

OAuth 2.0 mit Postman

Um auf der REST API via OAuth 2.0 einen Request in einem Schritt ausführen zu können, wird ein "Pre-request Script" gebraucht, welcher den Bearer Token vor dem gewünschten Request löst.

Für die Authentisierung müssen im Script 3 Variablen mitgegeben werden:

- AUTH_URL
- CLIENT_ID
- CLIENT_SECRET

OAuth 2.0 Pre-request Script

```
const echoPostRequest = {
  url: 'AUTH_URL',
  method: 'POST',
  header: 'Content-Type:application/json',
  body: {
    mode: 'application/json',
    raw: JSON.stringify(
      {
        client_id: 'CLIENT_ID',
        client_secret: 'CLIENT_SECRET',
        grant_type: 'client_credentials'
      }
    )
  }
};

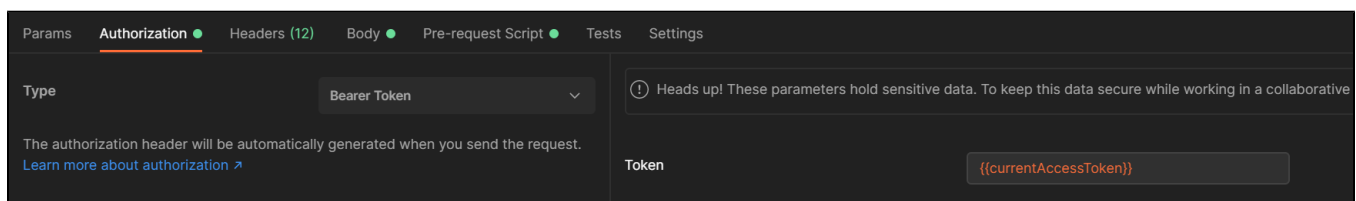
var getToken = true;

if (!pm.environment.get('accessTokenExpiry') ||
    !pm.environment.get('currentAccessToken')) {
  console.log('Token or expiry date are missing')
} else if (pm.environment.get('accessTokenExpiry') <= (new Date()).getTime()) {
  console.log('Token is expired')
} else {
  getToken = false;
  console.log('Token and expiry date are all good');
}

if (getToken === true) {
  pm.sendRequest(echoPostRequest, function (err, res) {
    console.log(err ? err : res.json());
    if (err === null) {
      console.log('Saving the token and expiry date')
      var responseJson = res.json();
      pm.environment.set('currentAccessToken', responseJson.access_token)

      var expiryDate = new Date();
      expiryDate.setMilliseconds(expiryDate.getMilliseconds + responseJson.expires_in);
      pm.environment.set('accessTokenExpiry', expiryDate.getTime());
    }
  });
}
```

Der mit dem Script gelöste Bearer Token wird in der Postman Variable `{{currentAccessToken}}` abgelegt. Diese Variable muss unter 'Authorization' mit 'Type = Bearer Token' hinterlegt werden.



Anwendungsbeispiel:

