



BUNDESDRUCKEREI

Informationen zur Nutzung der ID-Issuer API

Zur Nutzung der API der ID-Issuer Test-Instanz müssen Sie sich nach Open-ID Connect (OIDC) authentisieren. Des Weiteren benötigen Sie ein TLS-Client-Zertifikat. Das Zertifikat erhalten Sie nach erfolgter Registrierung via E-Mail, sowie das dazugehörige Passwort via SMS-Nachricht.

Zum Abholen des OIDC-Access-Tokens muss folgende URL angesprochen werden:

Public-Testsystem:

<https://keycloak.test.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token>

Produktivsystem:

<https://iam.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token>

Der POST-Request zur Anfrage eines Access-Token muss folgende Parameter enthalten:

Parameter	Wert
clientId	Fester Wert "self-management-gui"
clientUsername	Erhält man über des Selfmanagementportal
clientPassword	Erhält man über des Selfmanagementportal
clientSecret	Erhält man über des Selfmanagementportal
grantType	Fester Wert "password"

Beispiel zum Abholen des Access-Token mit CURL:

Initial:

```
curl -d "grant_type=password&username=[Erhält man im SP]&password=[Erhält man im SP]&client_id=self-management-gui&client_secret=[Erhält man im SP]" -H "Content-Type: application/x-www-form-urlencoded" -X POST "https://keycloak.test.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token"
```

Mit Werten belegt:

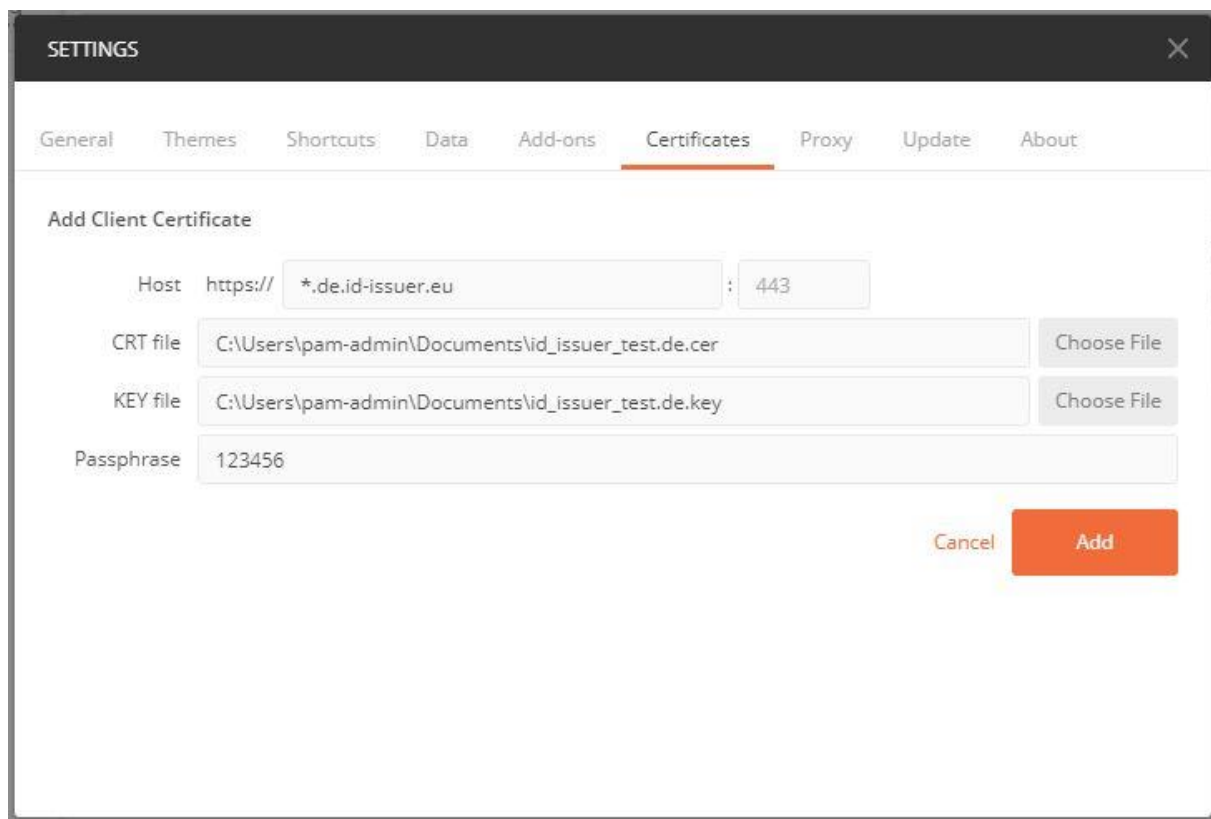
```
curl -d "grant_type=password&username=Uhhkzzh6&password=Jgfou75!&client_id=self-management-gui&client_secret=3d308e29-13d6-46eb-be3b-567ac5275277" -H "Content-Type: application/x-www-form-urlencoded" -X POST "https://keycloak.test.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token"
```

