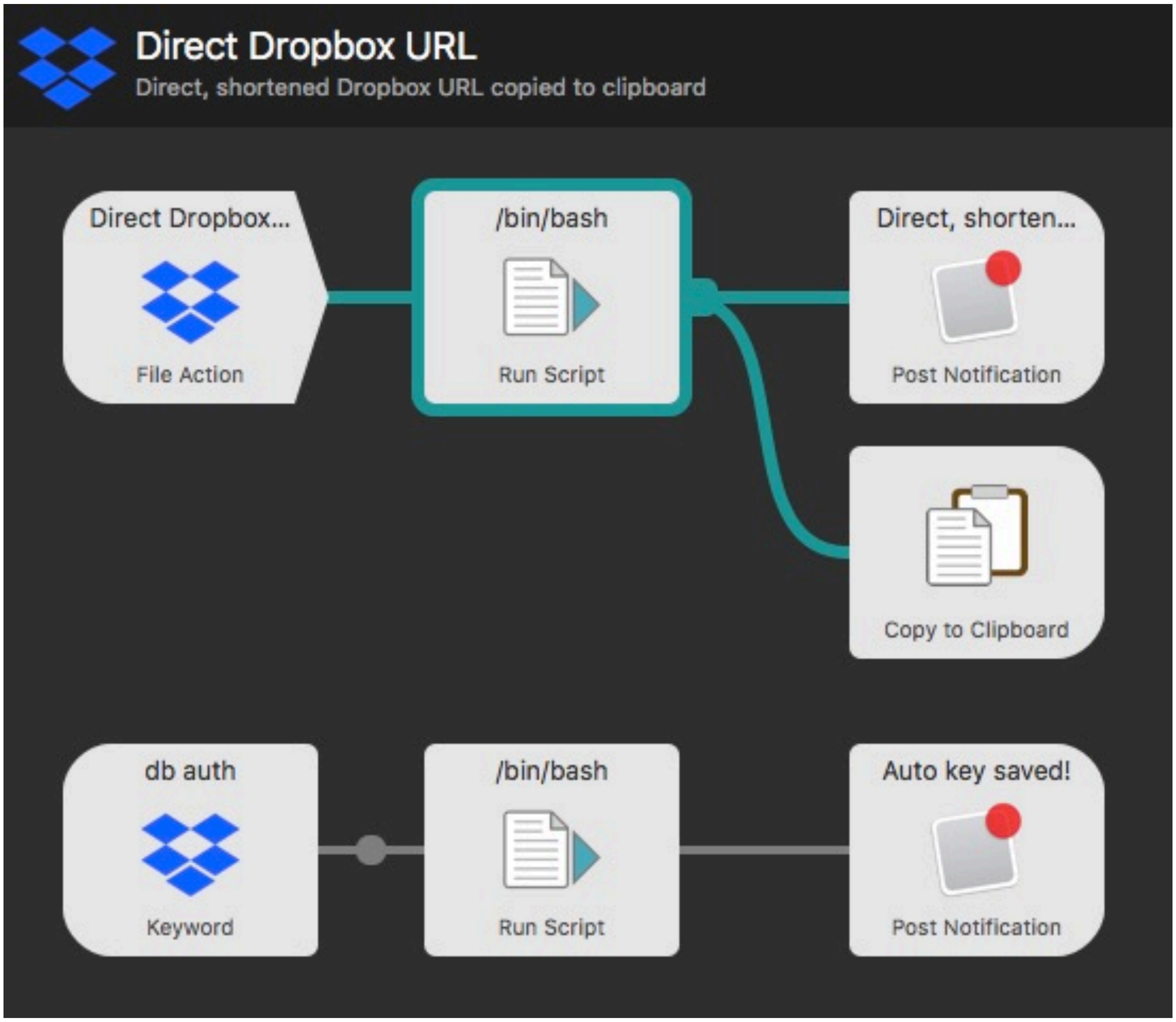
Dropbox link to direct short URL - ALFRED Workflow

This file-action triggered workflow...

