Autorisierung

Was ist OAuth?

OAuth (Open Authorization) ist ein offenes Protokoll, das eine standardisierte, sichere API-Autorisierung für Desktop-, Web- und Mobile-Anwendungen erlaubt.

Token-System

OAuth verwendet Tokens zur Autorisierung eines Zugriffs auf geschützte Ressourcen. Dadurch kann einem Client Zugriff auf geschützte Ressourcen gewährt werden, ohne die Zugangsdaten des Dienstes an den Client weitergeben zu müssen.

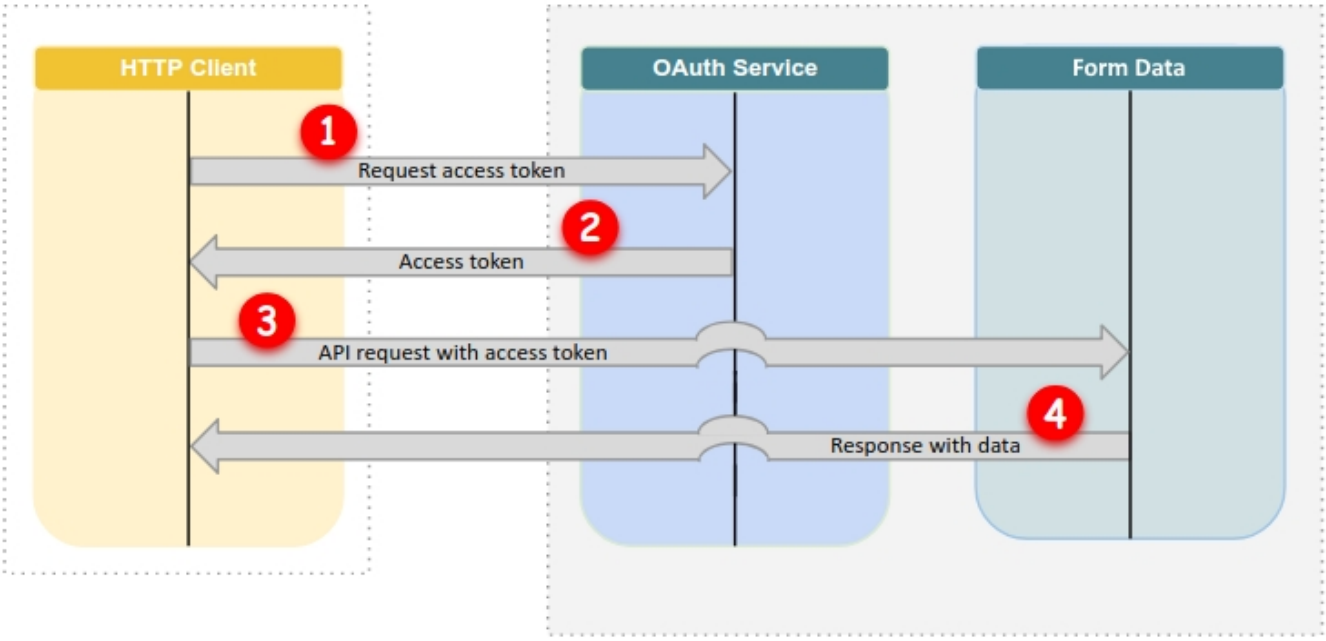
Access-Token

Um auf geschützte Daten auf dem Resource Server zuzugreifen, muss ein Access-Token vom Client als Repräsentation der Autorisierung übermittelt werden. Mittels des Parameters scope können die mit dem Access-Token verbundenen Berechtigungen festgelegt werden. Zum einen kann der Client gewünschte Berechtigungen beim Authorization Server anfragen, zum anderen teilt dieser die gewährten Berechtigungen mit. Der Access-Token hat eine zeitlich begrenzte Gültigkeit.

