

Token-based authentication sample for Teamwork Cloud REST API

On this page

- Setting up token-based authentication
- Using token-based authentication
- Limitations

Teamwork Cloud REST API has an endpoint, which implements token-based authentication described in the [Token-based authentication](#) page.

Setting up token-based authentication

To set up Teamwork Cloud and Authentication server for token-based authentication using REST API endpoint

1. Set up Authentication server to work with your SAML/SSO, if needed.
2. Open `<install_root>/WebAppPlatform/shared/conf/authserver.properties`.
3. Find key **authentication.client.ids** and add **twc-rest-api** at the end of the value. Save and close the file.



The comma before **twc-rest-api** is a separator.

4. Open `<install_root>/TeamworkCloud/configuration/application.conf`.
5. Find the **esi.auth** block and set the server value to your authentication server IP. Save and close the file.
6. Restart the Authentication server and Teamwork Cloud server.

Using token-based authentication

To use token-based authentication

1. Open REST API at the following URL: <https://<ip>:8111/osmc/authn/login> on a browser.
2. You should be redirected to the Authentication server login page.
3. Enter your credentials. The browser shows you a token, usually starting with `eyJ...`

Login token is

```
gYvhGc0i0JlSuZtIINi9j.eYjzdw0i0JvdW0iLCJhdWQio0J0d2MtcMvzdC1hcGskLJCpc3MiOi0JodHwRwcwpCLlwmVMTKqYlE2OC4yMjgmUmtZOj  
g1NTVC2PLfghdXwvRP2foA9uNwi1WiZXhwijoxNjA4MTF0MTgxLjCpYXQoOjE2MDgxMTYxOTYODF9.H9PHeHWS5DmrHwxRxKTQ=  
lPeOdQjiEH6kCXlYagZvYmHO0fJB.ecgg5B-  
87Lw3T3ti8zxKLrHCa8Ag0Ytm5S3gcgwBcm41.FKqUG6_4bpuSzGu2bVFf3L3M7ZdcOc7vQFhbMcMri8GlEnhgLnJGCxtc_a3tv00m_44i6-9  
J3VMvAvTKDNWjyvpeeAYcl3au7xd8ElXopz3fbfpX-lqgd-  
FWl7o6XqXVjHRJigXvDWuIrNVt80yc2LhPlqNg0iNr9g1gl9pmcGGzp1FvDCGb36eVlJKB52Xaarly7SDvDUlVfPKM2GaAsdsQk  
XNUXSW9cyqv5SLg.  
All attributes: {org.restlet.start.time=1608113281959, Metric-Start-Time-In-Milleseconds=1608113281959,  
org.restlet.https.keySize=0, Metric-Start-Time-In-Milleseconds=0.com.nmagic.esi.rest.osmc.a.h.b=1608113281959,  
esi.dm.query cors_request=false, e.session=session(User=cum, ID=0ae19c80-65f0-4ceb-818a-26a3891d346e),  
com.nmagic.esi.rest.osmc.a.j.login_token=eyJhbGciOiJSUzU2Iiwia3Vzdw0iLCJhdWQio0J0d2MtcMvzdC1hcGskLJCpc3MiOi0JodHwRwcwpCLlwmVMTKqYlE2OC4yMjgmUmtZOjg1NTVC2PLfghdXwvRP2foA9uNwi1WiZXhwijoxNjA4MTF0MTgxLjCpYXQoOjE2MDgxMTYxOTYODF9.H9PHeHWS5DmrHwxRxKTQ=lPeOdQjiEH6kCXlYagZvYmHO0fJB.ecgg5B-87Lw3T3ti8zxKLrHCa8Ag0Ytm5S3gcgwBcm41.FKqUG6_4bpuSzGu2bVFf3L3M7ZdcOc7vQFhbMcMri8GlEnhgLnJGCxtc_a3tv00m_44i6-9J3VMvAvTKDNWjyvpeeAYcl3au7xd8ElXopz3fbfpX-lqgd-FWl7o6XqXVjHRJigXvDWuIrNVt80yc2LhPlqNg0iNr9g1gl9pmcGGzp1FvDCGb36eVlJKB52Xaarly7SDvDUlVfPKM2GaAsdsQkXNUXSW9cyqv5SLg.org.restlet.http.headers={Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8}, [Upgrade-Insecure-Requests: 1], [User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64;  
rv:78.0) Gecko/20100101 Firefox/78.0], [Connection: keep-alive], [Referer: https://192.168.228.133:8555/  
authentication/authenticate?scope=openid&response_type=code&redirect_uri=https:  
%2F%2F192.168.228.133:8111%2Fosmc%2Fauthen%2Fauthenticate&client_id=twc-rest-api], [Host:  
192.168.228.133:8111], [Accept-Language: en-US,en;q=0.5], [Accept-Encoding: gzip, deflate, br], [DNT: 1]],  
org.restlet.https.sslSessionID=297CE4C17b67f953966a05E26800ce29Eale07DF090E74c324be5937f541bd42}
```

4. Copy the token and use it to log on to REST API.



The token is used in an Authorization header with the **Token Type**.

