-standalone-<version>.jar -role node -nodeConfig nodeConfig.json

```
{
   "capabilities":
   [
        "browserName": "opera",
        "platform": "WINDOWS",
        "maxInstances": 5,
        "seleniumProtocol": "WebDriver",
        "webdriver.opera.driver": "C:/Selenium/drivers/operadriver.exe",
        "binary":"C:/Program Files/Opera/44.0.2510.1159/opera.exe"
```

```
},
  {
    "browserName": "chrome",
    "platform": "WINDOWS",
    "maxInstances": 5,
    "seleniumProtocol": "WebDriver",
    "webdriver.chrome.driver": "C:/Selenium/drivers/chromedriver.exe",
    "binary":"C:/Program Files/Google/Chrome/Application/chrome.exe"
  },
  {
    "browserName": "firefox",
    "platform": "WINDOWS",
    "maxInstances": 5,
    "seleniumProtocol": "WebDriver",
    "webdriver.gecko.driver": "C:/Selenium/drivers/geckodriver.exe",
    "binary":"C:/Program Files/Mozilla Firefox/firefox.exe"
 }
],
"proxy": "org.openqa.grid.selenium.proxy.DefaultRemoteProxy",
"maxSession": 5,
"port": 5555,
"register": true,
"registerCycle": 5000,
"hub": "http://localhost:4444",
"nodeStatusCheckTimeout": 5000,
"nodePolling": 5000,
"role": "node",
"unregisterIfStillDownAfter": 60000,
"downPollingLimit": 2,
"debug": false,
"servlets" : [],
"withoutServlets": [],
"custom": {}
```

オンラインでセレングリッドをむ https://riptutorial.com/ja/selenium-webdriver/topic/2504/セレン グリッド

}

# 19: セレンのロボット

- int ms
- keyPressint keycode
- keyReleaseint keycode
- mouseMoveint x int y
- mousePressintボタン
- mouseReleaseintボタン
- mouseWheelint wheelAmt

パラメ—タ—

パラメ <u>ー</u> タ	
ミズ	ミリでスリープする
キ—コ— ド	されたキ―をして <sub>A</sub> ⊐―ドをすなどのは <sub>VK_A</sub> です。は https://docs.oracle.com/javase/7/docs/api/java/awt/event/KeyEvent.htmlをしてく ださい。
х∝ у	スクリーンコーディネート
ボタン	Buttonマスク。1つまたはのマウスボタンマスクのみわせ
wheelAmt	マウスホイ―ルをするためのノッチの、ユ―ザ―にかって/にするためのの

このセクションでは、Selenium WebdriverをしたRobot APIのについてしくします。 Robotクラス は、セレンがマウスのキーをす、F1キーをすなどのをうことができないにネイティブシステムを するためにされます。

# **Examples**

ロボット APIJAVA をした Keypress イベント

```
import java.awt.AWTException;
import java.awt.Robot;
import java.awt.event.KeyEvent;
public class KeyBoardExample {
    public static void main(String[] args) {
        try {
            Robot robot = new Robot();
            robot.delay(3000);
            robot.keyPress(KeyEvent.VK_Q); //VK_Q for Q
```

```
} catch (AWTException e) {
    e.printStackTrace();
  }
}
```

セレニウム

Webアプリケーションでキープレスイベントをテストするために、のキーをすがあることがあり ます。ログインフォームのEnterキーをテストするインスタンスでは、Selenium WebDriverでのよ うにできます

```
import java.awt.AWTException;
import java.awt.Robot;
import java.awt.event.KeyEvent;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.Test;
public class LoginTest {
    @Test
    public void testEnterKey() throws InterruptedException
    {
        WebDriver driver=new FirefoxDriver();
       Robot robot=null;
        driver.get("test-url");
        driver.manage().window().maximize();
        driver.findElement(By.xpath("xpath-expression")).click();
        driver.findElement(By.xpath("xpath-expression")).sendKeys("username");
        driver.findElement(By.xpath("xpath-expression")).sendKeys("password");
        try {
           robot=new Robot();
        } catch (AWTException e) {
            e.printStackTrace();
        }
        //Keyboard Activity Using Robot Class
        robot.keyPress(KeyEvent.VK_ENTER);
   }
}
```

## ロボット APIJAVA をしたマウスイベント

#### マウスのき

```
import java.awt.Robot;
public class MouseClass {
  public static void main(String[] args) throws Exception {
    Robot robot = new Robot();
    // SET THE MOUSE X Y POSITION
    robot.mouseMove(300, 550);
    }
}
```

#### マウスの/ボタンをす

```
import java.awt.Robot;
import java.awt.event.InputEvent;
public class MouseEvent {
    public static void main(String[] args) throws Exception {
        Robot robot = new Robot();
        // LEFT CLICK
        robot.mousePress(InputEvent.BUTTON1_MASK);
        robot.mouseRelease(InputEvent.BUTTON1_MASK);
        // RIGHT CLICK
        robot.mousePress(InputEvent.BUTTON3_MASK);
        robot.mouseRelease(InputEvent.BUTTON3_MASK);
        }
    }
}
```

ホイールをクリックしてスクロールします

```
import java.awt.Robot;
import java.awt.event.InputEvent;
public class MouseClass {
  public static void main(String[] args) throws Exception {
    Robot robot = new Robot();
    // MIDDLE WHEEL CLICK
    robot.mousePress(InputEvent.BUTTON3_DOWN_MASK);
    robot.mouseRelease(InputEvent.BUTTON3_DOWN_MASK);
    // SCROLL THE MOUSE WHEEL
    robot.mouseWheel(-100);
    }
}
```

オンラインでセレンのロボットをむ https://riptutorial.com/ja/selenium-webdriver/topic/4877/セレ ンのロボット

# 20: ナビゲーション

- C
- void Back
- void Forward
- void GotToUrlurl
- void Refresh
- Python
- driver.back
- driver.forward
- driver.get "URL"
- driver.refresh
- Java
- driver.navigate back;
- driver.navigate forward;
- driver.navigate "URL";
- driver.navigate refresh;

