



DHBW MANNHEIM

TEAM „NORBERT“

Norbert - Your StudyBuddy

Softwareentwurf

28. März 2016

Projektleitung:
Projektmitglieder:

Arwed Mett
Dominic Steinhauser, Tobias Dorra,
Simon Oswald, Philipp Pütz

Inhaltsverzeichnis

1	Anbindung externer Dienste - Dropbox	1
1.1	Autorisierungsprozess (OAuth)	1
1.2	Dropbox Endpoints	2
1.3	Webhooks & Benachrichtigungen	3

1 Anbindung externer Dienste - Dropbox

Eines der Ziele von Norbert - Your StudyBuddy ist es Wissensmanagement zu betreiben und somit Wissen zu sammeln und weiterzugeben. Da Informationen und Wissen von Studierenden einer DHBW meistens in Dropbox-Ordner abgelegt werden, wird in diesem Kapitel die technische Anbindung von Dropbox an Norbert beschrieben. Dabei wird zunächst erst allgemein auf den standardisierten Autorisierungsprozess „OAuth“ eingegangen und daraufhin die verschiedenen Dropbox Endpoints mit ihren Schnittstellen vorgestellt.

1.1 Autorisierungsprozess (OAuth)

OAuth ist ein offenes Protokoll, dass eine standardisierte und sichere API-Autorisierung für mobile Endgeräte, Webanwendungen und Desktop-Applikationen ermöglicht. Der Benutzer kann über den OAuth-Autorisierungsprozess (Abbildung 1.1) einer Anwendung (in diesem Fall Norbert) Zugriff auf gespeicherte Daten erlauben. Dabei loggt sich der Benutzer über die Applikation (Norbert) die Zugriff benötigt, auf dem Dienst, der die freizugebenden Daten speichert, ein (hier Dropbox) und erlaubt der Applikation den Zugriff. Dabei wird ein spezieller App key mit entsprechendem App secret und ein Access-Token zwischen der Applikation und dem Dienst ausgetauscht. Der Access-Token wird individuell für jeden Benutzer erstellt und wird bei API-Aufrufen mitgesendet. Dieser dient somit zur Authentifizierung des Benutzers, sodass kein Benutzername oder Passwörter ausgetauscht oder gespeichert werden müssen.

