Joel Min

Monday, 4 April 2016

How to login to a website using Jsoup (Java)

How to login to a website using Jsoup (Java)

To be able to successfully login to a website using Jsoup, we need to have the following prepared:

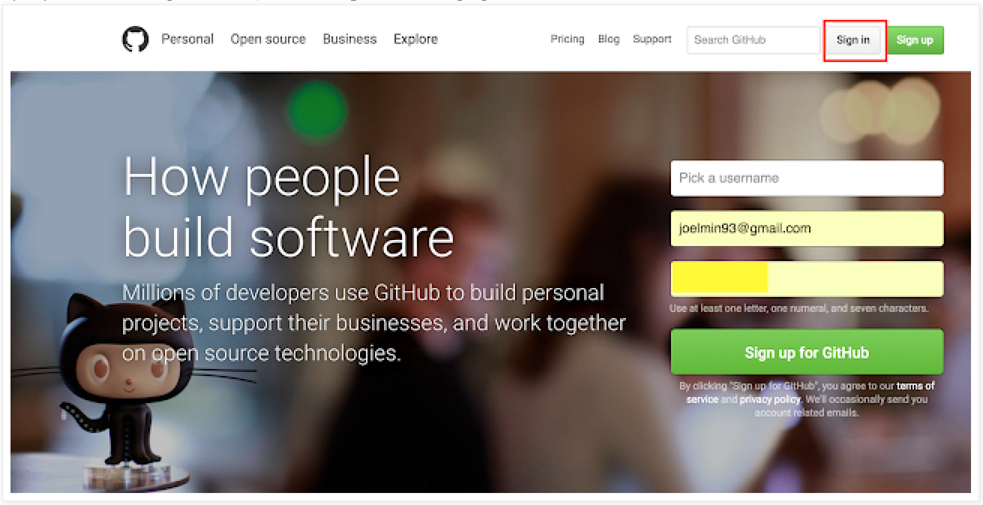
1. A browser with **developer tool** enabled (**Google Chrome** is recommended)
2. The form data that is sent with the login request
3. The cookies required

The key of logging into a website with Jsoup is to simulate the browser, in other words, it can be as simple as **copying** the browser's header and form data, then it is just a matter of POSTing them using Jsoup instead of the browser.

For this tutorial we will be logging into **Github**, a popular source code repository website.

Step 1: Inspect the login request

1. Open your browser and go to **Github**, then click **Sign In** button highlighted in the below screenshot.



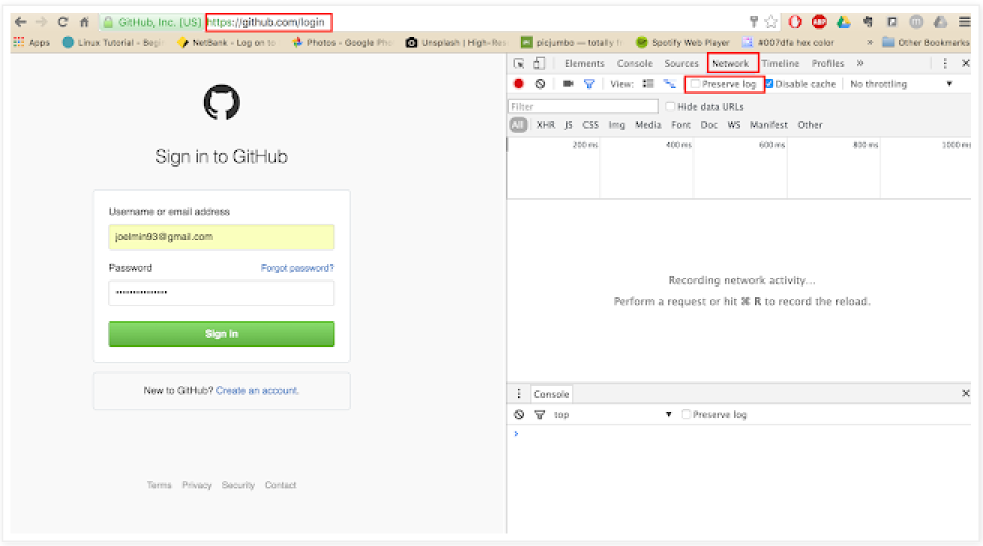
How people build software

Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.