# **Examples**

Navigate[C]

0

すべてのブラウザでなのツ―ルバ―コマンドをするなど、ブラウザをナビゲ―トすることはです

 $\leftarrow \rightarrow C$ (i) stackoverflow.com/d

### ドライバで $_{Navigate()}$ をびすと、ナビゲーションオブジェクトをできます。

```
IWebDriver driver
INavigation navigation = driver.Navigate();
```

## ナビゲーションオブジェクトをすると、Webでブラウザをナビゲートするさまざまなアクション をできます。

```
//like pressing the back button
navigation.Back();
//like pressing the forward button on a browser
navigation.Forward();
//navigate to a new url in the current window
navigation.GoToUrl("www.stackoverflow.com");
//Like pressing the reload button
navigation.Refresh();
```

## Navigate[Java]

### のURLにするには

driver.navigate().to("http://www.example.com");

### にする

driver.navigate().back();

### しの

driver.navigate().forward();

ページをリフレッシュするには

driver.navigate().refresh();

WebDriverのブラウザメソッド

WebDriver、されたWebブラウザをす、テストにするなインターフェイス。このクラスのメソッドは、の3つのカテゴリにされます。

- ブラウザの
- WebElementsの
- デバッグ

なメソッドは、しいWebページをみむためにされるgetStringと、WebElementsをつけるためにされるfindElementByとのさまざまなメソッドです。このでは、ブラウザのをびます。する

void get(java.lang.String url)

のブラウザウィンドウにしいWebページをみみます。これはHTTP GETをしてわれ、メソッドは ロードがするまでブロックされます。このタイムアウトがするまでつことがです。これは、テス トがされているにとなるページがされたに、このインターフェイスにするのびしのが、しくみま れたページにしてわれるからです。

```
//Initialising driver
WebDriver driver = new FirefoxDriver();
//setting timeout for page load
driver.manage().timeouts().pageLoadTimeout(20, TimeUnit.SECONDS);
//Call Url in get method
driver.get("https://www.google.com");
//or
driver.get("https://seleniumhg.org");
```

#### getCurrentUrl

java.lang.String getCurrentUrl()

ブラウザがているURLをすをします。ブラウザにロードされているページのURLをします。

```
//Getting current url loaded in browser & comparing with expected url
String pageURL = driver.getCurrentUrl();
Assert.assertEquals(pageURL, "https://www.google.com");
```

#### getTitle

```
java.lang.String getTitle()
```

とのをりいたで、のページのタイトルをします。まだされていないはnullをします。

```
//Getting current page title loaded in browser & comparing with expected title
String pageTitle = driver.getTitle();
Assert.assertEquals(pageTitle, "Google");
```

getPageSource

java.lang.String getPageSource()

にロードされたページのソースをします。みみにページがされたたとえばJavascriptによって、されたテキストがされたページのものであるというはありません。

```
//get the current page source
String pageSource = driver.getPageSource();
```

#### じる

void close()

いているのウィンドウであれば、ブラウザをしてのウィンドウをじます。そのドライバインスタンスでのウィンドウがかれている、このメソッドはフォーカスをっているウィンドウをじます。

```
//Close the current window
    driver.close();
```

#### する

void quit()

このドライバをし、するすべてのウィンドウをじます。このメソッドをびすと、じドライバイン スタンスをするのメソッドをすることはできません。

//Quit the current driver session / close all windows associated with driver

これらは、にじてブラウザをするためにSelenium 2.0でなになメソッドです。

オンラインでナビゲーションをむ https://riptutorial.com/ja/selenium-webdriver/topic/7272/ナビゲ ーション
# 21: ブラウザウィンドウサイズの/

# き

のブラウザのウィンドウサイズのまたは

- window maximize; driver.manage.
- window setSize DimensionObject;
- driver.manage window getSize

# Examples

JAVA

ブラウザウィンドウサイズにする

```
//Initialize Browser
System.setProperty("webdriver.gecko.driver", "E:\\path\\to\\geckodriver.exe");
WebDriver driver = new FirefoxDriver();
driver.get("https://www.google.com/");
```

//Set Browser window size
driver.manage().window().maximize();

### のウィンドウサイズをする

```
//Initialize Browser
System.setProperty("webdriver.gecko.driver", "E:\\path\\to\\geckodriver.exe");
WebDriver driver = new FirefoxDriver();
driver.get("https://www.google.com/");
```

//Initialize Dimension class object and set Browser window size
org.openqa.selenium.Dimension d = new org.openqa.selenium.Dimension(400, 500);
driver.manage().window().setSize(d);

### ブラウザのウィンドウサイズをする

```
//Initialize Browser
System.setProperty("webdriver.gecko.driver", "E:\\path\\to\\geckodriver.exe");
WebDriver driver = new FirefoxDriver();
driver.get("https://www.google.com/");
//Get Browser window size and print on console
```

```
System.out.println(driver.manage().window().getSize());
```

### オンラインでブラウザウィンドウサイズの/をむ https://riptutorial.com/ja/seleniumwebdriver/topic/10093/ブラウザウィンドウサイズの-

22: ブラウザウィンドウとのやりとり

### **Examples**

アクティブなウィンドウの

# С

ウィンドウをする

driver.Manage().Window.Maximize();

これはかなりで、アクティブなウィンドウがされます。

ウィンドウの

driver.Manage().Window.Position = new System.Drawing.Point(1, 1);

ここでは、アクティブなウィンドウをにしいにします。 Pointオブジェクトでは、 xとyをします。これらをのからのオフセットとしてして、ウィンドウのをします。にウィンドウのをすることもできます。

System.Drawing.Point windowPosition = driver.Manage().Window.Position;

ウィンドウのサイズ

ウィンドウサイズのとは、とじをします。

driver.Manage().Window.Size = new System.Drawing.Size(100, 200); System.Drawing.Size windowSize = driver.Manage().Window.Size;