Nutzung der API mit Postman

Für die Nutzung der API mit Postman sind zwei Dinge notwendig:
Zum einen muss das TLS-Client-Zertifikat eingebunden werden und zum anderen muss der Access-Token abgeholt werden. Des Weiteren kann man die YAML-Datei direkt als Collection in Postman importieren.

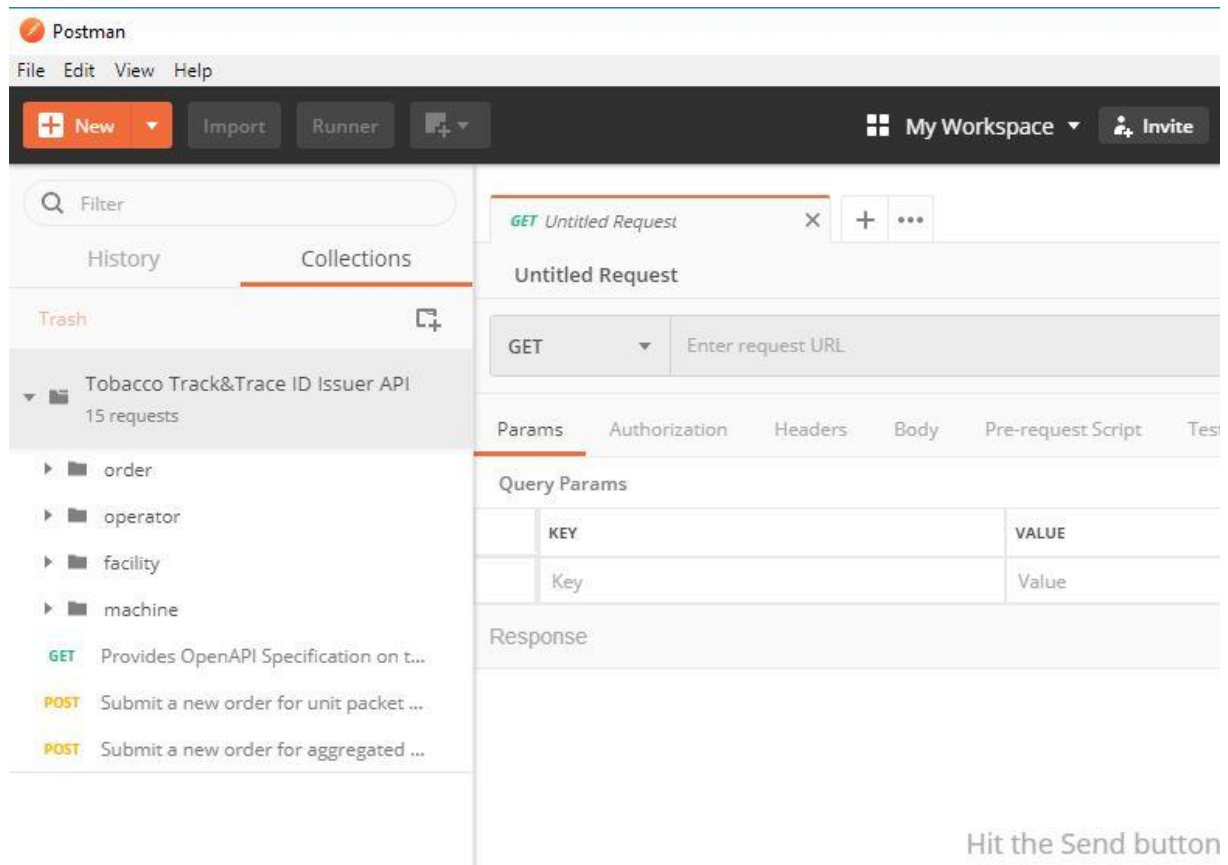
1. TLS-Client-Zertifikat in Postman einbinden.

