

OAuth 2.0 – Handout

Autorisierungsstandard für API Endpunkte. Bietet Autorisierung für Desktopanwendungen, Webanwendungen und mobile Anwendungen. Konzentriert sich dabei auf die Einfachheit in der Cliententwicklung, kann aber auch spezifiziert für komplexere Anwendungen eingesetzt werden.

Protocol Flow



- Client: Anwendung, welche Zugriff zu geschützten Daten möchte, braucht aber Erlaubnis von Resource Owner (Authorization Request)
- Resource Owner: Eigentümer der geschützten Daten, teilt Erlaubnisse aus, um auf seine Daten zuzugreifen (Authorization Grant)
- Authorization Server: Erteilt das Access Token, mit welchem der Client Zugriff auf den Resource Server erhält, Speicherort der angefragten Daten (meistens eine API)

Wie erhält man das Authorization Grant

Meist verbreiteten Varianten sind:

Authorization Code	Client Credentials	Refresh Token
<ul style="list-style-type: none">• Autorisierungsserver als Vermittler zwischen client und resource owner• Autorisierungsdaten des Resource Owner werden dem Client niemals direkt offenbart	<ul style="list-style-type: none">• Client_id und client_secret autorisieren den Client• Vergleichbar mit Benutzername und Passwort	<ul style="list-style-type: none">• Nach Ablauf des eigentlich Access Tokens kann damit ein Neues angefragt werden• Verringert die Größe der Anfrage da Client bereits autorisiert war• Wird nur bei vertraulichen Clients genutzt

Quellen

- <https://datatracker.ietf.org/doc/html/rfc6749> (letzter Aufruf: 16.04.2023)
- <https://oauth.net/2/grant-types/> (letzter Aufruf: 16.04.2023)
- <https://oauth.net/2/client-authentication/> (letzter Aufruf: 16.04.2023)
- <https://oauth.net/2/bearer-tokens/> (letzter Aufruf: 16.04.2023)
- <https://auth0.com/de/intro-to-iam/what-is-oauth-2/> (letzter Aufruf: 16.04.2023)