Vladimir Rahm

Follow About

How to call REST API with CSRF login and security tokens (Node.js)

Vladimir Rahm Nov 24, 2020 · 6 min read

This article is about an API call, which is hidden behind the CSRF login form. Such API is considered to be quite non-standard, but reading this article should be beneficial and maybe some parts will help you with your project.

Who should be interested:

- An IT guy who is interested in API calls
- Call of Duty player who wants to create a tailor-made solution for his (and his friends) statistics
- Someone who is interested in CSRF security

This article is written in the form of a step-by-step procedure on a specific example of an API provided by the company Activision. The API provides an overview of player statistics from Call of duty online games. Important remark, the author of this article is not a professional developer therefore do not seek perfect code notation. This article also does not provide any detailed explanation of security tokens.

Disclaimer: This article is for study purposes and does not offer anyone for any criminal activity.

Requirements:

- Basic knowledge of programming
- Installed Node.js
- Activision account (it shouldn't matter whether you play Call of duty or not because you have access to other players statistics)

The API that is accessed is documented at https://documenter.getpostman.com/view/7896975/SW7aXSo5. Many thanks to the author of the documentation Andries "iShot" Verbanck. If you are a Call of Duty player, please check out his Discord bot https://modernwarfarediscordbot.com/, which uses this API, and try out some of its features.

Steps to successfully call the API with player statistics are:

1. Get CSRF token from the login page (GET)
2. Log in using CSRF token and Activison credentials (POST)
3. Use Cookies received from successful login (GET)

Let's get started with the first step. Login page of Call of duty is available here https://profile.callofduty.com/ (you can see HTTPS, so use port 443). Create an

```
const http = require('https');

var options1 = {
 host: 'profile.callofduty.com',
 port: 443,
 path: '/cod/login',
 method: 'GET'
};

http.request(options1, function(res) {
 var body = '';

 res.on('data', function(chunk){
  body+= chunk;
 });

 res.on('end', function(){
  console.log(body);
 });

}).end();
```

Now you can retrieve the CSRF token from the body. Feel free to use some prepared library for HTML parsing (for example jsdom is popular and easy https://www.npmjs.com/package/jsdom) or create your own parsing mechanism. You can find one of the possible approaches without an additional library below:

```
const http = require('https');

var options1 = {
 host: 'profile.callofduty.com',
 port: 443,
 path: '/cod/login',
 method: 'GET'
};

http.request(options1, function(res) {
 var body = '';
 res.on('data', function(chunk){
  body+= chunk;
 });

res.on('end', function(){
  let bodyLength = body.length;
  let csrfToken = '';
y[i+11]+body[i+12]+body[i+13];

   if(somePart=='_csrf" value="'){
    for (y=i+13; y<bodyLength; y++){
     anotherPart=body[y];
     if(anotherPart=='>'){
      break;
     }
    }
    for (z=i+14; z<y-3; z++){
     csrfToken=csrfToken+body[z];
    }
    console.log(csrfToken);
    break;
   }
  }
 });
}).end();
```

Congratulations! If you followed the manual then you should have the CSRF token in a separate variable. Now let's login using the CSRF token. The login page is available here https://profile.callofduty.com/cod/login, however, if you check the Network activity then you'll see that the authentication (login magic) is happening on this endpoint "/do_login?new_SiteId=cod". You must also enhance your CSRF token with some additional data and besides the CSRF token, it's also required to send the parameter "new_SiteId=cod". First, enhance your token and add the additional parameter:

```
cookieToken=String('XSRF-TOKEN='+myToken+'; new_SiteId=cod');
```

Now the variable is prepared to be used as a Cookie parameter in the POST call. You need to also build form input data besides the Cookie parameter. Use the CSRF token in the parameter "_csrf", set the parameter "remember_me" as true, and use your Activision credentials in the parameters "username" and "password":

```
var querystring = require('querystring');

const postData = querystring.stringify({
 'username': %Activison_Username%,
 'password': %Activison_Password%,
 'remember_me': true,
var options2 = {
 host: 'profile.callofduty.com',
 port: 443,
 path: '/do_login?new_SiteId=cod',
 method: 'POST',
 headers: {
  'Content-Type': 'application/x-www-form-urlencoded',
  'Cookie': cookieToken
 },
};
```

As you can see, the code in the previous code block is not complete. This is so that the article is not stretched unnecessarily. But don't worry, the whole functional piece of code is available at the end of the article. You can either try the previous and next challenges yourself, or you can get a fully working piece of code at the end of the article.

