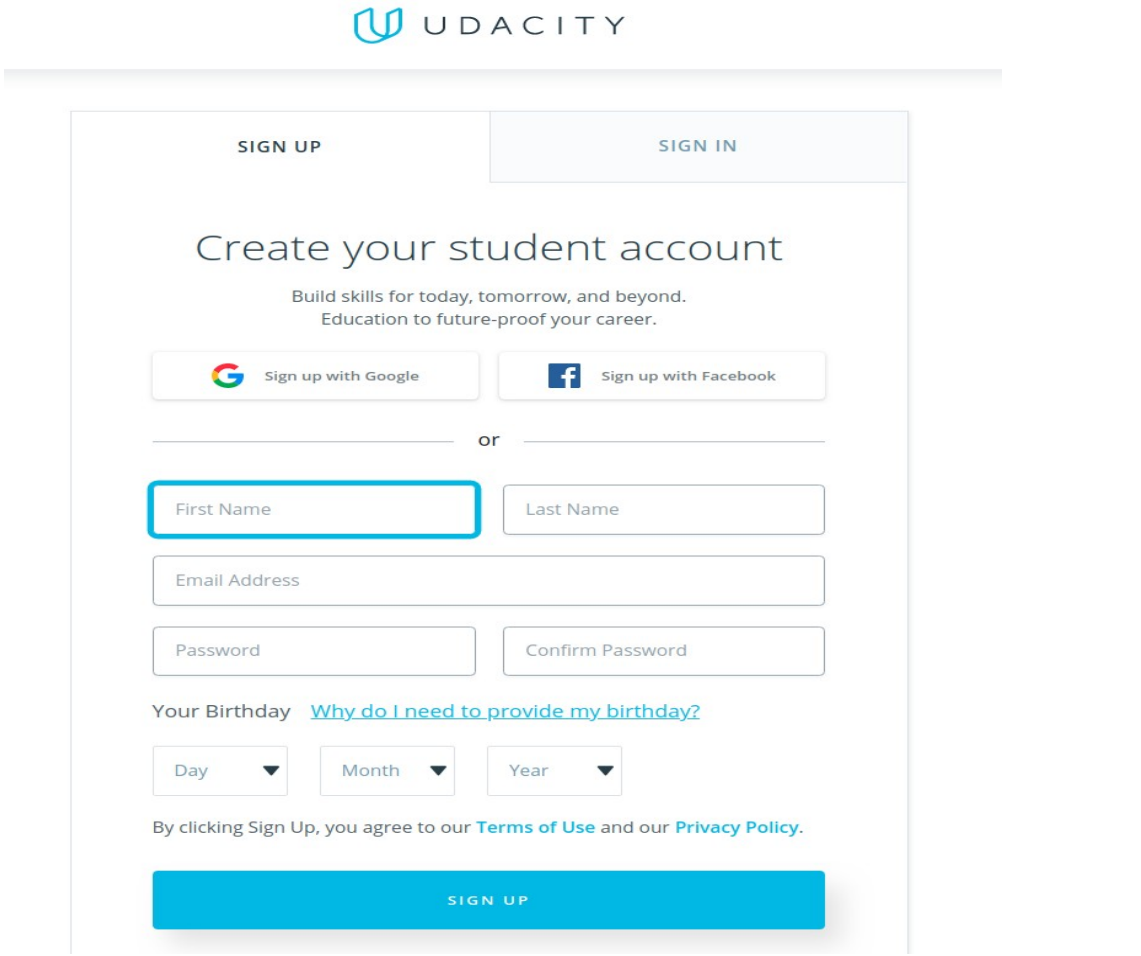
B56 Projektstudium - Softwareentwicklung anhand eines Big Data Projekts

Thema: OpenID Connect

Lilit Patzwaldt-Sergoyan
Matr.Nr.: s0561376

Inhaltsverzeichnis


UDACITY
SIGN UP
SIGN IN
Create your student account
Build skills for today, tomorrow, and beyond.
Education to future-proof your career.
Sign up with Google
Sign up with Facebook
or
First Name
Last Name
Email Address
Password
Confirm Password
Your Birthday
Why do I need to provide my birthday?
Day
Month
Year
By clicking Sign Up, you agree to our Terms of Use and our Privacy Policy.
SIGN UP

