entwickler.press
shortcuts

OAuth 2.0

Client & Server

Sven Haiges

Sven Haiges

OAuth 2.0

Client und Server

ISBN: 978-3-86802-457-9

Ein Imprint der Software & Support Media GmbH

1 OAuth 2.0 – die Clientseite

Facebook, Google, Foursquare oder Pinterest haben eines gemeinsam: Die APIs dieser Dienste setzen allesamt auf OAuth 2.0. OAuth 2.0 ist ein Protokoll zur Autorisierung von API-Zugriffen, beispielsweise durch server- oder clientseitige Webanwendungen oder mobile Apps. Trotz des spektakulären Rückzugs von OAuth-2.0-Editor Eran Hammer werden wohl auch in Zukunft mehr und mehr APIs auf dieses Protokoll setzen: OAuth 2.0 hat sich bereits bei den großen Services (Google, Facebook) etabliert und ist im Vergleich zu OAuth1 viel einfacher zu benutzen. Wir betrachten zunächst den wichtigsten OAuth 2.0 Grant, den Authorization Code Grant.

Zugegeben: Der Blog Post, mit dem sich Eran Hammer von der OAuth-2.0-Spezifikation verabschiedet hat [1], war für viele schockierend zu lesen. Als Hauptgrund für seinen Rückzug gibt er „Indecision Making“ an, also das Fehlen von Entscheidungen. Die OAuth-2.0-Spezifikation sei wegen der Mitwirkung zu vieler Parteien zu offen, zu viele Bereiche seien absichtlich ungenau gehalten und diverse Implementierungsmöglichkeiten offen zu halten. Viele dieser Gründe sind im Kern sicherlich richtig. Dennoch bewegt sich die Welt der APIs mit großen Schritten in Richtung OAuth 2.0. OAuth 2.0 ist ein sicheres und gut durchdachtes Protokoll, auch wenn manche sich vielleicht Details besser spezifiziert gewünscht hätten. Und auch wenn die APIs von Google und Facebook derzeit und vielleicht auch für immer nicht hundertprozentig standardkonform sein werden: OAuth 2.0 – in leicht unterschiedlichen Ausprägungen – ist bereits überall im Einsatz. In diesem Kapitel möchte ich nicht allzu lange philosophieren. Fakt ist, dass Kenntnisse über OAuth 2.0 relevanter sind denn je. Egal ob Mobile, klassischer Webclient oder moderne In-Browser-JavaScript/HTML5-App – um den Clientcode zu autorisieren und damit mit Zustimmung des Anwenders auf seine Ressourcen zuzugreifen, benutzt man

OAuth 2.0. Und auch wenn die verschiedenen OAuth-2.0-kompatiblen APIs noch leichte Unterschiede aufweisen, so muss man OAuth 2.0 nur einmal aus der Vogelperspektive verstanden haben.

Der erste Teil dieses OAuth-2.0-shortcuts geht auf die Sicht der OAuth-2.0-Clients ein und stellt Ihnen besonders den allgegenwärtigen OAuth 2.0 Authorization Code Grant vor. Im Laufe des Textes werden zahlreiche Codebeispiele die praktische Anwendung aus der Sicht des Clients und Servers beleuchten. Die Codebeispiele für die Benutzung aus Sicht des Clients sind dabei einer Gaelyk Webapplikation entnommen, und meistens handelt es sich um Code aus Gaelyk-Controllern. Weitere Informationen zu Gaelyk, einem schlanken Webframework für Google App Engine, finden Sie unter [2]. Neben den OAuth-2.0-„Flows“ erhalten Sie auch Informationen dazu, welcher Flow für welches Anwendungsszenario am besten geeignet ist.

Rollen

Um OAuth 2.0 besser verstehen zu können, sollte man die Rollen dieses Autorisierungsprotokolls verstanden haben. Es gibt den Resource Owner, den Resource Server, den Client und den Authorization Server.

- Der **Resource Owner** ist oftmals der Endanwender, beispielsweise ein Benutzer, der auf seine in der Cloud gespeicherten Bilder zugreifen möchte.
- Der **Resource Server** ist ein Server, auf dem die Daten/Dienste des Resource Owners vorliegen. Diese Ressourcen sind dadurch geschützt, dass der Resource Server ein so genanntes *Access Token* verlangt. Nur wenn das Access Token validiert werden konnte, genehmigt der Resource Server den Zugriff.
- Der **Client** ist eine Applikation, die für den Resource Owner auf die Daten/Dienste zugreift. Um auf diese zugreifen zu können, muss er im Besitz eines Access Tokens sein.