Use at least one letter, one numeral, and seven characters.

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We'll occasionally send you account related emails.

2. You are now in the login page, keep note of the login page url displayed in the browser because we need it later. Now **right-click** anywhere on the page and click **Inspect** (Or just open the developer tool if you know how). You will now have the developer tool open, click on **Network** tab then make sure **Preserve log** is checked.



Recording network activity...

Perform a request or hit ⌘ R to record the reload.

About Me

Unknown

View my complete profile

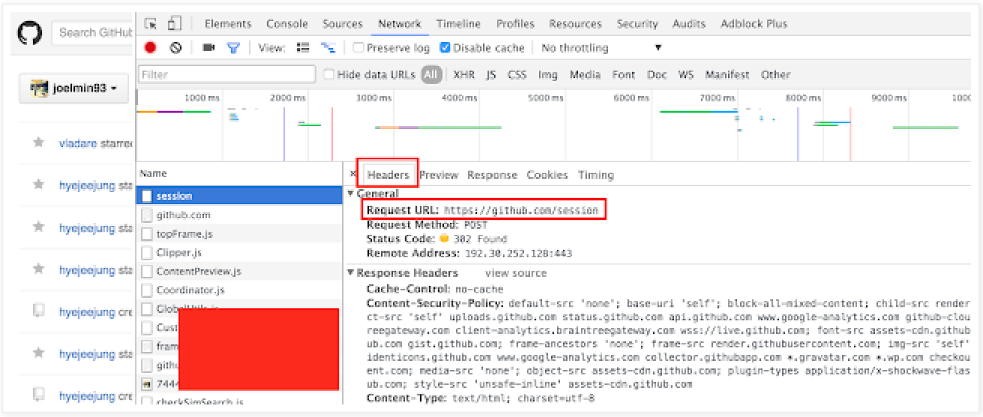
Blog Archive

▼ 2016 (2)
▼ April (2)
How to login to a website using Jsoup (Java)
How to disable ALL buttons in a layout - Android

3. Now login to the website. As soon as you click login button, you will see a request called **"session"** has been made to the server. **Click** on it to display details (There will be many other requests made, just scroll up to the very top to view the **FIRST** request that was made after login button was clicked).

On the right panel you will be presented with details of the request, click the **Headers** tab.

Make note of the **Request URL** address, this is the url we will POST to using Jsoup.