Bestellübersicht

Sie bestellen 1 Artikel	21,99
Standardversand	0,00
Gesamtsumme (inkl. MwSt.)	**21,99**

Weiter einkaufen

Zur Kasse

oder

Direkt zu PayPal

(Identity, Authentication) + OAuth 2.0 = OpenID Connect

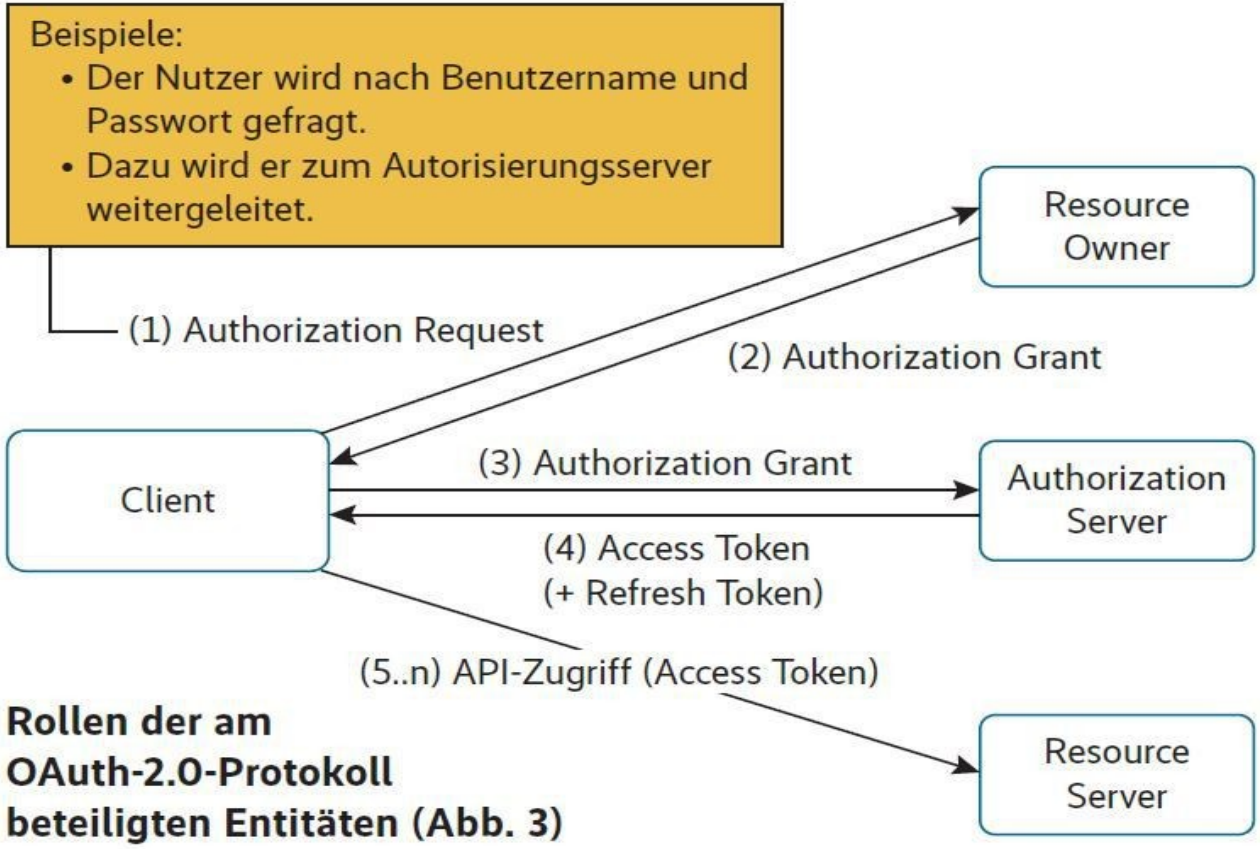
https://openid.net/connect/faq/

Kurz über OAuth 2.0

- Ein Authoriesierungsprotokoll
- bietet Access Tokens an, um die Implementierung einer API von der eigentlichen Authentifizierung des Benutzers und der Autorisierung des Zugriffs zu trennen.
- Die Sicherheit beruht auf Verwendung von HTTPS und Zufallswerten
- Im Mittelpunkt steht der Schutz von Ressourcen eines Benutzers, zum Beispiel Dokumenten oder Adressbucheinträgen, vor unbefugtem Zugriff.

