

BITE API AUTHENTICATION

2023-06-22

BITE API supports two different authentication types:

- BASIC
- OAUTH2

It is recommended to use OAUTH2. This method does not share password data but instead that uses authorization tokens to prove an identity between consumer and service provider. OAUTH2 requests are processed faster than using the BASIC method as well.

1. Authentication by BASIC method

In basic HTTP authentication, request contains a header field in the form of `Authorization: Basic <credentials>`, where credentials is the Base64 encoding of ID and password joined by a single colon `:`.

If there is a user: **username** and password: **password**.

1. Make line like this: **username:password**
2. Code that line with base64: `dXNlcm5hbWU6cGFzc3dvcmQ=`
3. Add to the encoded line the word „Basic“ (there should be a space symbol between those two words): `Basic dXNlcm5hbWU6cGFzc3dvcmQ=`
4. Add created line to the authorization header in every request:

```
curl -X POST \
  https://biteapi.bite.lt/XXXXXX \
  -H 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=' \
  -H 'Content-Type: application/json'
-d '{"xxxxxx": "xxxxxx"}
```

If successful, you will access the service API, otherwise „401 Unauthorized“ will be returned.

2. Authentication by OAUTH2 method

For each request, we must send a tag (TOKEN) that can be obtained after authentication through the OAUTH2 server. Example.:

If there is a user: **username** and password: **password**.

1. Perform all the steps as in BASIC authentication to get the encrypted string.
2. Perform a request to the OAUTH2 server by authenticating with the BASIC method and passing the parameter *grant_type*, with the value *client_credentials*. See example below:

```
curl -X POST \
  https://biteapi.bite.lt/login \
  -H 'Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ=' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -d 'grant_type=client_credentials'
```

3. In response there is an `access_token` tag:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6ImNlcm5hbWU6cGFzc3dvcmQ="
  "token_type": "Bearer",
  "expires_in": 43199
}
```

4. It is possible now to send requests to BITE API services using access_token tag. Example:

```
curl -X POST \  
  https://biteapi.bite.lt/XXXXXX \  
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJyZWZkIiwid3JpdGUiLCJ0cnVzdCJdLCJleHAiOiJlNTUzNTY2NjMsImFldGhvcml0aWVzIjpbIlJPTEVfTENBUkQiXSswianRpIjoibWVjN2Y4NDgtNzUxNC00Y2RlLWI2MGQ tNjQ4NjAlNWZiZGJiIiwiaWF0IjoiY2xpZW50X2lkIjoibXktY2xpZW50In0.XYU010_1KhMNvW15XQuFb3hCoEoW3lAtAACvNq0 jp7JRRSArsGdkIBTHDxz5VzdSTdksi-NWJuhJHSIJvXgm4RztWHma5g9poq8JRD91MjrRgpLft9dhKN-TU8tfDmAKXP8bwF5sZRC6uCplkBc5OIARgekLcbpAp0Hgshv7jKiHwQ9WPISZ-1h8SXIqyP0JNfYG-KO6K-xsAIwcNE3nFtEFJ9bvYnfbbnIOqBPWeM7e51zo_pr_Z3C29JC11cSHe862FTZ1-ewqCTqbLpQWDWpzacd1VBT0Dkm6T6D4-N5ouDhk9xYpniI3AmhVCH6qYKgH79CqPexrXFR8EpKLQaQ' \  
  -H 'Content-Type: application/json' \  
  -d '{"XXXXXX": "XXXXXX"}'
```

5. The access_token tag can be used many times before it becomes invalid. When this happens, the tag needs to be updated again by performing all the actions again.