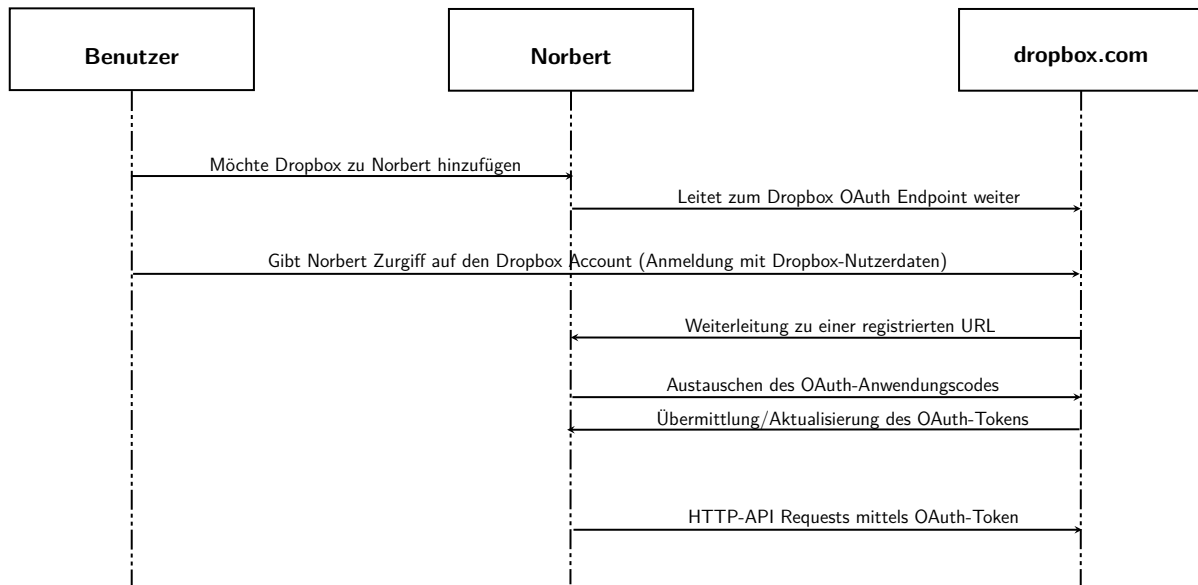


Abbildung 1.1: OAuth-Autorisierungsprozess zwischen Norbert und Dropbox.com

1.2 Dropbox Endpoints

Die Dropbox HTTP-API besteht aus mehreren Endpoints (Abbildung 1.2), die jeweils unterschiedliche Funktionen bereitstellen. Alle API-Aufrufe benötigen zur Autorisierung den individuellen Access-Token eines Benutzers.

1. *dropbox.com*: Über diesen Endpoint/Webseite wird die OAuth-Autorisierung durchgeführt. Es wird nur einmalig zur Generierung des Access-Tokens ein Kontakt zu diesem Endpoint hergestellt.
2. *api.dropboxapi.com (RPC-Endpoint)*: Über die Domain *api.dropboxapi.com* können RPCs (remote procedure calls) ausgeführt werden. Dabei nimmt dieser Endpoint JSON-Strings im HTTP-Body entgegen und führt je nach aufgerufener URL verschiedene Funktionen aus. Ein RPC-Funktion ist beispielsweise *list_folder* (https://api.dropboxapi.com/2/files/list_folder), die als Parameter einen Pfad erwartet und dann alle im entsprechenden Ordner sich befindenden Dateien auflistet. Das Ergebnis der Funktion wird daraufhin wieder im JSON-Format zurückgesendet.
3. *content.dropboxapi.com (Download/Upload-Endpoint)*: Wie der Name schon sagt, können über den Download/Upload-Endpoint Dateien heruntergeladen oder hochgeladen werden. Die benötigten Meta-Daten (Pfad, Datei ID etc.) werden dabei im HTTP-Header als JSON oder

URL-Argument übergeben. Funktionsergebnisse werden im JSON-Format im HTTP-Response Header übergeben und Dateien im HTTP-Body übersendet.

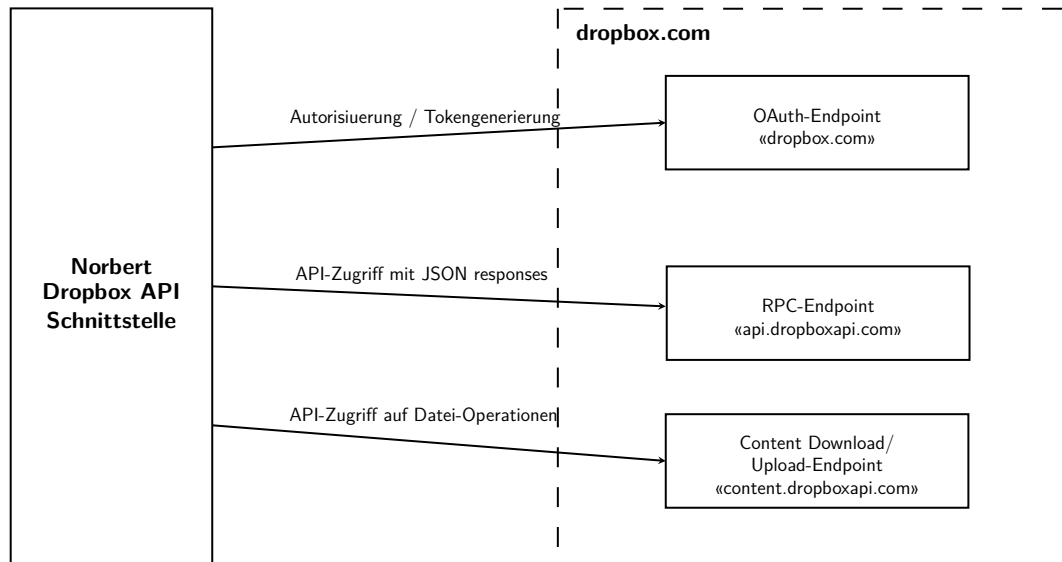


Abbildung 1.2: API-Zugriffe über die Dropbox-Endpoints

1.3 Webhooks & Benachrichtigungen

Die Benutzer von Norbert wollen nicht jedes mal wenn Sie Änderungen an ihrer Dropbox vornehmen, diese manuell an Norbert weitermelden. Stattdessen sollte dies automatisch vonstatten gehen. Dazu existieren zwei Architekturansätze:

1. Norbert „scannt“ automatisch in bestimmten Zeitintervallen die Dropbox nach Änderungen (keine zusätzlichen Komponenten nötig).
2. Dropbox benachrichtigt Norbert vollautomatisch bei Änderungen mittels Webhooks.

Beide Ansätze erfüllen ihren Zweck und haben ihre Vor- und Nachteile. Der erste Ansatz ist einfach umzusetzen und kann beispielsweise jede Nacht ausgeführt werden. Doch der zweite Ansatz kann Norbert schneller auf Änderungen reagieren lassen, benötigt aber zusätzliche Komponenten. Da schnelles Wissensmanagement im Vordergrund steht wird in diesem Kapitel auf die benötigten Komponenten zur Implementierung des zweiten Ansatzes eingegangen.

Webhooks sind ein nicht standardisiertes Verfahren um Echtzeitbenachrichtigungen zwischen Servern und Software auszutauschen. Die Kommunikation findet dabei über HTTP-POST-Requests statt. Abbildung 1.3 zeigt den Ablauf der Benachrichtigungen und die daraufhin angestoßenen Prozesse.

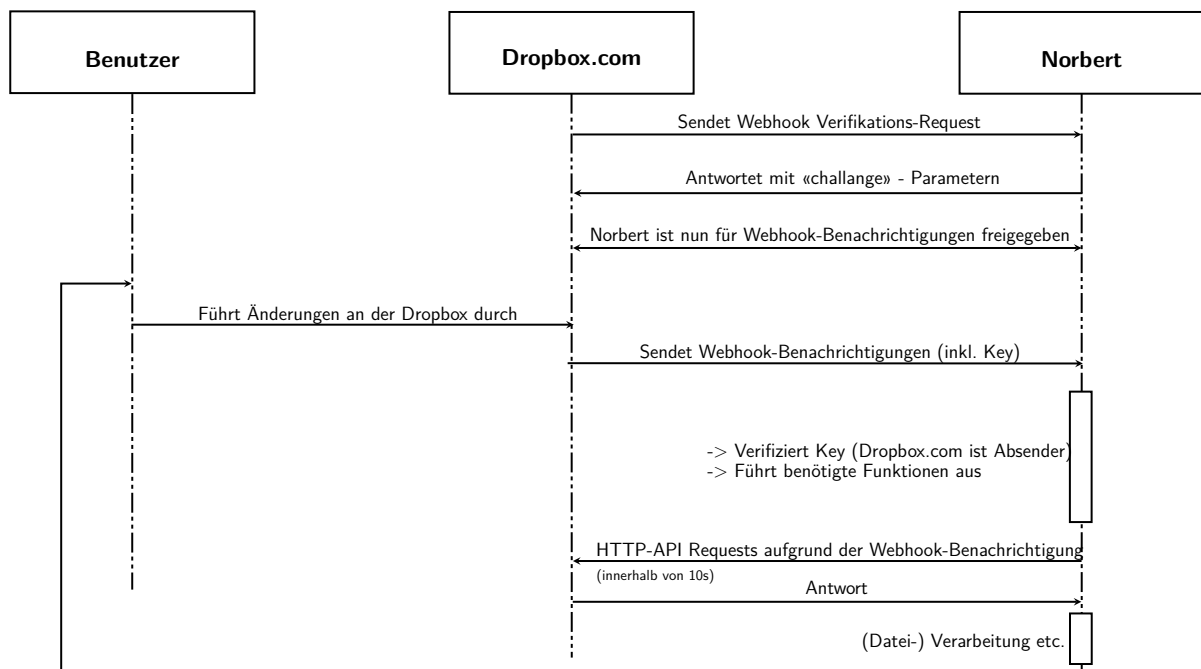


Abbildung 1.3: API-Zugriffe über die Dropbox-Endpoints