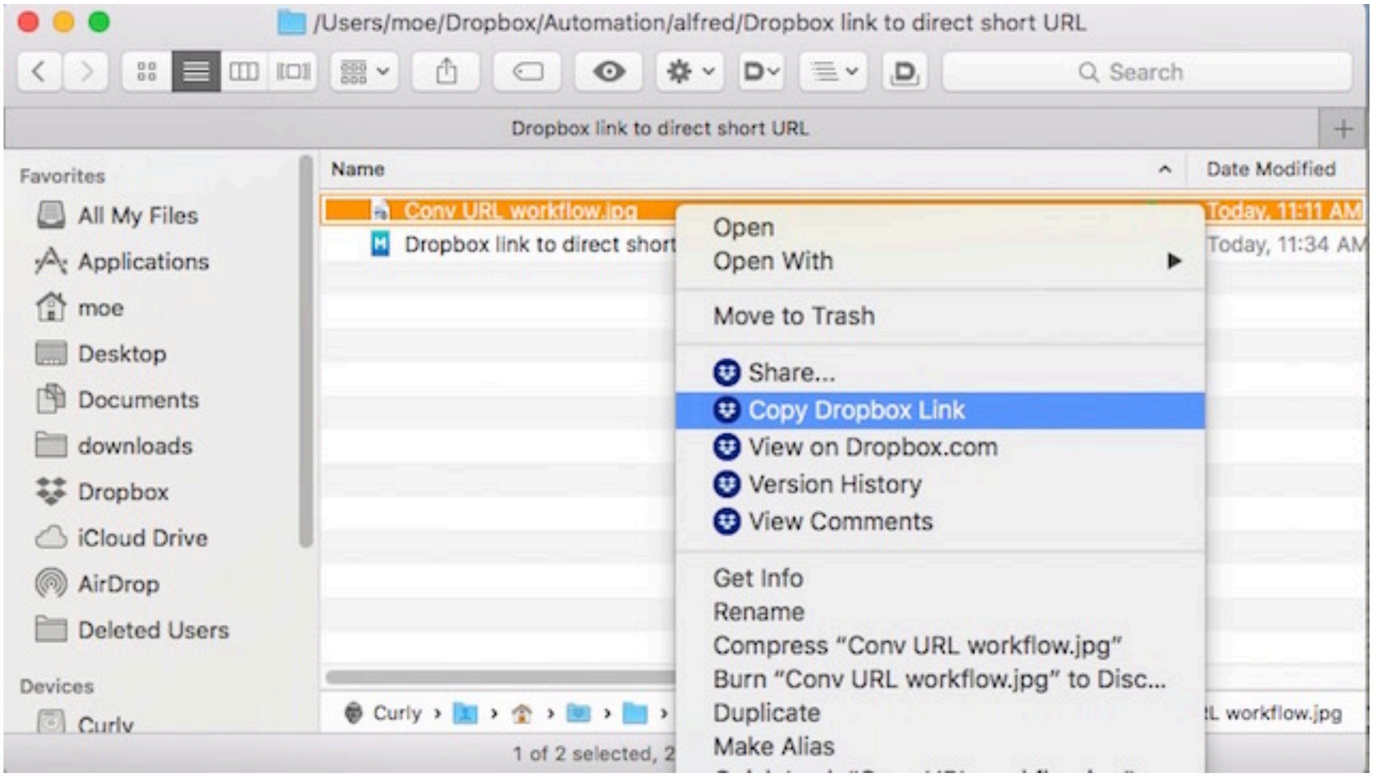
1. Gets the Dropbox link of the selected file
2. Converts the link to a direct URL
3. Shortens the URL and copies the result to the clipboard



For example, the local path to the screenshot above is:

```
~/Dropbox/Automation/alfred/Dropbox link to direct short URL/Conv URL workflow.jpg
```

Right-clicking on the file to get the Dropbox link provides an indirect link that presents the JPEG within an interstitial page framing the image and providing options for downloading the file, etc.



The **indirect** link looks like this:

```
https://www.dropbox.com/s/5fjavy7agy24txy/Conv%20URL%20workflow.jpg?dl=0
```

Converting to a direct link for embedding is straightforward per this note - just replace the `dl=0` at the end of the ULR with `raw=1`. The resulting **direct** URL is:

```
https://www.dropbox.com/s/5fjavy7agy24txy/Conv%20URL%20workflow.jpg?raw=1
```

The URL is then shortened for convenience via BITLY:

```
http://bit.ly/2szJUlv
```

How it works...

The first step - using a file action to get the Dropbox link - relies on a workflow created by Uri Cohen. His code gets the link via the Dropbox API.

Before running the workflow for the first time, access to the API must be authorized. To do this, log into your Dropbox web account and then click this link to the **Dropbox API Explorer** page. Click the **Get Token** button and then copy the token from the **Access Token** field. Activate **Alfred** with the hot-key and type `md auth` and paste the token code into the command field.

Once authorized, select a file in your Dropbox folder and invoke the file action hot-key and select **Direct Dropbox URL**.