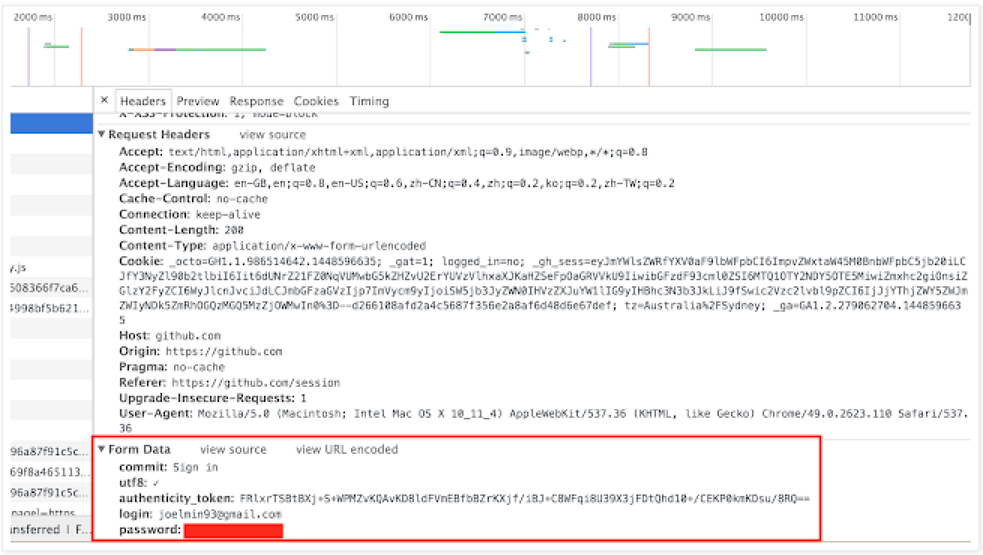
[redacted]

```
Headers  Preview  Response  Cookies  Timing
▼ General
  Request URL: https://github.com/session
  Request Method: POST
  Status Code: 302 Found
  Remote Address: 192.30.252.128:443
▼ Response Headers     view source
  Cache-Control: no-cache
  Content-Security-Policy: default-src 'none'; base-uri 'self'; block-all-mixed-content; child-src render
  ct-src 'self' uploads.github.com status.github.com api.github.com www.google-analytics.com github-clou
  reegateway.com client-analytics.braintreegateway.com wss://live.github.com; font-src assets-cdn.github
  ub.com gist.github.com; frame-ancestors 'none'; frame-src render.githubusercontent.com; img-src 'self'
  identicons.github.com www.google-analytics.com collector.githubapp.com *.gravatar.com *.wp.com checkou
  ent.com; media-src 'none'; object-src assets-cdn.github.com; plugin-types application/x-shockwave-flas
  ub.com; style-src 'unsafe-inline' assets-cdn.github.com
  Content-Type: text/html; charset=utf-8
```

4. Scroll down to **Form Data** section. There you will see pairs of data that has been sent by the browser to login. We will be sending the exact same data pairs using Jsoup, except for the **authenticity_token** value.

The **authenticity_token** is a *unique identifier* that has been sent with the login form by the server. This value is valid **only** for the login page we just visited. So we need a way to **get this value from the login page we just visited**.

