nginx - njs - ngx_http_auth_request_module how to return 302 to user?

Asked 3 years, 6 months ago Modified 3 years, 4 months ago

Viewed 1k times

0

I am using njs ngx_http_auth_request_module.

I have js function like this;

```
function introspectAccessToken(r) {
    r.subrequest("/_oauth2_send_request",
        function(reply) {
            if (reply.status == 200) {
                r.return(204); // Token is valid,
return success code
            } else {
                // need 302 here
            }
        }
    );
}
```

The documentation says "If the subrequest returns a 2xx response code, the access is allowed. If it returns 401 or 403, the access is denied with the corresponding error code. Any other response code returned by the subrequest is considered an error." http://nginx.org/en/docs/http/ngx_http_auth_request_module.html

I need to return 302 to user if subsequest enters "else" block.

Is there anyway to achieve this? If I set r.return(302), it shows Error page in browser as documentation said.

edit: my nginx.conf

```
location /my-webclient {
    auth_request /_oauth2_token_introspection;
    root   /usr/share/nginx/html;
    rewrite ^ /hmb-profile-webclient/index.html
break;
}

location = /_oauth2_token_introspection {
    internal;
    js_content introspectAccessToken;
}

location /_oauth2_send_request {
    internal;
    proxy_method      POST;
    proxy_set_body    $cookie_jwt;
    proxy_pass http://my-specific-url;
}
```

http://my-specific-url returns

- 200, if jwt cookie is valid
- 302 (with return location), if jwt cookie is invalid

nginx nginx-module nginx-njs njs

Share

edited Aug 16, 2021 at 12:47

Improve this question

Follow

asked May 27, 2021 at 10:52

denizg

934 ● 9 ● 22

still open? Please share the the nginx configuration as well. – Timo Stark Jul 13, 2021 at 14:01

@TimoStark yes still open. i added my related nginx.conf to question – denizg Aug 16, 2021 at 12:47

1 Answer

Sorted by: Highest score (default)

1

Got it - I don't think you even need `njs` to do this. In case your token introspection endpoint is NOT build with NJS and its not validating the if the token is a valid json but it is implemented with some other language this NGINX config will work.

```
#cache for auth_request responses

proxy_cache_path /var/cache/nginx/oauth
keys_zone=token_responses:1m max_size=2m;

#Your Server
server {

  listen 80;


  location / {
    #Redirect Unauthed Request to some other
location...
```

```
    error_page 401 =302
http://localhost:9000/login;

    auth_request /_auth_request;
    proxy_pass http://127.0.0.1:8080;



  }

  location /_auth_request {
    internal;
    proxy_pass http://127.0.0.1:8090;
    ###place caching responses here to be more
effective;
    proxy_cache           token_responses; #
Enable caching
    proxy_cache_key       $cookie_jwt;    # Cache
for each access token
    proxy_cache_lock      on;              #
Duplicate tokens must wait
    proxy_cache_valid     200 10s;         # How
long to use each response
    proxy_ignore_headers  Cache-Control Expires
Set-Cookie;
```

In case you want to build more logic with NJS like validate the token structure your approach was the right one using NJS with js_set.

```
js_include oauth2.js; # Location of JavaScript
code

server {
    listen 80;

    location / {
        error_page 401 403 =302
http://localhost:9000/login;
        auth_request /_oauth2_token_introspection;
        proxy_pass http://my_backend;
    }

    location = /_oauth2_token_introspection {
        internal;
        js_content introspectAccessToken;
    }
```

njs function

```
function introspectAccessToken(r) {
    r.subrequest("/_oauth2_send_request",
        function(reply) {
            if (reply.status == 200) {
                var response =
JSON.parse(reply.responseBody);
                if (response.active == true) {
                    r.return(204); // Token is
valid, return success code
                } else {
                    r.return(403); // Token is
invalid, return forbidden code
                }
            } else {
                r.return(401); // Unexpected
response, return 'auth required'
            }
        }
    );
}
```

The `error_page` directive will work likewise. Based on you code I assume you already checked out this article?

https://www.nginx.com/blog/validating-oauth-2-0-access-tokens-nginx/

ICYMI it's a great blog post about this use case.

Share Improve this answer

Follow

answered Aug 16, 2021 at 23:12

Timo Stark

3,051 ●1 ●14 ●27

thank you. you right, i didnt really need njs. actually, i follow nginx guide you shared, but my use case is little bit different. – denizg Aug 17, 2021 at 8:32