akana
Powering the API Economy

Using Dropbox's 3-Legged OAuth 2 Security with the Akana Dropbox API Hook

1. Overview 3
2. The 3-Legged Dance Explained 3
3. Making a Dropbox API hook request.. 4

1. Overview

This document describes how the Akana Dropbox API Hook supports Dropbox's 3-legged OAuth, and how the Hook user can use it to ensure that all Akana Dropbox API Hook requests are successfully Authorised with Dropbox.

Figure 1 shows the entire 3-legged OAuth process. Steps 1 through 7 are the steps in the OAuth 'Dance' that enable the creation of the Dropbox Access Token, and its associated 'key' that is returned to the Hook User. The bottom process shows how the Hook User would use an App to enter the auth key, that would send it in subsequent calls to the Dropbox Hook.

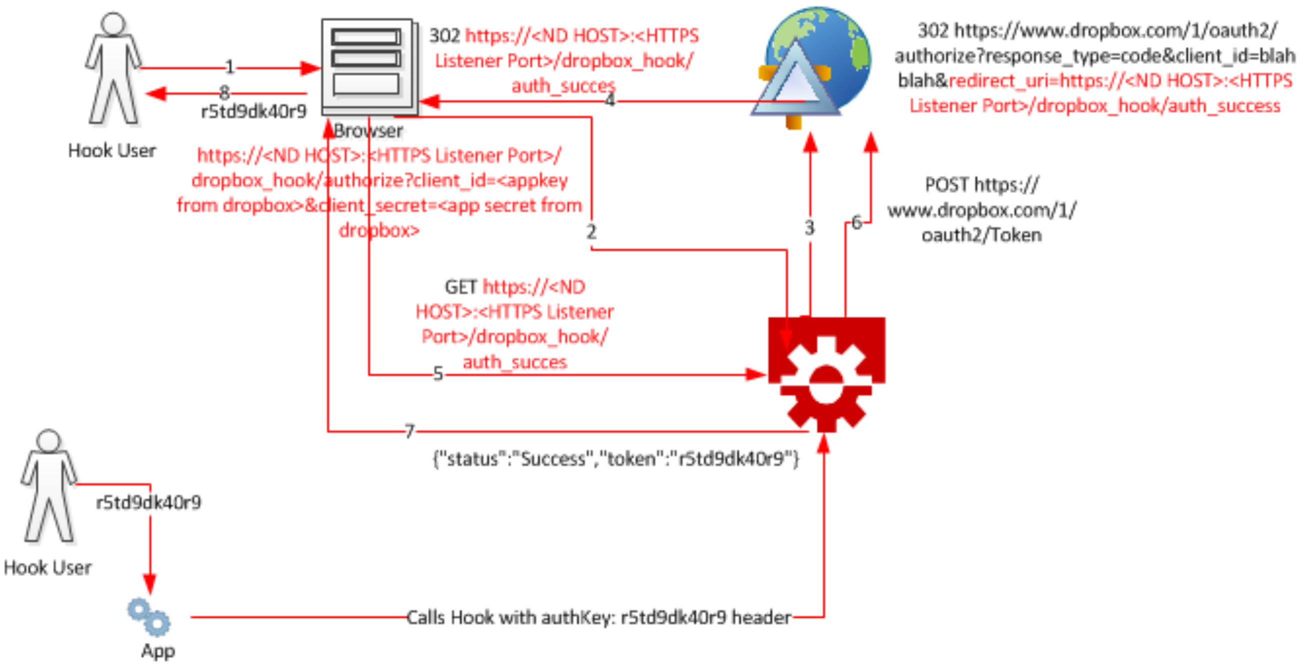
Hook User
1
8
r5td9dk40r9
Browser
302 https://<ND HOST>:<HTTPS Listener Port>/dropbox_hook/auth_succes
4
302 https://www.dropbox.com/1/oauth2/authorize?response_type=code&client_id=blah blah&redirect_uri=https://<ND HOST>:<HTTPS Listener Port>/dropbox_hook/auth_success
https://<ND HOST>:<HTTPS Listener Port>/dropbox_hook/authorize?client_id=<appkey from dropbox>&client_secret=<app secret from dropbox>
2
3
6
POST https://www.dropbox.com/1/oauth2/Token
GET https://<ND HOST>:<HTTPS Listener Port>/dropbox_hook/auth_succes
5
7
{"status":"Success","token":"r5td9dk40r9"}
Hook User
r5td9dk40r9
App
Calls Hook with authKey: r5td9dk40r9 header

Figure 1 - 3-Legged OAuth Client Process

2. The 3-Legged Dance Explained

The Hook user must do this before using the Hook. Go to any browser and make a call to the Dropbox Hook

1. OAuth 2 authorization URL. Specifiying the needed parameters of the client_id and the client_secret**.**
   https://<ND HOST>:<HTTPS Listener Port>/dropbox_hook/authorize?client_id=<appkey from dropbox>&client_secret=<app secret from dropbox>
   Note that the ND Host must be accessible from the Internet and that the Policy Manager Admin must have configured a HTTPS listener.
2. The browser passes the GET request to the Hook that gets the parms from the request and the re-direct URL from the variables policy.
3. The Hook then creates the correct authorize request and sends the 302 to the browser that redirects it to the Dropbox authorizarion server
4. Dropbox either rejects the request because of the client_id, or sends a response back to the browser to

redirect it to the

5. Dropbox API Hook’s Dropbox_API_Hook_Re-Direct VS operation of /auth_success.
6. This Operation has a process that takes the code from the “?code=ur6I9QuI4goAAAAAAAGKYNaE2ggOLGeANM-eUMuUUKQ” request parameter and makes a call to the Dropbox OAuth 2 /token operation.
7. The response from the /token call is either an error of the access token. The access token is then added to a hash table with it’s randomly generated key. This hash table is placed into a Hazelcast based global cache. The key is returned back to the browser in a JSON response, e.g. {"status":"Success","token":"r5td9dk40r9"}
8. The user then stores that key value to be used in all calls to the DropBox API Hook via the “authKey” header.

3. Making a Dropbox API hook request

Any user of the API hook, be that an App, or a mash-up MUST send the authorization key that was the result of authorizing the Resources Owner via a browser in section 2. This must be sent to any of the Dropbox API hook operations as a ‘authKey’ HTTP header.

This key will be used by the API hook to look up the appropriate OAuth 2 access token belonging to that Resource Owner from a global cache.