ウィンドウの**URL** 

アクティブウィンドウののURLをできます

string url = driver.Url;

アクティブウィンドウのURLをして、ドライバがしいにナビゲートするようにすることもできま す

driver.Url = "http://stackoverflow.com/";

ウィンドウハンドル

https://riptutorial.com/ja/home

#### のウィンドウのハンドルをできます。

string handle = driver.CurrentWindowHandle;

#### いているすべてのウィンドウのハンドルをできます。

IList<String> handles = driver.WindowHandles;

# Python

#### ウィンドウをする

driver.maximize\_window()

### ウィンドウのをする

driver.get\_window\_position() # returns {'y', 'x'} coordinates

#### ウィンドウのをする

driver.set\_window\_position(x, y) # pass 'x' and 'y' coordinates as arguments

#### ウィンドウのサイズをする

driver.get\_window\_size() # returns {'width', 'height'} values

### ウィンドウのサイズをする

driver.set\_window\_size(width, height) # pass 'width' and 'height' values as arguments

#### のページタイトル

driver.title

#### のURL

driver.current\_url

#### ウィンドウハンドル

driver.current\_window\_handle

いているウィンドウのリスト

のブラウザウィンドウをじる

しいいたタブにりえます。のウィンドウをじますこのはしいタブ。のウィンドウにります。

```
browser.getAllWindowHandles().then(function (handles) {
    browser.driver.switchTo().window(handles[1]);
    browser.driver.close();
    browser.driver.switchTo().window(handles[0]);
});
```

JAVAセレン

```
Set<String> handlesSet = driver.getWindowHandles();
List<String> handlesList = new ArrayList<String>(handlesSet);
driver.switchTo().window(handlesList.get(1));
driver.close();
driver.switchTo().window(handlesList.get(0));
```

のウィンドウをする

# **Python**

もにされるシナリオ

```
    1. しいウィンドウでページをく
    2. それにりえる
    3. かをする
    4. それをじる
    5. ウィンドウにる

# Open "Google" page in parent window driver.get ("https://google.com")
```

```
driver.title # 'Google'
```

```
# Get parent window
parent_window = driver.current_window_handle
```

# Open "Bing" page in child window
driver.execute\_script("window.open('https://bing.com')")

```
# Get list of all windows currently opened (parent + child)
all_windows = driver.window_handles
```

```
# Get child window
child_window = [window for window in all_windows if window != parent_window][0]
```

```
# Switch to child window
```

```
driver.switch_to.window(child_window)
driver.title # 'Bing'
# Close child window
driver.close()
# Switch back to parent window
driver.switch_to.window(parent_window)
driver.title # 'Google'
```

オンラインでブラウザウィンドウとのやりとりをむ https://riptutorial.com/ja/seleniumwebdriver/topic/5181/ブラウザウィンドウとのやりとり

23: ページオブジェクトモデル

### き

WebサイトやWebアプリケーションをするになをたすのは、のをし、それらとすることです。は、ロケータと $_{By}$ クラスをしてSeleniumでされます。これらのロケータとインタラクションは、コードのをけ、メンテナンスをにするためのベストプラクティスとして、Page Objectにされています。これは、 $_{WebElements}$ をカプセルし、ページまたはWebアプリケーションのページのでビヘイビアとをすとします。

ページオブジェクトモデルは、Webページののビューへのインタフェースとしてするオブジェク トクラスをするパターンです。そのページクラスのメソッドをして、なアクションをします。、 たちはテストクラスのWebページのHTMLコードをしていましたが、これはUIのにでするのがに でした。

ただし、ページオブジェクトパターンのでコードをすると、アプリケーションのAPIがされ、 HTMLをりげることなくページをできるようになります。のなRueによれば、あなたのページオブ ジェクトには、がそのWebページでうことができるすべてがあるはずです。たとえば、Webペー ジのテキストフィールドにアクセスするには、すべてのをったにテキストをしてをすメソッドが です。

ページオブジェクトをするにすべきなはほとんどありません。

- ページオブジェクトは、はページにのみすべきではありませんが、ページのなにすることを おめします。たとえば、のなるをするのタブをつページは、タブのとじのページをつがあり ます。
- 2. あるビューからのビューにすると、ページクラスのインスタンスがされます。
- のビューまたはWebページにしてのみするがあるユーティリティメソッドは、そのページク ラスにのみしているがあります。
- アサーションメソッドはページクラスによってをうべきではありません。ブールをすメソッドをつことはできますが、そこではしないでください。たとえば、ユーザーのフルネームをするには、ブールをするメソッドをできます。

```
public boolean hasDisplayedUserFullName (String userFullName) {
    return driver.findElement(By.xpath("xpathExpressionUsingFullName")).isDisplayed();
}
```

5. あなたのウェブページがiframeにづいているは、iframeのページクラスもつことをおめしま す。

ページオブジェクトパターンの

1. テストコードとページコードをに

- WebページのUIがされたは、のでコードをするはありません。ページクラスでのみします。
   するロケータはありません。
- 4. コードをしやすくする
- 5. なメンテナンス

## **Examples**

はじめにJavaの

Pageオブジェクトパタ―ンにづいてログインテストをする

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
/**
\star Class which models the view of Sign-In page
 */
public class SignInPage {
    @FindBy(id="username")
    private usernameInput;
    @FindBy(id="password")
    private passwordInput;
    @FindBy(id="signin")
    private signInButton;
    private WebDriver driver;
    public SignInPage(WebDriver driver) {
       this.driver = driver;
    }
    /**
     * Method to perform login
     */
    public HomePage performLogin(String username, String password) {
        usernameInput.sendKeys(username);
       passwordInput.sendKeys(password);
       signInButton.click();
        return PageFactory.initElements(driver, HomePage.class);
    }
}
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
/**
 \star Class which models the view of home page
 */
```

```
public class HomePage {
   @FindBy(id="logout")
   private logoutLink;
   private WebDriver driver;
   public HomePage(WebDriver driver) {
       this.driver = driver;
    }
    /**
    * Method to log out
    */
   public SignInPage logout() {
       logoutLink.click();
       wait.ForPageToLoad();
       return PageFactory.initElements(driver, SignInPage.class);
   }
}
/**
 * Login test class
*/
public class LoginTest {
  public void testLogin() {
       SignInPage signInPage = new SignInPage(driver);
        HomePage homePage = signInPage.login(username, password);
       signInPage = homePage.logout();
   }
}
```

### С

ページオブジェクトには、、アサーションのをすがあり、によってはのページレディステートメ ソッドのメソッドをめるがあります。 Seleniumは、をしてページオブジェクトをサポートしてい ます。 Cではのようになります

```
using OpenQA.Selenium;
using OpenQA.Selenium.Support.PageObjects;
using OpenQA.Selenium.Support.UI;
using System;
using System.Collections.Generic;
public class WikipediaHomePage
{
   private IWebDriver driver;
   private int timeout = 10;
   private By pageLoadedElement = By.ClassName("central-featured-logo");
    [FindsBy(How = How.Id, Using = "searchInput")]
    [CacheLookup]
    private IWebElement searchInput;
    [FindsBy(How = How.CssSelector, Using = ".pure-button.pure-button-primary-progressive")]
    [CacheLookup]
    private IWebElement searchButton;
    public ResultsPage Search(string query)
```

```
{
    searchInput.SendKeys(query);
    searchButton.Click();
}
public WikipediaHomePage VerifyPageLoaded()
{
    new WebDriverWait(driver, TimeSpan.FromSeconds(timeout)).Until<bool>((drv) => return
drv.ExpectedConditions.ElementExists(pageLoadedElement));
    return this;
    }
}
```

ノート

- CacheLookupは、キャッシュにをし、びしごとにしいをしてします。これによりパフォーマンスはしますが、にするにはしていません。
- searchButtonは2つのクラスとIDはありません。そのため、クラスやIDはできません。
- デベロッパーツールChrome、FireBugなどをできるのブラウザーでロケーターがにすがある ことをしました。
- $_{Search()}$ クリックはのページにリダイレクトされるので、  $_{Search()}$ メソッドはのページオブ ジェクト  $_{ResultsPage}$ をします。

ベストプラクティスペ**―ジオブジェクトモデル** 

- ヘッダーとフッターに々のファイルをしますすべてのページにであり、1ページのにするの はがありません
- の/る/のようにを々のファイルにしますアイデアはあらゆるのをし、をにつことです
- Driverについては、のDriverクラスをしてDriverをにして、すべての $^{-}$ ジにアクセスできるようにすることをおめします。のすべてのWeb $^{-}$ ジでDriverClassがされています
- PageObjectsでされているは、とそれがびされるをにいてできるりさなチャンクにされてい ますログインはenterUsernameとenterPasswordにされますが、くの、enterUsernameと enterPasswordへのびしをするのではなく、Loginがびされるため、Loginはよりです
- PageObjectsをすると、elementLocatorsからTestスクリプトがされます。
- のutilsフォルダえば、DateUtil、excelUtilsなどでユーティリティをする
- のconfフォルダにするテストのの、フォルダとフォルダのなど
- にscreenCaptureをみむ
- DriverClassになをいくつかのなをいたようにしてください。にwait.untilExpectedConditions のようななではなく、きのをしてください。これにより、がをにくしていないことがされます。

オンラインでページオブジェクトモデルをむ https://riptutorial.com/ja/seleniumwebdriver/topic/4853/ページオブジェクトモデル



- オブジェクトExecuteAsyncScriptスクリプト、paramsオブジェクト[] args;
- オブジェクトExecuteScriptスクリプト、paramsオブジェクト[] args;

## **Examples**

### С

でJavaScriptをするために<sub>IWebDriver</sub>インスタンスあなたはキャストするが<sub>IWebDriver</sub>しいインタ ーフェースに<sub>IJavaScriptExecutor</sub>

```
IWebDriver driver;
IJavaScriptExecutor jsDriver = driver as IJavaScriptExecutor;
```

これで<sub>IJavaScriptExecutor</sub>インスタンスでできるすべてのメソッドにアクセスできるようになり、 Javascriptをできます。たとえば、<sub>IJavaScriptExecutor</sub>ようになります。

jsDriver.ExecuteScript("alert('running javascript');");

### **Python**

PythonでJavascriptをするには、 execute\_script ("javascript script here")ます。 execute\_script はwebdriverインスタンスでびされ、なjavascriptにすることができます。

```
from selenium import webdriver
driver = webdriver.Chrome()
driver.execute_script("alert('running javascript');")
```

Java

```
JavaでJavascriptをするには、JavascriptをサポートするしいWebdriverをします。

executeScript()をするには、ドライバをJavascriptExecutorにキャストするか、しいをキャストさ

れたドライバのにするか、((JavascriptExecutor)driver)。 driver.executeScript()はなJavascript

であるStringをけります。
```

```
WebDriver driver = new ChromeDriver();
JavascriptExecutor JavascriptExecutor = ((JavascriptExecutor)driver);
JavascriptExecutor.executeScript("alert('running javascript');");
```

ルビー

require "selenium-webdriver"

driver = Selenium::WebDriver.for :chrome
driver.execute\_script("alert('running javascript');")

オンラインでページでJavascriptをするをむ https://riptutorial.com/ja/seleniumwebdriver/topic/6986/ページでjavascriptをする

25: ヘッドレスブラウザ

## **Examples**

PhantomJS [C]

PhantomJSは、JavaScriptをサポートしたフルのヘッドレスWebブラウザです。

めるに、 PhantomJSドライバをダウンロードして、コードのにれてください

```
using OpenQA.Selenium;
using OpenQA.Selenium.PhantomJS;
```

、に

```
var driver = new PhantomJSDriver();
```

これにより、にPhantomJSDriverクラスのしいインスタンスがされます。あなたはそれをのよう なすべてのWebDriverとじでうことができます