https://www.heise.de/developer/imgs/06/1/2/4/4/0/5/4/illustration-5e772acd1ca11b05.jpeg

OAuth 2.0



Rollen der am OAuth-2.0-Protokoll beteiligten Entitäten (Abb. 3)

https://www.heise.de/developer/imgs/06/1/1/4/6/2/8/8/abb3-387e8f4063a2dfa9.jpeg

Begriffe

1) Client = jede Anwendung, die im Auftrag des Benutzers auf geschützte Ressourcen desselben zugreift
2) Resource owner (=Benutzer)
3) Resource Server: akzeptiert und durchführt nur vom jeweiligen Benutzer autorisierte Zugriffe
4) Access Token = erlaubt es dem Resource Server festzustellen, welche Rechte der Benutzer an die Anwendung delegiert hat. Wenn es abläuft, lässt sich ein sogenanntes Refresh Token verwenden, welches das Abfragen eines neuen Access Token ohne Benutzerinteraktion ermöglicht.
5) Authorization Server = stellt Access Token im Namen des Benutzers aus. Authentifiziert den Benutzer und holt dessen Autorisierung für die Durchführung des Zugriffs ein.

https://www.heise.de/developer/artikel/Flexible-und-sichere-Internetdienste-mit-OAuth-2-0-2068404.html