Refresh-Token

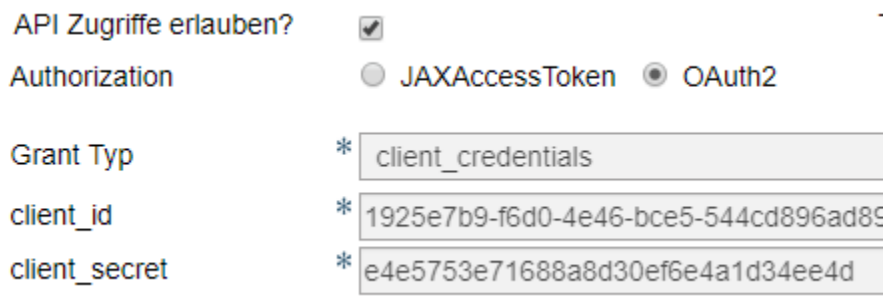
Ein Refresh-Token kann dazu verwendet werden beim Authorization Server einen neuen Access-Token anzufragen, falls der Access-Token abgelaufen oder ungültig geworden ist. Der Refresh-Token hat ebenfalls eine zeitlich begrenzte Gültigkeit. Diese wird in der Regel höher gewählt als die des Access-Token. Der Refresh-Token wird wie der Access-Token nach der Autorisierung durch den Resource Owner vom Authorization Server an den Client gesendet. Da der Refresh-Token selbst schon die Autorisierung des Resource Owners repräsentiert, muss für diese Neuanfrage eines Access-Tokens keine weitere Autorisierung des Resource Owners mehr eingeholt werden.

Der Einsatz von Access-Token und Refresh-Token besitzt den Vorteil, dass die Lebensdauer des Access-Tokens gering (wenige Minuten) gehalten werden kann und somit die Sicherheit des Protokolls erhöht wird. Dieses lässt sich durch folgendes Szenario begründen: Unter der Bedingung, dass der Resource Server die Autorisierung nur bei der ersten Anfrage überprüft, würde ein Rechteentzug keine Folgen haben. Ein Zugriff auf Daten und Dienste beim Resource Server wäre dann für den Client weiterhin möglich. Da jedoch die Lebenszeit des Access-Tokens nur wenige Minuten beträgt, würde ein späteres Erlangen des Access-Tokens durch einen Angreifer keine weitreichenden Folgen haben.



JAXForms API und OAuth2

Die REST und SOAP Schnittstellen werden ab Version 4.80.2 über OAuth2 mit dem Grant Type Client Credentials geschützt. Eine Client ID und ein Client Secret kann über die Benutzerverwaltung erstellt werden:



Ein Access Token kann via Token Service bezogen werden:

[service-url]/formservice/services/rest/auth/token

Method	POST
HTTP Request Header	Content-Type: application/json

HTTP Request-Body

```
{
  "grant_type": "client_credentials",
  "client_id": "[Client ID]",
  "client_secret": "[Client Secret]"
}
```

Antwort des Servers

```
{
    "access_token": "[Access Token]",
    "expires_in": "3600000",
    "token_type": "Bearer"
}
```

Autorisierte REST/SOAP Anfrage

Das über den Token Service angefragte Access Token kann nun für die Autorisierung geschützter Schnittstellen verwendet werden:

HTTP Request Header	authorization: Bearer [Access Token]

Beispiel in Request-Header:

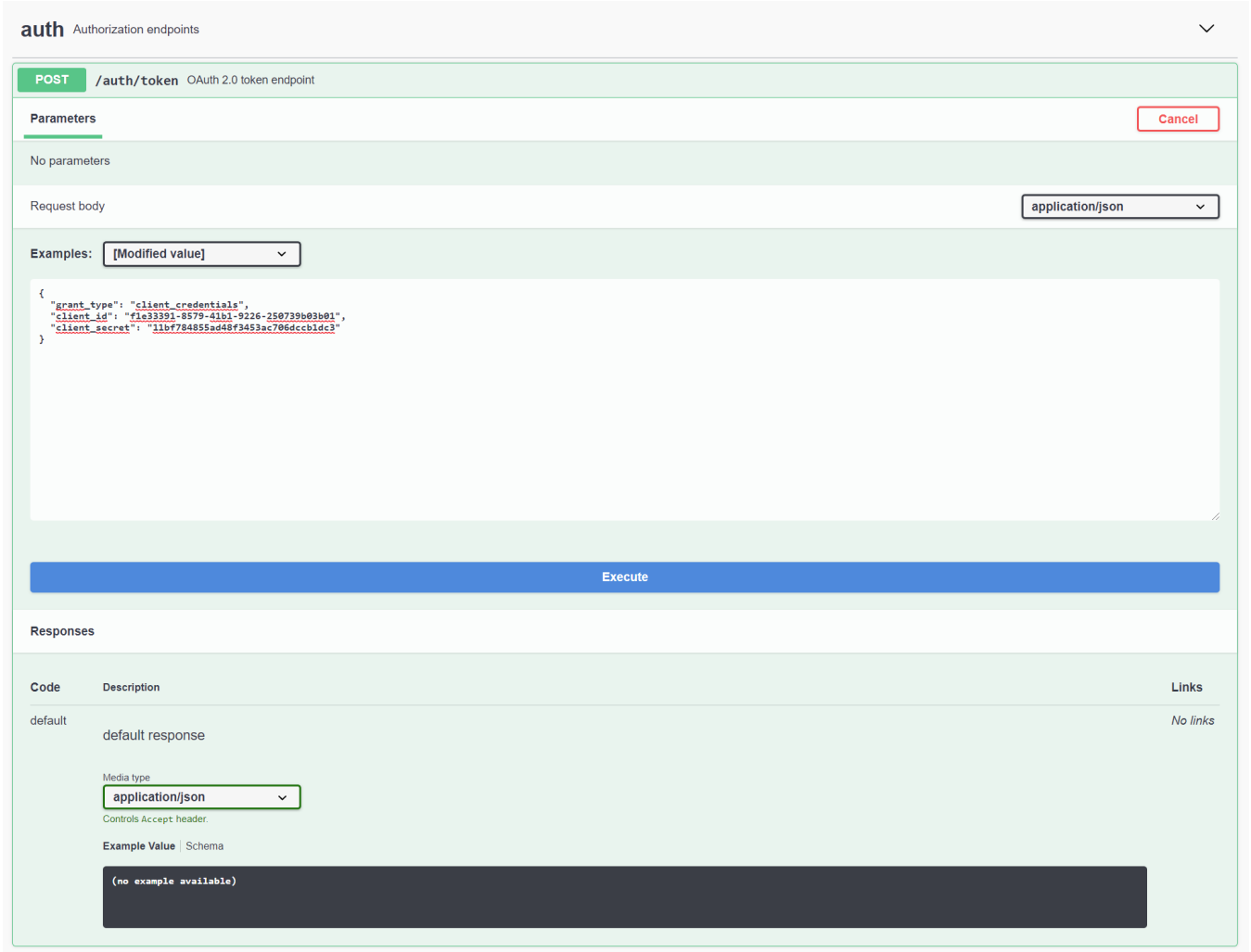
KEY	VALUE
Authorization	Bearer eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiI5ZWNlOTI5Yy0yMzkyLTRkMjctYmE0OC00MDY5MmI2OWRkYzEiL...

Hier befindet sich die offizielle Spezifikation des Bearer Tokens.