After successful login, take the entire cookies and store them in a separate variable that will be used for the final GET call with player statistics. Feel free to use the following player's nickname for the API call if you don't have a better candidate: "Freezy#2376". The character "#" needs to be escaped in the URL, therefore the final URL has the following form: https://my.callofduty.com/api/papi-client/stats/cod/v1/title/mw/platform/battle/gamer/Freezy%232376/profile/type/mp. You can try to open this URL from your web browser and see the API output. But you must be logged into the Activision account (https://profile.callofduty.com/cod/login). If you are not logged in then you will receive the following output:

```
{"status":"error","data":
{"type":"com.activision.mt.common.stdtools.exceptions.NoStackTrac
eException","message":"Not permitted: not authenticated"}}
```

Now back to the code. Take the cookies from the previous POST login and use them in the final API statistics GET call:

```
var options3 = {
 host: 'my.callofduty.com',
 port: 443,
 method: 'GET',
 headers: {
  'Cookie': %Cookies from POST login%
 }
};
```

That's it! If you successfully called the API with the steps for CSRF token and authentication, then you should see player's statistics in the following form:

```
var playerStats= JSON.parse(body);
console.log(playerStats);

Output:
{
  status: 'success',
  data: {
    title: 'mw',
    platform: 'battle',
    username: 'Freezy#2376',
    type: 'mp',
    level: 132,
    maxLevel: 0,
    levelXpRemainder: 3000,
    levelXpGained: 7000,
    prestige: 0,
    prestigeId: 0,
    maxPrestige: 0,
    totalXp: 960000,
    paragonRank: 0,
    paragonId: 0,
    s: 0,
    p: 0,
    lifetime: {
      all: [Object],
      mode: [Object],
      map: {},
      itemData: [Object],
      scorestreakData: [Object],
      accoladeData: [Object]
    },
    weekly: { all: [Object], mode: {}, map: {} },
    engagement: null
  }
}
```

A few words at the end. This article provides one possible solution for logging on using a CSRF token and calling an API that requires form authentication. It is uncertain how

PS: Entire code block with error handling fully functional on 24.11.2020 (don't forget to fill your Activision Username and Password):

```
// Setp 1) Obtain CSRF token from the login page
// Step 2) Login using CSRF token, username and password
// Step 3) Use Cookies received from successful login and reuse
them in API statistics calls

const http = require("https");

var querystring = require("querystring");

var options1 = {
 host: "profile.callofduty.com",
 port: 443,
 path: "/cod/login",
 method: "GET",
};

http.request(options1, function (res) {
  var body = "";
  res.on("data", function (chunk) {
   body += chunk;
  });

  res.on("end", function () {
   let bodyLength = body.length;
   let myToken = "";
   let somePart="";
   let anotherPart="";

   for (i = 0; i < bodyLength; i++) {
    somePart = body[i] + body[i+1] + body[i+2] + body[i+3] +
body[i+4] + body[i+5] + body[i+6] + body[i+7] + body[i+8] +
body[i+9] + body[i+10] + body[i+11] + body[i+12] + body[i+13];

    if (somePart == '_csrf" value="') {
     for (y = i + 13; y < bodyLength; y++) {
      anotherPart = body[y];
      if (anotherPart == ">") {
       break;
      }
     }
     for (z = i + 14; z < y - 3; z++) {
      myToken = myToken + body[z];
     }
     break;
    }
   }
```

```
    const postData = querystring.stringify({
     username: %Activison_Username%,
     password: %Activison_Password%,
     remember_me: true,
     _csrf: myToken,
    });

    var options2 = {
     host: "profile.callofduty.com",
     port: 443,
     path: "/do_login?new_SiteId=cod",
     method: "POST",
     headers: {
      "Content-Type": "application/x-www-form-urlencoded",
      Cookie: cookieToken,
     },
    };

    var req = http.request(options2, function (res2) {
     let myHeader = JSON.stringify(res2.headers);
     let myHeaderLength = myHeader.length;
     let myPostCookiesFull = "";
     for (i = 0; i < myHeaderLength; i++) {
      somePart = myHeader[i];
      if (somePart == "[") {
       for (y = i; y < myHeaderLength; y++) {
        anotherPart = myHeader[y];
        if (anotherPart == "]") {
         console.log("end", y, anotherPart);
         break;
        }
      }
      for (x = i; x < y - 1; x++) {
       myPostCookiesFull = myPostCookiesFull + myHeader[x];
      }
      break;
     }
    }

    res2.setEncoding("utf8");
    res2.on("data", function (chunk) {});

    var options3 = {
     host: "my.callofduty.com",
     port: 443,
     path: "/api/papi-
  client/stats/cod/v1/title/mw/platform/battle/gamer/Freezy%232376/
  profile/type/mp",
     method: "GET",
     headers: {
     Cookie: myPostCookiesFull,
     },
    };
```

```
      res3.on("data", function (chunk) {
       body += chunk;
      });

      res3.on("end", function () {
       var playerStats = JSON.parse(body);
       console.log(playerStats);
      });
     }).end();
    });

     req.on("error", function (e) {
      console.log("problem with request: " + e.message);
     });

     req.write(postData);
     req.end();
     });
    }).end();
```

Rest Api Nodejs Rest Csrf Call Of Duty Warzone