Das Tool Postman kann mit P12-Container nicht umgehen. Daher muss der private Schlüssel als PKCS8 und die Zertifikatskette als PEM-Format (base64) aus dem P12-Container exportiert werden. Anschließend kann man unter den Einstellungen von Postman das TLS-Client-Zertifikat einbinden, wie in der nachfolgenden Darstellung abgebildet.



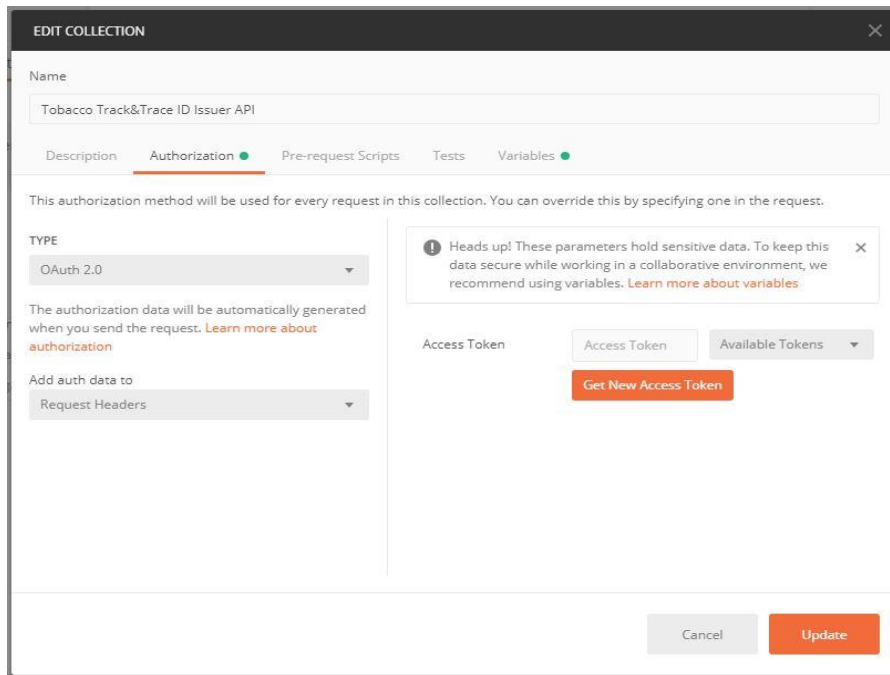
2. YAML-Datei importieren

Die YAML-Datei kann importiert werden und anschließend erhält man eine Collection, welche alle Requests für die ID-Issuer-Instanz aufführt.