```
using (var driver = new PhantomJSDriver())
{
    driver.Navigate().GoToUrl("http://stackoverflow.com/");
    var questions = driver.FindElements(By.ClassName("question-hyperlink"));
    foreach (var question in questions)
    {
        // This will display every question header on StackOverflow homepage.
        Console.WriteLine(question.Text);
    }
}
```

これはにします。しかし、おそらくするは、UIを $_{PhantomJS}$ でする、  $_{PhantomJS}$ はしいコンソールウィンドウをきます。これはほとんどのには $_{PhantomJS}$ ありません。いにも、々はウィンドウをにすることができ、そしてしでもしてパフォーマンス $_{PhantomJSOptions}$ 、そして $_{PhantomJSDriverService}$ 。のな

```
// Options are used for setting "browser capabilities", such as setting a User-Agent
// property as shown below:
var options = new PhantomJSOptions();
options.AddAdditionalCapability("phantomjs.page.settings.userAgent",
"Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:25.0) Gecko/20100101 Firefox/25.0");
// Services are used for setting up the WebDriver to your likings, such as
// hiding the console window and restricting image loading as shown below:
var service = PhantomJSDriverService.CreateDefaultService();
service.HideCommandPromptWindow = true;
service.LoadImages = false;
```

```
// The same code as in the example above:
using (var driver = new PhantomJSDriver(service, options))
{
    driver.Navigate().GoToUrl("http://stackoverflow.com/");
    var questions = driver.FindElements(By.ClassName("question-hyperlink"));
    foreach (var question in questions)
    {
        // This will display every question header on StackOverflow homepage.
        Console.WriteLine(question.Text);
    }
}
```

プロのヒントクラス PhantomJSDriverService をクリックし、F12 PhantomJSDriverServiceをしてがに されているかどうかをにします。

SimpleBrowser [C]

simpleBrowserは、JavaScriptをサポートしないWebDriverです。

の<sub>PhantomJS</sub>よりもはるかにですが、にしては、ファンシーなをたないなタスクにされています。

まず、 SimpleBrowser.WebDriverパッケージをダウンロードし、このコードをにくがあります

using OpenQA.Selenium; using SimpleBrowser.WebDriver;

さて、これをするのなをにします

```
using (var driver = new SimpleBrowserDriver())
{
    driver.Navigate().GoToUrl("http://stackoverflow.com/");
    var questions = driver.FindElements(By.ClassName("question-hyperlink"));
    foreach (var question in questions)
    {
        // This will display every question header on StackOverflow homepage.
        Console.WriteLine(question.Text);
    }
}
```

Javaのヘッドレスブラウザ

# **HTMLUnitDriver**

HTMLUnitDriverは、HtmlUnitにづくWebdriverのヘッドレスGUIレスブラウザのでもなです。 HTMLドキュメントをモデルし、のブラウザとに、ページのびし、フォームへの、リンクのクリ ックなどをにするAPIをします。 JavaScriptをサポートし、AJAXライブラリでします。ウェブサ HTMLUnitDriverをしてhttp://stackoverflow.com/からのリストをする。

```
import java.util.List;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.htmlunit.HtmlUnitDriver;
class testHeadlessDriver{
           private void getQuestions() {
                    WebDriver driver = new HtmlUnitDriver();
                    driver.get("http://stackoverflow.com/");
                    driver.manage().timeouts().implicitlyWait(60, TimeUnit.SECONDS);
                    List<WebElement> questions = driver.findElements(By.className("question-
hyperlink"));
                    questions.forEach((question) -> {
                        System.out.println(question.getText());
                    });
                   driver.close();
                }
    }
```

これは、のブラウザMozilla Firefox、Google Chrome、IEとじですが、GUIをたないため、がにされません。

オンラインでヘッドレスブラウザをむ https://riptutorial.com/ja/selenium-webdriver/topic/3931/ヘッドレスブラウザ



# **Examples**

JUnit

```
するためにJUnitをしているは、 TestWatcher クラスをできます。
```

```
public class TestRules extends TestWatcher {
    @Override
    protected void failed(Throwable e, Description description) {
        // This will be called whenever a test fails.
    }
```

だからあなたのテストクラスでは、にそれをびすことができます

```
public class testClass{
  @Rule
  public TestRules testRules = new TestRules();
  @Test
  public void doTestSomething() throws Exception{
        // If the test fails for any reason, it will be caught be testrules.
  }
```

### **EventFiringWebDriver**

EventFiringWebDriverのWebDriverEventListenerをWebDriverEventListenerにアタッチし、メソッド、つまりonExceptionメソッドをオーバーライドすることができます。

```
EventFiringWebDriver driver = new EventFiringWebDriver(new FirefoxDriver());
WebDriverEventListener listener = new AbstractWebDriverEventListener() {
    @Override
    public void onException(Throwable t, WebDriver driver) {
        // Take action
    }
};
driver.register(listener);
```

オンラインでリスナーをむ https://riptutorial.com/ja/selenium-webdriver/topic/8226/リスナー

# 27: なSelenium Webdriverプログラム

## き

このトピックでは、C、Groovy、Java、Perl、PHP、Python、Rubyなどのセレンサポートのな Webドライバプログラムをします。

Journeyには、ブラウザのドライバをく - > Google ~ ジ - > ブラウザをシャットダウンする

# **Examples**

### С

の「プログラム」はGoogleホームページにナビゲートし、ページをにみんだにブラウザをします。

using (var driver = new ChromeDriver())

NuGetをしてWebDriverをダウンロードしていないは、ドライバのchromedriver.exeがかれている ディレクトリへのパスでをすがあります。

driver.Navigate().GoToUrl("http://www.google.com");

### そして

driver.Url = "http://www.google.com";

これらのはどちらもじことをします。ドライバはのURLにし、ページがロードされるのをってからのステートメントにするようにします。

Back()、Forward()、Refresh()など、ナビゲーションにするのメソッドがあります。

その、<sub>using</sub>ブロックはにし、オブジェクトをします。

#### Python

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

def set_up_driver():
    path_to_chrome_driver = 'chromedriver'
    return webdriver.Chrome(executable_path=path_to_chrome_driver)

def get_google():
    driver = set_up_driver()
    driver.get('http://www.google.com')
    tear_down(driver)

def tear_down(driver):
    driver.quit()

if '__main__' == __name__:
```

```
get_google()
```