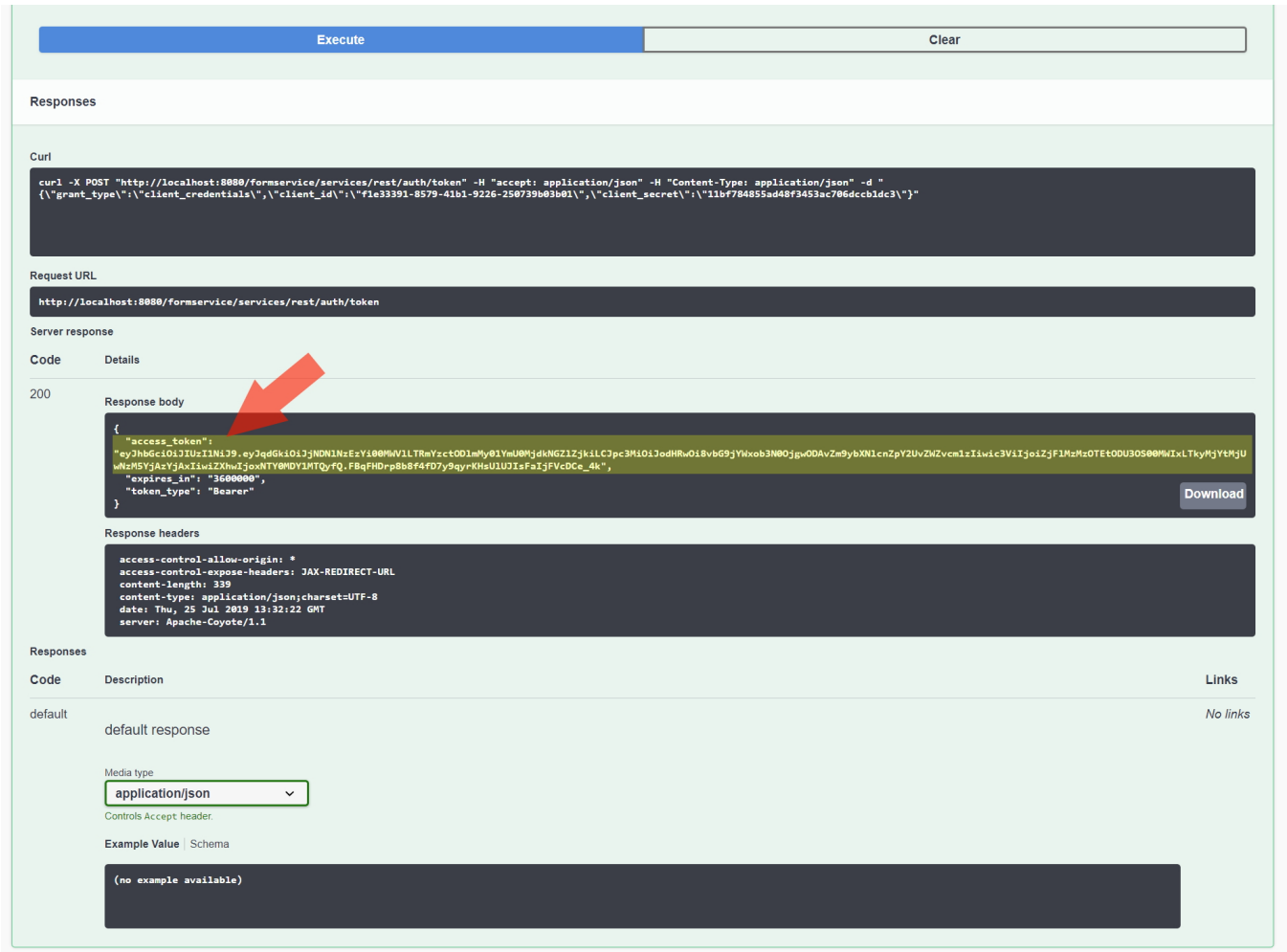
Ein Access Token ist für 60 Minuten gültig. Danach muss ein neues Token angefordert werden.