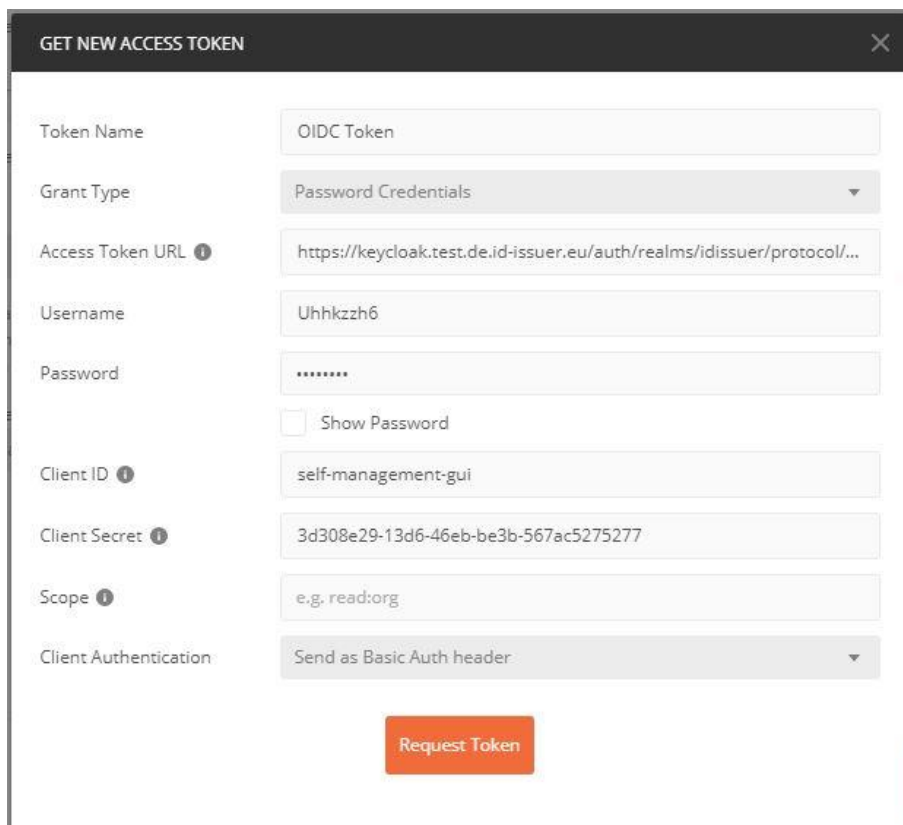


3. Access-Token für die Collection bereitstellen

Um die Requests nutzen zu können, muss noch die Autorisierung für die Collection eingestellt werden. Hierfür klickt man mit der rechten Maustaste auf die Collection und wählt im Kontextmenü „edit“ aus. Anschließend klickt man auf den Reiter Authorization und stellt den Autorisierungstyp „OAuth2.0“ ein.



Auf die Schaltfläche „Get New Access Token“ klicken und die Parameter für den Abruf des Access-Token eingeben. Die Parameter wurden am Anfang des Dokuments beschrieben. Des Weiteren ein Beispiel in der nachfolgenden Abbildung.



4. Beispiel 3rd-Party-Registration durchführen

Um einen Request auszuführen, ruft man den gewünschten Request aus der Collection auf. Wie zum Beispiel die 3rd-Party-Registration. Zum Ausführen des Requests muss unter dem Reiter Authorization der abgerufene Access-Token eingestellt und im Reiter Body der Request mit Werten gefüllt werden.

The screenshot displays the Postman REST client interface. On the left, a sidebar shows a collection named 'Tobacco Track&Trace ID Issuer API' with 15 requests. The 'operator' folder is expanded, and the request 'Submit an order for a 3rd party e...' is selected. The main panel shows the details of this POST request. The URL is set to '{{baseUrl}}/operator'. The 'Body' tab is active, showing a raw JSON payload. Below the body, the 'Test Results' section shows a status of 200. The response body is also visible, showing a JSON object with an 'idList' array containing one item with details like 'position', 'eoId', and 'confirmationCode'.

```
POST Submit an order for a 3rd party economic operator registration. Multiple EOs may be reg

POST {{baseUrl}}/operator

Params Authorization Headers (2) Body Pre-request Script Tests

none form-data x-www-form-urlencoded raw binary JSON (application/json)

1 {
2   "position": "1",
3   "details": {
4     "eoName1": "Edeka Kreuzberg",
5     "eoAddress": {
6       "street": "Oranienstraße",
7       "houseNumber": "81",
8       "zipCode": "10696",
9       "city": "Berlin"
10    },
11    "eoCountryReg": "DE",
12    "vatR": "true",
13    "eoExciseNumber1": "true",
14  }
}

Body Cookies Headers (13) Test Results Status: 200

Pretty Raw Preview JSON

1 {
2   "idList": [
3     {
4       "position": 1,
5       "eoId": "000+1!743121440414",
6       "confirmationCode": "E0V30Y"
7     }
8   ]
9 }
```