How to download a file with Selenide

Selenide Advent Calendar
Day 10

Good evening!
Today is December - and in this Selenide Advent calendar article we will take a look at the file download options in Selenide.

How can I download some file in my test?

At some point of our career all of us will face the problem of downloading the file in the test.

As we remember, in Selenium it wasn't that easy, because different browsers required different configurations like creating new user profile in Firefox and setting preferences:

```
profile.setPreference("browser.download.dir", downloadPath);
profile.setPreference("browser.download.folderList", 2);
profile.setPreference("browser.download.manager.showWhenStarting", false);
profile.setPreference("browser.helperApps.alwaysAsk.force", false);
profile.setPreference("browser.helperApps.neverAsk.saveToDisk", mimeTypes);
profile.setPreference("browser.download.manager.focusWhenStarting",false);
profile.setPreference("browser.download.manager.useWindow", false);
profile.setPreference("browser.download.manager.showAlertOnComplete", false);
profile.setPreference("pdfjs.disabled", true);
```

In Selenide,

this problem has been solved by adding `$.download()` method.

When we want to download some file, we just have to perform:

```
File report = element.download();
```

and Selenide will handle with all downloads window popup and just close them when process is finished.

Selenide will create new folder containing our file in `build/reports/tests` . This is the folder where Gradle generate its rest reports, so that it's convenient to have them all together.

We can edit our download directory by changing default settings:

```
Configuration.downloadsFolder = <desired location for downloaded files>;
```

BUT:

This way we will be able to download some file only when its element has "href" attribute.

What if my element does not have "href" attribute? For example, if file is generated as a result of form submission.

In this case case we have to organize downloading of our files in a bit different way. First of all we have to change settings:

```
Configuration.proxyEnabled = true;
Configuration.fileDownload = PROXY;
```

After changing these settings, we are able to download files from all kinds of elements, we don't need a "href" attribute anymore and we just have to use:

```
File report = element.download();
```

Tip:

Remember to set the appropriate timeout if you want to download files that have a big size - it will take more time to download them.

File will be downloaded to the default folder (something like `C:\downloads and settings\downloads`). We have to remember that in case of some browsers, files are also downloaded to system downloads folder so our file will be present in two locations.

In the next steps we can for example delete folder with downloaded file using:

```
FileUtils.deleteDirectory(new File(<folder we want to delete>));
```

For more detailed info about file downloading mechanism read this post.

Maciej Grymuza (figrym@gmail.com)

The project is maintained by

well-crafted software