の「プログラム」はGoogleのホームページにし、するにブラウザをします。

```
if '__main__' == __name__:
    get_google()
```

に、<sub>get\_google()</sub>をびすプログラムへのポイントというながあります。

```
def get_google():
    driver = set_up_driver()
```

 $get_google()$   $\mathcal{E}$   $\mathcal{O}$   $\mathcal{O}$ 

```
def set_up_driver():
    path_to_chrome_driver = 'chromedriver'
    return webdriver.Chrome(executable_path=path_to_chrome_driver)
```

chromedriver.exeがどこにあるかをchromedriver.exe、ドライバオブジェクトをこのパスでインス

driver.get('http://www.google.com')

そしてドライバオブジェクトをす<sub>tear\_down()</sub>をびします

tear\_down(driver)

tear\_down()にはドライバオブジェクトをシャットダウンするためのが1つだけまれています

driver.quit()

これはドライバーに、いているブラウザーウィンドウをすべてじ、ブラウザーオブジェクトをするようにします。このびしのにのコードがないので、プログラムをにします。

Java

のコードはすべて3つのステップです。

- 1. クロームブラウザをく
- 2. Googleのページをく
- 3. ブラウザをシャットダウンする

```
import org.openqa.selenium;
import org.openqa.selenium.chrome;
public class WebDriverTest {
    public static void main(String args[]) {
        System.setProperty("webdriver.chrome.driver", "C:\\path\\to\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("http://www.google.com");
        driver.quit();
    }
}
```

の「プログラム」はGoogleのホームページにし、するにブラウザをします。

System.setProperty("webdriver.chrome.driver", "C:\\path\\to\\chromedriver.exe");
WebDriver driver = new ChromeDriver();

のは、  $_{ChromeDriver}$  chromedriver.exeファイルがどこにあるかをシステムにします。に、  $_{ChromeDriver()}$ コンストラクタをびしてドライバオブジェクトをします。ここでも、のブラウザ/プ ラットフォームコンストラクタをびすことができます。

driver.get("http://www.google.com");

これは、ドライバにされたURL http://www.google.comにするようにします。 Java WebDriver

APIはWebDriverインターフェイスにget()メソッドをしますが、navigate()メソッド driver.navigate.back()してさらなるナビゲーションメソッドをつけることができます。

ページのみみがしたら、すぐにのびしをいます。

driver.quit();

これは、このびしのにのコードがないので、いているすべてのブラウザウィンドウをじてドライ バオブジェクトをするようにドライバにします。

driver.close();

このでは、がじをquit()をびすとなるのウィンドウしかたないので、このでは、アクティブなウィンドウのみをじるようにここにはされていませんがドライバにあります。

```
Java - ページクラスをしたベストプラクティス
```

ユースケースFBアカウントにログイン

```
import org.openqa.selenium.WebDriver;
import org.openga.selenium.firefox.FirefoxDriver;
import org.testng.annotations.BeforeClass;
import org.testng.annotations.Test;
public class FaceBookLoginTest {
   private static WebDriver driver;
   HomePage homePage;
   LoginPage loginPage;
   @BeforeClass
   public void openFBPage() {
        driver = new FirefoxDriver();
        driver.get("https://www.facebook.com/");
       loginPage = new LoginPage(driver);
    }
    @Test
   public void loginToFB() {
      loginPage.enterUserName("username");
       loginPage.enterPassword("password");
      homePage = loginPage.clickLogin();
       System.out.println(homePage.getUserName());
    }
```

#### ページクラスログインページとホームページログインページクラス

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
public class LoginPage {
    WebDriver driver;
```

```
public LoginPage(WebDriver driver) {
    this.driver = driver;
}
@FindBy(id="email")
private WebElement loginTextBox;
@FindBy(id="pass")
private WebElement passwordTextBox;
@FindBy(xpath = ".//input[@data-testid='royal_login_button']")
private WebElement loginBtn;
public void enterUserName(String userName) {
    if(loginTextBox.isDisplayed()) {
        loginTextBox.clear();
        loginTextBox.sendKeys(userName);
    }
    else{
        System.out.println("Element is not loaded");
    }
}
public void enterPassword(String password) {
    if(passwordTextBox.isDisplayed()) {
        passwordTextBox.clear();
        passwordTextBox.sendKeys(password);
    }
    else{
        System.out.println("Element is not loaded");
    }
}
public HomePage clickLogin() {
    if(loginBtn.isDisplayed()) {
       loginBtn.click();
    }
    return new HomePage(driver);
}
```

ホームページクラス

}

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
public class HomePage {
    WebDriver driver;
    public HomePage(WebDriver driver) {
        this.driver = driver;
    }
    @FindBy(xpath=".//a[@data-testid='blue_bar_profile_link']/span")
    private WebElement userName;
    public String getUserName() {
        if(userName.isDisplayed()) {
            return userName.getText();
        }
        else {
            return "Username is not present";
```

		}	
	}		
}			

オンラインでなSelenium Webdriverプログラムをむ https://riptutorial.com/ja/seleniumwebdriver/topic/3990/なselenium-webdriverプログラム



## **Examples**

Selenium WebDriverでするタイプ

のWebアプリケーションをしているは、みみをするがあります。コードがまだロードされていないにアクセスしようとすると、WebDriverはをスローし、スクリプトはします。

Waitには3つのタイプがあります -

- *の*
- な
- なち

なはのでのみされるのにし、のはプログラムでをするためにされます。

 $\mathcal{O}$ 

なとは、WebDriverに、すぐにでないをつけようとすると、の、DOMをポーリングするようにす ることです。のとは、に、されたWebがWebDriverがしているものがないに、WebDriverにしたい をするです。のは0です。されると、のがWebDriverオブジェクトインスタンスのにされます。な は、のスニペットをしてコードのインスタンスでされます。