For example (using a token with cURL):

```
curl -v -k -H "Authorization: Token
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJBZG1pbmlzdHJhdG9yIiwiaXVkiOiJoidHdjLXJlc3QtYXBp
IiwiaXNzIjoiaHR0cHM6XC9cLzEyNy4wLjAuMTQ0NTU1XC9hdXRoZW50aWNhdGlvbiIsImV4cCI6M
TYwNjI5OTc3NywiaWF0IjoxNjA2Mjk4ODc3fQ.bA-
S5hHeSlV8AFoQVzzfIseC3qlmqQoBQREiapHN6I5CcvwetKdSVztWKkssSGjm3lYlZqoULio7_1Ma
mtGBbbzvAlWWQYFRiYk0D6l2yNDv4uNHBBNLNEv6lTYNLwdPwPh0atVRehkh-
LSgjjipXTvXj4mZViE0NHKIG9U7htA9Zzvzvck2JDxe_eU2-
4TCNm8II89ROaEb1tZ5nD84ieRbzJWqrcVTdqU2YfbIUeew5Nir8obkLYgixBXFKWstHi3jNuoBx3
KcAIyZqL6cjtscER4wbk4PEEDC57UVsOcsXWr6yvXIovdJMOiDHo_fJMKgOjDqSyIL-2B2lO-Y-GA"
https://127.0.0.1:8111/osmc/login
```

The result is as follows:

```
> GET /osmc/login HTTP/1.1
> Host: 127.0.0.1:8111
> User-Agent: curl/7.55.1
> Accept: */*
> Authorization: Token
eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJBZG1pbmlzdHJhdG9yIiwiaXVkiOiJoidHdjLXJlc3QtYXBp
IiwiaXNzIjoiaHR0cHM6XC9cLzEyNy4wLjAuMTQ0NTU1XC9hdXRoZW50aWNhdGlvbiIsImV4cCI6M
TYwNjI5OTc3NywiaWF0IjoxNjA2Mjk4ODc3fQ.bA-
S5hHeSlV8AFoQVzzfIseC3qlmqQoBQREiapHN6I5CcvwetKdSVztWKkssSGjm3lYlZqoULio7_1Ma
mtGBbbzvAlWWQYFRiYk0D6l2yNDv4uNHBBNLNEv6lTYNLwdPwPh0atVRehkh-
LSgjjipXTvXj4mZViE0NHKIG9U7htA9Zzvzvck2JDxe_eU2-
4TCNm8II89ROaEb1tZ5nD84ieRbzJWqrcVTdqU2YfbIUeew5Nir8obkLYgixBXFKWstHi3jNuoBx3
KcAIyZqL6cjtscER4wbk4PEEDC57UVsOcsXWr6yvXIovdJMOiDHo_fJMKgOjDqSyIL-2B2lO-Y-GA
>
< HTTP/1.1 204 No Content
< Content-Length: 0
< Content-Type: application/octet-stream
< Date: Wed, 25 Nov 2020 10:08:44 GMT
< Accept-Ranges: bytes
< Server: Restlet-Framework/2.2.3
< Vary: Accept-Charset, Accept-Encoding, Accept-Language, Accept
< Set-Cookie: twc-rest-current-user=Administrator; Path=/osmc; Expires=Wed, 25 Nov 2020 10:23:44 GMT
< Set-Cookie: twc-rest-session-id=f40ef933-5461-4058-ale7-9b8d4021aa8a; Path=/osmc; Expires=Wed, 25 Nov 2020
10:23:44 GMT
<
* Connection #0 to host 127.0.0.1 left intact
```

Limitations

This REST API endpoint only displays the ID token. Usually, ID token is not very long-living. You can configure the ID token expiration in *authserver.properties* file (using property **authentication.token.expiry**).

Usually the ID token needs to be refreshed as described in the page [Token-based authentication](#). However, this REST API does not display a refresh token, which is needed to refresh the ID token.

As a workaround, long-living ID tokens can be generated by adding **,twc-rest-api** to the *authserver.properties* file property **authentication.client.unlimited**. In such case, ID token expiration will be calculated using property **authentication.unlimited.token.expiry**.



Use this feature with caution and make sure that such long-living ID token is adequately protected.