


How to properly use Bearer tokens?

Asked 6 years, 10 months ago Modified 1 year, 9 months ago Viewed 99k times  Part of [PHP Collective](#)



I'm making an authorization system in `PHP`, and I came across this Bearer scheme of passing JWT tokens, I read [RFC 6750][1]. I've got the following doubts:

60



1. How is this improving the security?
2. The server responses the client with a JWT token in its body after a successful authorization and login, and now when the client makes another request, I am not clear how to actually do that, I want to send token from client in Authorization header in the request, so now should I just prefix "Bearer" to the token which I received in the previous response from the server and If yes, then server on receiving the Authorization header, should just split the string with space, and take the second value from the obtained array and then decode it? For example `Authorization: Bearer fdbghfbfjbhg_something`, how is server supposed to handle this, `decodeFunc(explode(" ", $this->getRequest()->getHeader("Authorization"))[1])` ? [1]: <https://www.rfc-editor.org/rfc/rfc6750>

PHP

php

authentication

oauth

http-headers

bearer-token

Share Follow

edited Oct 7, 2021 at 7:34



Community Bot

1 ● 1

asked Nov 14, 2016 at 5:03



Ashish Ranjan

12.8k ● 5 ● 27 ● 51

2 Answers

Sorted by: Highest score (default) 



- 1.Improving the security because if token is not sent in the header that sent in url, it will be logged by the network system, the server log

184



- 2.A good function to get Bearer tokens



```
/**
 * Get header Authorization
 * */
function getAuthorizationHeader(){
    $headers = null;
    if (isset($_SERVER['Authorization'])) {
        $headers = trim($_SERVER["Authorization"]);
    }
    else if (isset($_SERVER['HTTP_AUTHORIZATION'])) { //Nginx or fast CGI
        $headers = trim($_SERVER["HTTP_AUTHORIZATION"]);
    } elseif (function_exists('apache_request_headers')) {
        $requestHeaders = apache_request_headers();
        // Server-side fix for bug in old Android versions (a nice side-effect
        of this fix means we don't care about capitalization for Authorization)
        $requestHeaders = array_combine(array_map('ucwords',
        array_keys($requestHeaders)), array_values($requestHeaders));
    }
```

```

        //print_r($requestHeaders);
        if (isset($requestHeaders['Authorization'])) {
            $headers = trim($requestHeaders['Authorization']);
        }
        return $headers;
    }

    /**
     * get access token from header
     */
    function getBearerToken() {
        $headers = getAuthorizationHeader();
        // HEADER: Get the access token from the header
        if (!empty($headers)) {
            if (preg_match('/Bearer\s(\S+)/', $headers, $matches)) {
                return $matches[1];
            }
        }
        return null;
    }
}

```

Share Follow

edited Dec 31, 2021 at 5:17

answered Nov 14, 2016 at 5:39



STA

31k ● 9 ● 45 ● 59



Ngô Văn Thao

3,701 ● 1 ● 20 ● 24

It's a nice function, you have suggested, but the `$headers` I am returning, It will have, say: `Bearer <space> <AuthToken>` so now, is it right to just explode the string by space and take the actual token, or the full string (`Bearer <space> <token>`) is supposed to be taken as a whole?

– Ashish Ranjan Nov 14, 2016 at 6:46

`<AuthToken>` that's right. You can explode by space. – Ngô Văn Thao Nov 14, 2016 at 7:29

I missed a function. XD – Ngô Văn Thao Nov 15, 2016 at 5:44 ✎

can you please add setBearerToken function?? – Fayyaz Ali May 19, 2017 at 17:39 ✎

4 I'd just use a `substr()` to get the actual token but not the bearer. Should be much faster than `preg_match()`. – InputOutput Feb 27, 2018 at 15:19



I would recommend to use the following RegEx to check, if it's a valid jwt-token:

4

```
/Bearer\s((.*)\.(.*)\.(.*))/
```



and access it also with `matches[1]`.



This is the structure of a JWT-Token, see: <https://jwt.io/>



Share Follow

edited Mar 23, 2017 at 18:32

answered Mar 23, 2017 at 18:15



Unkn0wn0x

1,051 ● 1 ● 12 ● 14

1 Regex is not always the recommend approach for parsing string unless you can't acheive with the existing PHP String functions. – sk8terboi87 ʘ Apr 30, 2017 at 4:56 ✎

-
- 5 Regex is used to find patterns. Imo, you can use regex to parse the jwt token which is in the format 'Bearer <token>'. I believe this is the best approach I can think of, when it comes to token validation. The accepted answer also uses Regex. I don't know why this is down voted even if it has a better regex. – [Arvind](#) Jan 17, 2018 at 6:06
-
- 2 This is not a good approach at all. JWT tokens can be signed with private/public certificates, which regex won't check against. – [sridesmet](#) Apr 16, 2018 at 11:49
-