#### Javaの

driver.manage().timeouts().implicitlyWait(15, TimeUnit.SECONDS);
// You need to import the following class - import java.util.concurrent.TimeUnit;

#### **C**0

driver.Manage().Timeouts().ImplicitlyWait(TimeSpan.FromSeconds(15));

### この、UIDOMでされたができない、WebDriverが15つようにします。

## な

のがみまれるまでにがかかるがあります。このようなのののはがありません。ブラウザがすべて のにしてじだけにし、がくなるからです。なは、いくつかののについてなをバイパスすることに よってここでちます。 なとは、のWebにされたインテリジェントなです。なをすると、にWebDriverにをえることは、 それがするまでにXのをつことです。

なは、WebDriverWaitクラスとExpectedConditionsクラスをしてわれます。のでは、idが usernameのがのコマンドにむにされるまで10つものとします。ここにがあります。

#### Java の

//Import these two packages: import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait; //Declare a WebDriverWait variable. In this example, we will use myWaitVar as the name of the variable. WebDriverWait myWaitVar = new WebDriverWait(driver, 30); //Use myWaitVar with ExpectedConditions on portions where you need the explicit wait to occur. In this case, we will use explicit wait on the username input before we type the text tutorial onto it. myWaitVar.until(ExpectedConditions.visibilityOfElementLocated(By.id("username")));

driver.findElement(By.id("username")).sendKeys("tutorial");

ExpectedConditionsクラスには、をつためのいくつかのされたがあります。 Javaバインディング におけるこれらののリストをするには、 ここをクリックしてください。

#### $\mathbf{C}$

```
using OpenQA.Selenium;
using OpenQA.Selenium.Support.UI;
using OpenQA.Selenium.PhantomJS;
// You can use any other WebDriver you want, such as ChromeDriver.
using (var driver = new PhantomJSDriver())
{
    driver.Navigate().GoToUrl("http://somedomain/url_that_delays_loading");
    // We aren't going to use it more than once, so no need to declare this a variable.
    new WebDriverWait(driver, TimeSpan.FromSeconds(10))
       .Until(ExpectedConditions.ElementIsVisible(By.Id("element-id")));
    // After the element is detected by the previous Wait,
    // it will display the element's text
    Console.WriteLine(driver.FindElement(By.Id("element-id")).Text);
}
```

このでは、がされるまで10します。がタイムアウトにされませんは、webdriverをがスロ―されま すWebDriverTimeoutException。

が10のタイムアウトにされている、システムはすぐにをします。

### な

およびなとはなり、なは2つのパラメ―タをします。タイムアウトとポ―リング。たとえば、タイムアウトが30、ポ―リングが2であるとします。 WebDriverは、タイムアウト30まで2ごとにをチェックします。のないタイムアウトをえた、がスロ―されます。は、なのをすサンプルコ―ドです。

#### Javaの

```
Wait wait = new FluentWait(driver).withTimeout(30, SECONDS).pollingEvery(2,
SECONDS).ignoring(NoSuchElementException.class);
WebElement testElement = wait.until(new Function() {
    public WebElement apply(WebDriver driver) {
        return driver.findElement(By.id("testId"));
    }
});
```

なをするもう1つのは、にのタイプのNoSuchElementExceptionsをできることです。これらのす べてののため、AJAXアプリケーションやのみみがにするシナリオでは、ながちます。なのなは、 のりみをにします。

#### さまざまなのな

なでは、がすることをします。たとえば、がクリックになるまでしたいとします。

ここにいくつかのなのデモンストレーションがあります。

これらすべてのでは、  $_{classname}$ 、  $_{xpath}$ 、  $_{link text}$ 、  $_{tag name}$ 、  $_{cssSelector}$ などのロケータとして  $_{By}$ をできます。

## がされるまでちます

たとえば、Webサイトのみみにがかかるは、ページのみみがし、WebDriverにがされるまでつことができます。

#### С

```
WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
wait.Until(ExpectedConditions.ElementIsVisible(By.Id("element-id")));
```

#### Java

```
WebDriverWait wait = new WebDriverWait(driver, 10);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("element-id")));
```

## がもうえなくなるまでつ

#### とじですが、です。

#### С

```
WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
wait.Until(ExpectedConditions.InvisibilityOfElementLocated(By.Id("element-id")));
```

#### Java

```
WebDriverWait wait = new WebDriverWait(driver, 10);
wait.until(ExpectedConditions.invisibilityOfElementLocated(By.id("element-id")));
```

# されたにテキストがするまでちます。

#### С

```
IWebElement element = driver.FindElement(By.Id("element-id"));
```

```
WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromSeconds(10));
wait.Until(ExpectedConditions.TextToBePresentInElement(element, "text"));
```

#### Java

```
WebElement element = driver.findElement(By.id("element-id"));
```

```
WebDriverWait wait = new WebDriverWait(driver, 10);
wait.until(ExpectedConditions.textToBePresentInElement(element, "text"));
```

のリンクにくと、そこにすべてのちがされます。

これらのののいは、パラメ--タにあります。

つまり、WebElementのパラメータがWebElementの、WebElementをすがあることをします。パ ラメータとしてByロケータをする、ロケータをすがあります。

どんなのをするかをにしてください。

Ajaxリクエストのをつ

# С

using OpenQA.Selenium using OpenQA.Selenium.Chrome; using System.Threading;

```
namespace WebDriver Tests
{
    class WebDriverWaits
    {
        static void Main()
        {
            IWebDriver driver = new ChromeDriver(@"C:\WebDriver");
            driver.Navigate().GoToUrl("page with ajax requests");
            CheckPageIsLoaded (driver);
            // Now the page is fully loaded, you can continue with further tests.
        }
        private void CheckPageIsLoaded(IWebDriver driver)
        {
            while (true)
                bool ajaxIsComplete = (bool)(driver as
IJavaScriptExecutor).ExecuteScript("return jQuery.active == 0");
                if (ajaxIsComplete)
                    return;
                Thread.Sleep(100);
            }
       }
   }
}
```