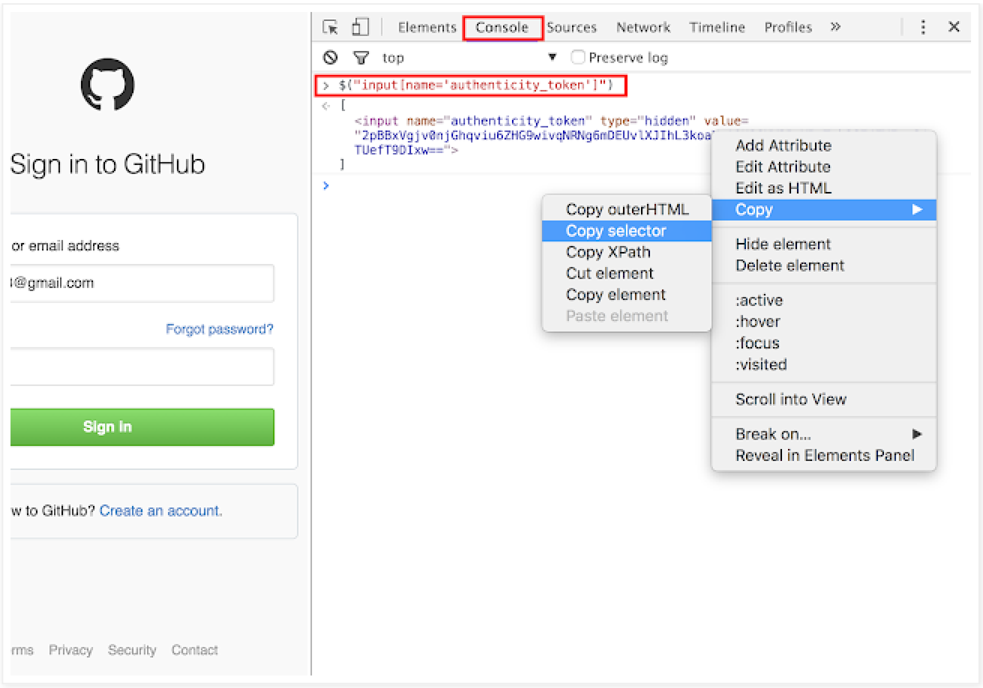
Fortunately this is easy because the value can be extracted using Jsoup from the login page. Please follow next paragraph.



```
Headers  Preview  Response  Cookies  Timing
▼ Request Headers     view source
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
  Accept-Encoding: gzip, deflate
  Accept-Language: en-GB,en;q=0.8,en-US;q=0.6,zh-CN;q=0.4,zh;q=0.2,ko;q=0.2,zh-TW;q=0.2
  Cache-Control: no-cache
  Connection: keep-alive
  Content-Length: 208
  Content-Type: application/x-www-form-urlencoded
  Cookie: _octo=GH1.1.986514642.1448596635; _gat=1; logged_in=no; _gh_sess=eyJmYWlsZWRfYXV0aF9lbWFpbCI6ImpvZWxtaW45M0BnbWFpbC5jb20iLC
  JfY3NyZl90b2tlbiI6Iit6dUNrZ21FZ0NqVUMwbG5kZHZvU2ErYUVzVlhxaXJKaHZSeFp0aGRVVkU9IiwibGFzdF93cml0ZSI6MTQ1OTY2NDY5OTE5MiwiZmxhc2giOnsiZ
  GlzY2FyZCI6WyJlcnJvciJdLCJmbGFzaGVzIjp7ImVycm9yIjoiSW5jb3JyZWN0IHVzZXJuYW1lIG9yIHBhc3N3b3JkLiJ9fSwic2Vzc2lvbl9pZCI6IjJjYThjZWY5ZWJm
  ZWIyNDk5ZmRhOGQzMGQ5MzZjOWMwIn0%3D--d266188afd2a4c5687f356e2a8af6d48d6e67def; tz=Australia%2FSydney; _ga=GA1.2.279862704.144859663
  5
  Host: github.com
  Origin: https://github.com
  Pragma: no-cache
  Referer: https://github.com/session
  Upgrade-Insecure-Requests: 1
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/49.0.2623.110 Safari/537.
  36
▼ Form Data     view source     view URL encoded
  commit: Sign in
  utf8: ✓
  authenticity_token: FRlxrTSBtBXj+S+WPMZvKQAvKD8ldFVnEBfbBZrKXjf/iBJ+C8WFqi8U39X3jFDtQhd10+/CEKP0kmKDsu/8RQ==
  login: joelmin93@gmail.com
  password: [redacted]
```

5. Logout of the website, then go back to the login page. Then open the developer tool and go to **Console** tab. Then enter the following: **$("input[name='authenticity_token']")**. This is a JQuery script that will retrieve the element with tag **input** and attribute name **authenticity_token**, which is what we are after.

This script will return the element that contains authenticity token, right click on the element and copy the **selector**, then paste it somewhere to save it because we will use it to get the authenticity value with Jsoup later.



```
> $("input[name='authenticity_token']")
< [
    <input name="authenticity_token" type="hidden" value=
    "2pBBxVgjv0njGhqviu6ZHG9wivqNRNg6mDEUvlXJIhL3koa
    TUefT9DIxw==">
  ]
>
```

Step 2:

We are almost there, we now have everything we need below:

Login form url: https://github.com/login
Login Action url: https://github.com/session

Login Form data:

- commit -> Sign in
- utf8 -> e2 9c 93 (this is the small check mark)
- authenticity_token -> we will get this value later
- login -> your email
- password -> your password

Make sure you have the above data ready, then proceed to next step!