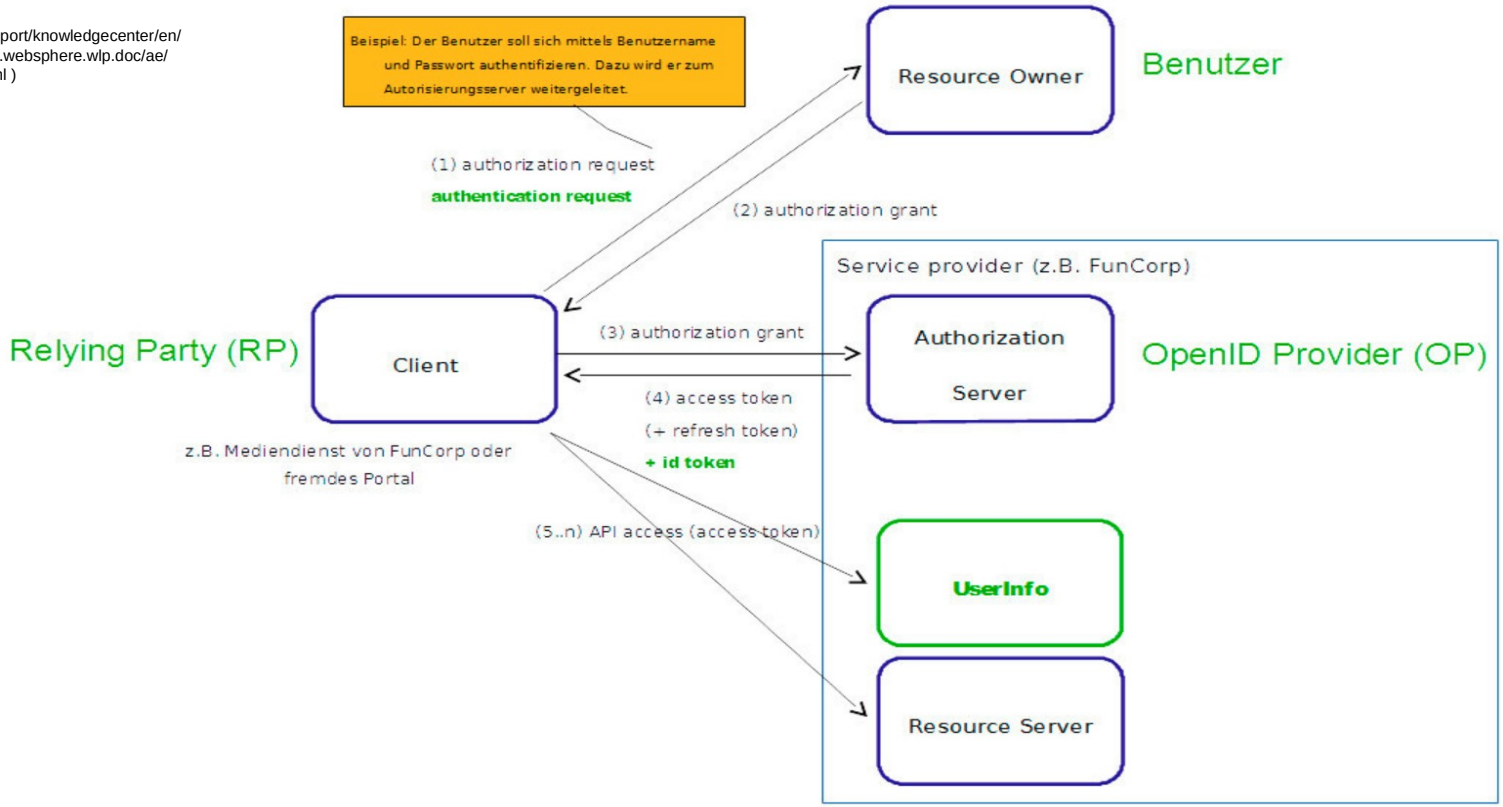
Der Ablauf

1. Client → Resource owner: Authorization request (die Anfrage an den Benutzer sich mit Login & Passwort anzumelden)
2. Resource owner→ Client: Authorization grant (der Benutzer meldet sich an / stimmt den Zugriff zu, damit wird Authorization Grant ausgehändigt)
3. Client → Authorization Server: Authorization Grant
4. Authorization Server → Client: Access Token (nachdem der Client den Grant an Authorization Server sendet, bekommt es den Access Token)
5. Client → Ressource Server: API-Zugriff(Access Token)

https://www.heise.de/developer/artikel/Flexible-und-sichere-Internetdienste-mit-OAuth-2-0-2068404.html

OAuth 2.0 und OpenID Connect

ID-Token = Ein JSON Web Token (JWT), das Ansprüche zum authentifizierten Benutzer enthält.
(https://www.ibm.com/support/knowledgecenter/en/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_openid_connect.html)



https://www.heise.de/developer/imgs/06/1/2/4/4/0/5/4/illustration-5e772acd1ca11b05.jpeg

Authentication Request

HTTP/1.1 302 Found
Location: https://openid.c2id.com/login?
response_type=code
&scope=openid
&client_id=s6BhdRkqt3
&state=af0ifjsldkj
&redirect_uri=https%3A%2F
%2Fclient.example.org%2Fcb

ID Token

```
{
  "sub"       : "alice",
  "iss"       : "https://openid.c2id.com",
  "aud"       : "client-12345",
  "nonce"     : "n-0S6_WzA2Mj",
  "auth_time" : 1311280969,
  "acr"       : "c2id.loa.hisec",
  "iat"       : 1311280970,
  "exp"       : 1311281970
}
```

Zusammenfassung

- Erweiterung von OAuth 2.0 um eine Identitätsschicht
- Einfach implementierbar, mit einem einfachen HTTP Request kann ein komplettes Login inklusive Zugriff auf Benutzerdaten implementiert werden
- Der Entwickler muss sich nicht um Kryptographie kümmern, die Verbindungen über OAuth 2.0 sind HTTPS gesichert.
- Authentifizierungsmethode kann leicht geändert werden ohne den Client umschreiben zu müssen.

Fragen?

Quellen

https://openid.net/connect/

https://www.heise.de/developer/artikel/Flexible-und-sichere-Internetdienste-mit-OAuth-2-0-2068404.html

https://www.ibm.com/support/knowledgecenter/SSEQTP_liberty/com.ibm.websphere.wlp.doc/ae/cwlp_openid_connect.html

https://www.heise.de/developer/artikel/OpenID-Connect-Login-mit-OAuth-Teil-1-Grundlagen-2218446.html?seite=all

https://connect2id.com/learn/openid-connect