

# Seleniumを使って画面キャプチャを撮ってみる

---

## 概要

複数のブラウザで画面レイアウトが崩れないか確認を行うとき場合、Seleniumを使って自動化するにしてもスクロールがある状態だと、画面キャプチャがうまく取得できなかったため、隙間なく画面キャプチャをとる方法がないかを考えてみる。

## 環境情報

- Java 1.8.191
- Selenium 3.14.0
- ChromeDriver 2.42
- IEDriverServer 3.14

## 試した方法について

WebDriverによって初期表示時の画面サイズが異なるため、単純にスクロール移動させながらスクリーンショットを撮っても、スクリーンショットが取れていない箇所が存在してしまう。

そこで、フルスクリーンで表示させることで、ブラウザの初期サイズに依存せずに画面が表示させることができた。

※最大化させる方法も試したが、ブラウザによって、ステータスバーなどの表示の有無で若干の画面サイズに差が出てしまっていた。

また、表示させる端末を異なる場合、スクロールさせる距離で、スクリーンショットが取れていない箇所場合や、逆に、同じ箇所を複数枚スクリーンショットを撮ってしまった。

そこで、スクロールの移動距離を画面サイズに合わせて移動させることで、無駄なくスクリーンショットを撮ることができた。

SeleniumSample.java

```

/**
 * 指定されたURLにアクセスしスクリーンショットを取得する
 * @param driver WebDriver
 * @param url アクセス先
 * @throws InterruptedException 割り込みが発生した場合に発生
 * @throws IOException スクリーンショットのファイル保存に失敗した場合に発生
 */
public void getScreenShot(WebDriver driver, String url) throws InterruptedException, IOException

    driver.get(url);

    // 最大化
// driver.manage().window().maximize();
// フルスクリーンで表示
driver.manage().window().fullscreen();

    BigDecimal scrollYAfter = BigDecimal.valueOf(-1);

    int count = 1;
    // スクロールできなくなる(スクリーン位置が同じになる)まで繰り返す
    for (BigDecimal scrollYBefore = this.getScrollY(driver);
        (scrollYBefore != null && !scrollYBefore.equals(scrollYAfter));
        scrollYBefore = this.moveScrollY(driver)) {

        File screenFile = ((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);
        this.copy(screenFile, new File(DATA_DIR, "screenshot_" + count + ".png"), true);
        count++;
        scrollYAfter = scrollYBefore;
    }
}

/**
 * 現在の縦スクロール位置を取得する。
 * @param driver WebDriver
 * @return スクロール位置
 */
private BigDecimal getScrollY(WebDriver driver) {
    String scrollY = (String) ((JavascriptExecutor) driver).executeScript( "return String(window
    BigDecimal value = null;
    if (scrollY != null && scrollY.matches("^([0-9]+)$")) {
        value = new BigDecimal(scrollY);
    }
    return value;
}

```

```
/**
 * 画面サイズに合わせてスクロールを移動する
 * @param driver WebDriver
 * @return 移動後のスクロール位置
 */
private BigDecimal moveScrollY(WebDriver driver) {

    // 画面サイズを取得
    int height = driver.manage().window().getSize().height;

    // 見切れて画面確認ができない箇所ができないようにするため、
    // 画面サイズより少し小さい値でスクロールさせる
    String scrollY = (String) ((JavascriptExecutor) driver).executeScript( "scrollBy( 0, " + (height - height) + ");");
    BigDecimal value = null;
    if (scrollY != null && scrollY.matches("^[-9]+.$")) {
        value = new BigDecimal(scrollY);
    }

    return value;
}

/**
 * ファイルをコピーする
 * @param fromFile コピー元ファイル
 * @param toFile コピー先ファイル
 * @param overwrite 上書きフラグ(true:上書きする, false:既に存在した場合、IOExceptionが発生)
 * @throws IOException ファイルコピーに失敗した場合に発生
 */
private void copy(File fromFile, File toFile, boolean overwrite) throws IOException {
    if (fromFile == null) {
        throw new IllegalArgumentException("param[fromFile] is null");
    }
    if (toFile == null) {
        throw new IllegalArgumentException("param[toFile] is null");
    }
    if (!fromFile.exists()) {
        throw new FileNotFoundException("There is not fromFile[" + fromFile.getAbsolutePath() + "]");
    }
    if (toFile.exists() && !overwrite) {
        throw new FileAlreadyExistsException(toFile.getAbsolutePath());
    }

    try (BufferedInputStream in = new BufferedInputStream(new FileInputStream(fromFile))) {
        try (BufferedOutputStream out = new BufferedOutputStream(new FileOutputStream(toFile))) {
            byte[] buff = new byte[1024];
            int read;
            while ((read = in.read(buff)) != -1) {
                out.write(buff, 0, read);
            }
        }
    }
}
```


```
for (int size = in.read(buff); size >= 0; size = in.read(buff)) {
    out.write(buff, 0, size);
}
}
```



# 実行結果

ChromeとIEのブラウザで動作させ、取得したスクリーンショットをつなげてみると、でかぶった箇所もほぼ無い状態で画面すべてのキャプチャを取得することができた。

ChromeDriver	InternetExplorerDriver
<div></div>	

ChromeDriver	InternetExplorerDriver
	

## 今後の課題

横スクロールがある状態や、フレームが分かれてその中にスクロールがある場合について、確認ができていない。