Step3:

1. Now it's time for some code. First declare the variables we already know:

```
final String USER_AGENT = "\"Mozilla/5.0 (Windows NT\" +\n" +
         "          \" 6.1; WOW64) AppleWebKit/535.2 (KHTML, like Gecko) Chrome/15.0.874.1
 String loginFormUrl = "https://github.com/login";
 String loginActionUrl = "https://github.com/session";
 String username = "joelmin93@gmail.com";
 String password = "XXXX";
```

The **USER_AGENT** is a string that tells the server we are a normal browser (rather than a sneaky little web crawler) so that the server accepts our connection.

2. Declare **HashMap**s that will be used to store the cookies and form data:

```
HashMap<String, String> cookies = new HashMap<>();
HashMap<String, String> formData = new HashMap<>();
```

3. Connect to login form url and retrieve the response:

```
Connection.Response loginForm = Jsoup.connect(loginFormUrl).method(Connection.Method.GET)
Document loginDoc = loginForm.parse(); // this is the document that contains response htm
```

4. Save the cookies from the response, and retrieve the **authenticity_token**

```
cookies.putAll(loginForm.cookies()); // save the cookies, this will be passed on to next
/**
```

```
* Get the value of authenticity_token with the CSS selector we saved before
**/
String authToken = loginDoc.select("#login > form > div:nth-child(1) > input[type=\"hidde
     .first()
     .attr("value");
```

5. We now have everything we need, construct the form data to send to server:

```
formData.put("commit", "Sign in");
formData.put("utf8", "e2 9c 93");
formData.put("login", username);
formData.put("password", password);
formData.put("authenticity_token", authToken);
```

6. Last step, fire the request!

```
Connection.Response homePage = Jsoup.connect(loginActionUrl)
        .cookies(cookies)
        .data(formData)
        .method(Connection.Method.POST)
        .userAgent(USER_AGENT)
        .execute();
```

Step 4: DONE
That's it! Now **homePage** object will contains the response from the successful login, to check the html content, simply parse the html from the response and print it out to see that we are actually logged in :)

```
System.out.println(homePage.parse().html());
```

Conclusion
These steps may seem like a lot but they are really not once you know what you are doing. Using this technique you will most likely be able to login to any websites and retrieve the information you need :)

Posted by Unknown at 02:47

11 comments:

Huge G 8 November 2016 at 13:56
This comment has been removed by the author.
Reply

Huge G 8 November 2016 at 16:53
This comment has been removed by the author.
Reply

Huge G 8 November 2016 at 17:36
You cant use this on fb as they autogenerate auth key with javascript..
Reply

Unknown 21 April 2017 at 13:28
I dont have an authenticity_token, but a secret key. And this key doesnt have a value. What to do?
Reply

Unknown 20 June 2018 at 03:44

thanks for the blog
i have been try this on twitter but unfortunately it doesn't work, so if any one have an idea about this i would be thankful.
thanks in advance.

Reply

Unknown 31 October 2018 at 09:24

do all websites have auth token. for more primitive sites will this still be necessary.
is this step site specific or more general ?

String authToken = loginDoc.select("#login > form > div:nth-child(1) > input[type=\"hidden\"]:nth-child(2)")
.first()
.attr("value");

Reply

shizzledizzleeee 16 January 2019 at 11:29

I think this is a tick (formData.put("utf8", "e2 9c 93");)

Reply

Tnxah 1 April 2020 at 16:08

How can i use this connection for different pages on this website?
I need to get some information from website which i can get only if i am logged in

Reply

Cường 10 May 2020 at 09:52

good

Reply

Persian 11 June 2020 at 01:17

What if the login page has I'm not a robot one click capcha to login in

Reply

dhoni ramakrishna 1 April 2021 at 10:18

is it possible to login secure websites in jsoup

Reply

Enter your comment...

Comment as: roybaileybiz@g Sign out

Publish Preview Notify me

Home Older Post

Subscribe to: Post Comments (Atom)

Powered by Blogger.