OAuth mit Swagger UI

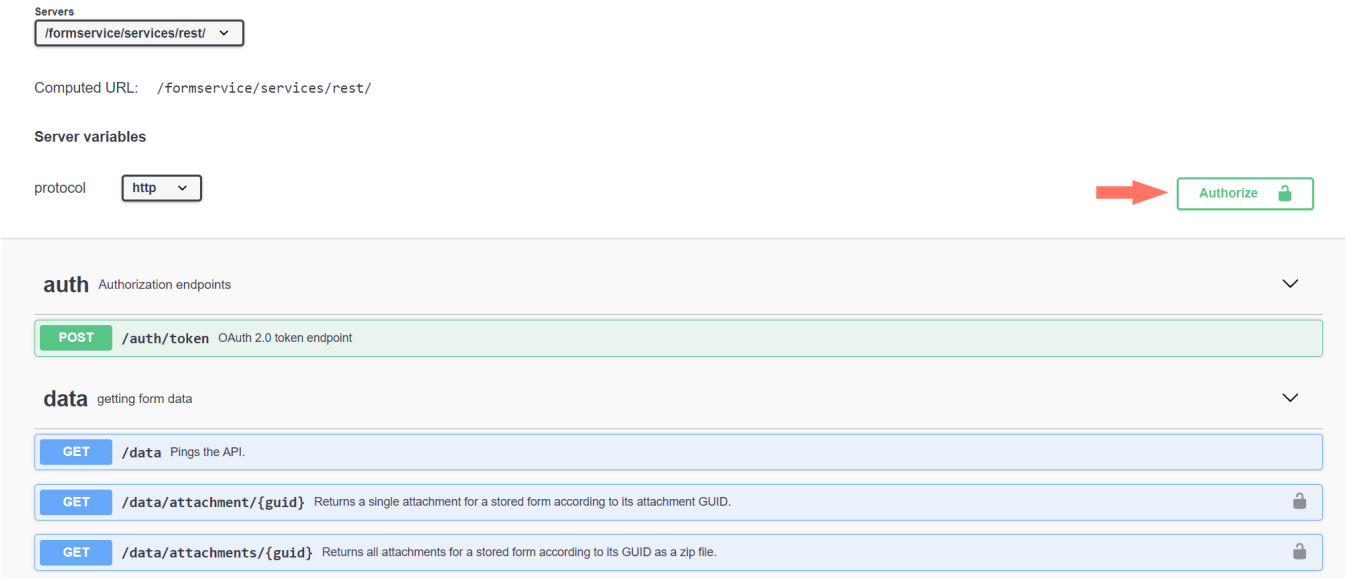
Über Swagger UI kann das Token ebenfalls erzeugt werden, um die REST-API auszutesten. Dazu mit einem Klick auf den Eintrag "/auth/token" den Bereich ausklappen und via "*Try it out*"-Button weitere Eingabefelder einblenden. Nun können im Request Body die oben beschriebenen Parameter eingetragen werden:

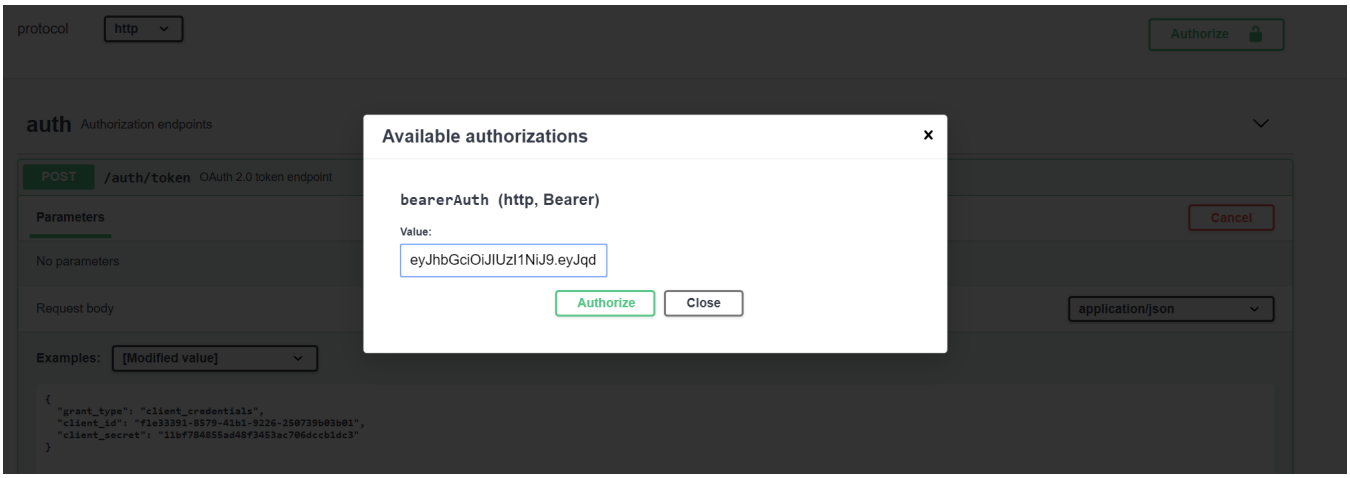


Danach erhält man mit einem Klick auf "*Execute*" eine Antwort vom Server, welche das Token (im Bild gelb markiert) beinhaltet:



Dieses wird dann für weitere API-Aufrufe in Swagger UI hinterlegt. Dazu auf den Button "*Authorize*" am Anfang der Seite klicken und im aufgehenden Fenster das Token hinterlegen:





Danach mit dem Button "*Authorize*" die Eingabe bestätigen. Nun können die API-Aufrufe via Swagger UI getätigt werden.