このは、ajaxリクエストがわれたページでにちます。ここではIJavaScriptExecutorをしての JavaScriptコードをします。 whileループのにあるので、ajaxIsComplete == trueになるまでしけ、returnがされます。

ち

#### Java

```
public WebElement apply(WebDriver driver) {
   return element.getText().length() > 0;
  }
});
```

Explicit waitをデフォルトですると、デフォルトでFluentWait<WebDriver>になります

FluentWait<WebDriver> DEFAULT\_SLEEP\_TIMEOUT = 500; NotFoundException  $\pounds$  UnotFoundException  $\circ$ 

な

FluentWaitインスタンスは、をつと、をするをします。さらに、ユーザーは、ページのをすると きにNoSuchElementExceptionsなど、にのタイプのをするようにをすることができます。これは 、ドライバにけられています。

Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
.withTimeout(30, SECONDS) //actuall wait for the element to pe present
.pollingEvery(5, SECONDS) //selenium will keep looking for the element after every 5seconds
.ignoring(NoSuchElementException.class); //while ignoring this condition
wait.until(ExpectedConditions.visibilityOf(driver.findElement(By.id("username"));

オンラインでつをむ https://riptutorial.com/ja/selenium-webdriver/topic/4435/つ

# 29: のフレームをする

き

Webページにはフレームがまれているため、セレンはフレームがセパレートウィンドウなので、 フレームにっているコンテンツにアクセスするとフレームにりえるがあります。くの、フレーム ウィンドウをするフレームをつWebがです。SeleniumはswithToメソッドをしています。

# **Examples**

フレームの

```
<iframe "id="iframe_Login1">
<iframe "id="iframe_Login2">
<iframe "id="iframe_Login3">
</iframe>
```

</iframe>

セレンのフレームにりえるには、swithToメソッドとframeメソッドをします。

driver.switchTo<sup> $\circ$ </sup>  $\neg \lor \neg \lor \neg \downarrow$  iframe\_Login1; driver.switchTo<sup> $\circ$ </sup>  $\neg \lor \neg \lor \neg \downarrow$  iframe\_Login2; driver.switchTo<sup> $\circ$ </sup>  $\neg \lor \neg \lor \neg \downarrow$  iframe\_Login3;

にすには、parentFrameとdefaultContestをします。

parentFrameフォーカスをコンテキストにします。のコンテキストがトップレベルのブラウジン グコンテキストである、コンテキストはされません。

driver.switchTo().parentFrame();

defaultContentページののフレーム、またはiframeがページにまれているときのメインドキュメントをします。

driver.switchTo().defaultContent();

オンラインでのフレームをするをむ https://riptutorial.com/ja/selenium-webdriver/topic/9803/のフレームをする



S. No		Contributors
1	selenium-webdriver をいめる	Abhilash Gupta, Alice, Community, Eugene S, iamdanchiv, Jakub Lokša, Josh, Kishor, Michal, Mohit Tater, Pan, Priyanshu Shekhar, rg702, Santoshsarma, Tomislav Nakic-Alfirevic, vikingben
2	HTMLレポート	Ashish Deshmukh
3	Javaで@FindByアノ テーションをする	Alex Wittig, Łukasz Piaszczyk
4	JavaでSelenium Webdriverをする	r_D, the_coder
5	Python、Ruby、 Javascriptをした Selenium-webdriver とCIツール	Brydenr
6	Selenium e2eセット アップ	Ashish Deshmukh
7	Selenium-WebDriver ග	Brydenr
8	Seleniumをしたオー トメーションのエラ 一	Andersson
9	Webとの	Jakub Lokša, Liam, Moshisho, Siva, Sudha Velan
10	Webの	alecxe, daOnlyBG, Jakub Lokša, Josh, Liam, Łukasz Piaszczyk, Moshisho, NarendraR, noor, Priya, Sakshi Singla, Siva
11	アクションなユ―ザ ―のジェスチャ―を エミュレ―トする	Josh, Kenil Fadia, Liam, Priyanshu Shekhar, Tom Mc
12	アラ―トをする	Andersson, Aurasphere, Priya, SlightlyKosumi
13	クラスを	Gaurav Lad, Liam

14	スイッチングフレ <del>ー</del> ム	Andersson, dreamwork801, Jakub Lokša, Java_deep, Jim Ashworth, Karthik Taduvai, Kyle Fairns, Liam, Iloyd, Priyanshu Shekhar, SlightlyKosumi
15	スクリーンショット をる	Abhilash Gupta, Sanchit
16	スクロ―ル	Andersson, Sagar007
17	セレングリッド	Eugene S, Y-B Cause
18	セレングリッド	mnoronha, Prasanna Selvaraj, selva, Thomas
19	セレンのロボット	Priyanshu Shekhar
20	ナビゲーション	Andersson, Liam, Santoshsarma, viralpatel
21	ブラウザウィンドウ サイズの/	Abhilash Gupta
22	ブラウザウィンドウ とのやりとり	Andersson, Josh, Sakshi Singla
23	ペ―ジオブジェクト モデル	JeffC, Josh, Moshisho, Priyanshu Shekhar, Sakshi Singla
24	ページでJavascript をする	Brydenr, Liam
25	ヘッドレスブラウザ	Abhilash Gupta, Jakub Lokša, Liam, r_D, Tomislav Nakic- Alfirevic
26	リスナ—	Erki M.
27	なSelenium Webdriverプログラ ム	Jakub Lokša, Josh, Liam, Priya, Sudha Velan, Thomas, vikingben
28	つ	Jakub Lokša, Josh, Kenil Fadia, Liam, Moshisho, noor, Sajal Singh, Saurav
29	のフレ―ムをする	Pavan T, Raghvendra