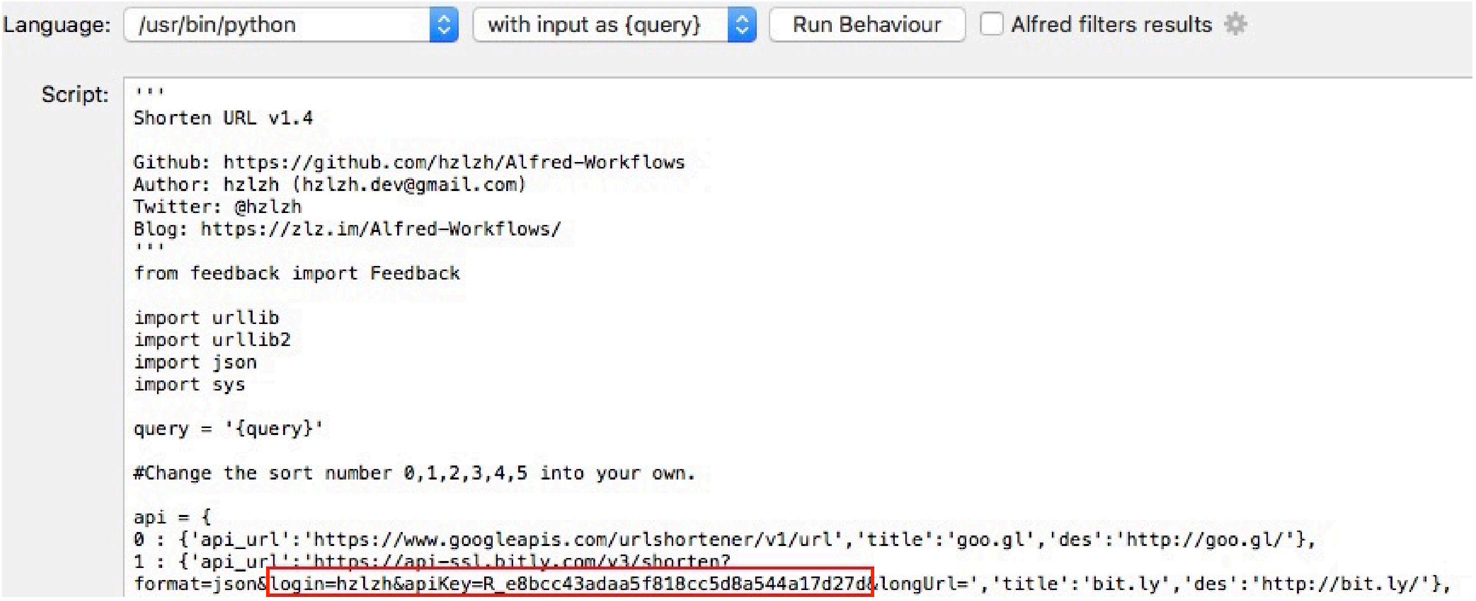
The Dropbox link is fed into a **Run Script** action with language: `/bin/bash`, input as `{query}` and instances run sequentially.

```
dburl=$(./get-db-link.sh {query})

apiKey="R_e8bcc43adaa5f818cc5d8a544a17d27d"
login="hzlzh"

curl -s --data "apiKey=$apiKey&login=$login&longUrl=$(echo $dburl | sed 's/dl=0/raw=1/g')&format=txt" http://api.bitly.com/v3/shorten | tr -d '\n'
```

The first line of the script invokes Uri Cohen's behind-the-scenes functions to get the Dropbox link of the file selected in the Finder.

```
dburl=$(./get-db-link.sh {query})
```

The second and third lines of the script are BITLY **apiKey** and **login** strings that were extracted from Shorten URL workflow by **hzlzh**. The **apiKey** and **login** values are found in the script filter code of that workflow.



```
'''
Shorten URL v1.4

Github: https://github.com/hzlzh/Alfred-Workflows
Author: hzlzh (hzlzh.dev@gmail.com)
Twitter: @hzlzh
Blog: https://zlz.im/Alfred-Workflows/
'''
from feedback import Feedback

import urllib
import urllib2
import json
import sys

query = '{query}'

#Change the sort number 0,1,2,3,4,5 into your own.

api = {
0 : {'api_url':'https://www.googleapis.com/urlshortener/v1/url','title':'goo.gl','des':'http://goo.gl/'},
1 : {'api_url':'https://api-ssl.bitly.com/v3/shorten?
format=json&login=hzlzh&apiKey=R_e8bcc43adaa5f818cc5d8a544a17d27d&longUrl=','title':'bit.ly','des':'http://bit.ly/'},
```

The last line in the bash script converts the indirect Dropbox link to a direct link and shortens the URL.

In the midst of that line, the `sed` function is applied to the indirect URL (`$dburl`) replacing `dl=0` with `raw=1`.

```
$(echo $dburl | sed 's/dl=0/raw=1/g')
```

The code around the `sed` operation was adapted from the Bit.ly link workflow by Chase Fegan.

The resulting direct, shortened URL is copied to the clipboard.