

# monkehTweets v1.2.4

Thank you for downloading monkehTweets, a ColdFusion Package developed to interact with the Twitter APIs.

monkehTweets v1.2.4 .....	1
License and Credits .....	2
Authors .....	2
Requirements .....	2
Installation .....	3
Getting Started .....	3
Figure 1 – The initial application page .....	4
Figure 2 – The required consumer details from the application page .....	4
Figure 3 – Obtaining the access details from your application page .....	5
Limitations .....	7
Remaining functionality .....	7
Testing .....	7

# License and Credits

Copyright 2010 Matt Gifford aka coldfumonkeh (<http://www.mattgifford.co.uk>)

Licensed under the Apache License, Version 2.0 (the "License");

You may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

For full License details, please read the LICENSE file available with this download.

## Authors

Developed by Matt Gifford AKA coldfumonkeh

<http://www.mattgifford.co.uk>

Got a lot out of this package? Saved you time and money?

Share the love and visit Matt's wishlist:

<http://www.amazon.co.uk/wishlist/B9PFNDZNH4PY>

## Requirements

monkehTweets requires ColdFusion 8+

# Installation

Unzip the package archive to the desired location (typically in the web root). After installation, you will see the following directories:

- Installation (contains this installation guide, no more, no less)
- com (contains all of the ColdFusion Components required for the application to interact with the Twitter API)

The application does not interact with any database, and as such needs no datasources.

## Getting Started

Version 1.2.4 of `monkehTweets` was released to include the OAuth authentication protocol, which now completely replaces the basic authentication method previously offered by Twitter.

To interact with the Twitter API, you will need to create an application with Twitter, which generates the consumer key and secret values that are required to instantiate and work with `monkehTweets` (and indeed the API itself).

Visit <http://dev.twitter.com/apps> to register a new application.

Items to note in here:

- Application type – set to browser
- Callback URL – you can set your initial callback URL here (e.g. <http://localhost:8500/authorize.cfm>) but you can overwrite this when making the actual authorization call from `monkehTweets`
- Access type – set to Read & Write to allow posts

Application Name:	<input type="text" value="monkehTweet"/>
Description:	<div><p>A twitter application using the monkehTweet ColdFusion API library</p></div>
Application Website:	<div><input type="text" value="http://127.0.0.1:8500/monkeh1"/><p>Where's your application's home page, where users can go to download or use it?</p></div>
Organization:	<input type="text" value="coldfumonkeh"/>
Application Type:	<div><div><input type="radio"/> Client <input checked="" type="radio"/> Browser</div><p>Does your application run in a Web Browser or a Desktop Client? Browser uses a Callback URL to return to your App after successful authentication. Client prompts your user to return to your application after approving access.</p></div>
Callback URL:	<div><input type="text" value="http://127.0.0.1:8500/monkeh1"/><p>Where should we return to after successfully authenticating? You can override this at any time by sending an <code>oauth_callback</code> while obtaining a request_token. You can <a href="#">authorize additional domains</a> if your app has more than one.</p></div>
Default Access type:	<div><div><input checked="" type="radio"/> Read &amp; Write <input type="radio"/> Read-only</div><p>What type of access does your application need? Note: @Anywhere applications</p></div>

**Figure 1 – The initial application page**

Having created your application, you need to make a note of the consumer details (the consumer key and consumer secret). These are required when instantiating the `monkehTweets` component.

### OAuth 1.0a Settings

OAuth 1.0a integrations require more work.

Consumer key

Consumer secret

**Figure 2 – The required consumer details from the application page**

If you are using the application for numerous users (a number of accounts using your application to interact with Twitter) then you're job here is done.

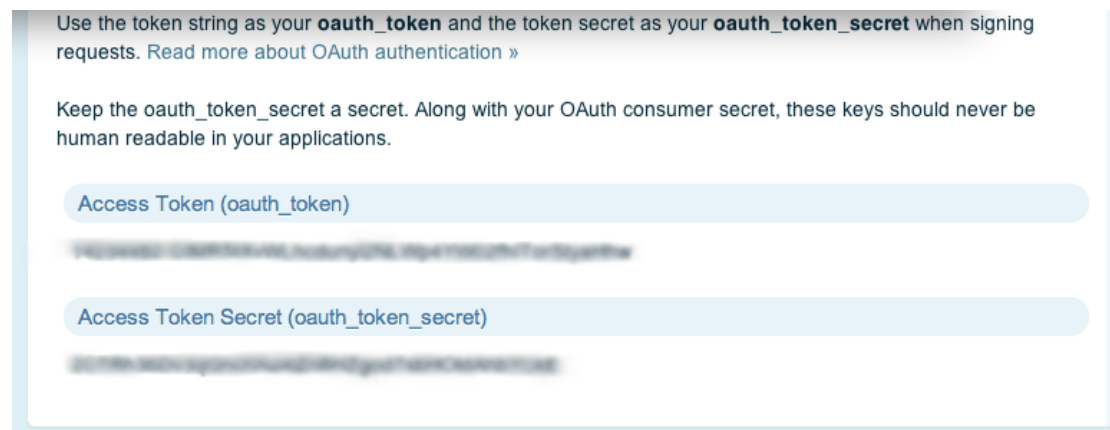
Simply instantiate the component with the consumer details and set up your code to request authentication and access for each user, as per the code samples within the monkehTweet download.

Instantiation is incredibly easy, as seen in the below example:

```
<cffunction name="OnApplicationStart"
    access="public"
    returntype="boolean"
    output="false">
    <cfscript>
    application.objMonkehTweet =
        createObject('component',
            'com.coldfumonkeh.monkehTweet')
        .init(
            consumerKey      = '< enter your consumer key >',
            consumerSecret   = '< enter your consumer secret >',
            parseResults     = true
        );
        return true;
    </cfscript>
</cffunction>
```

If you are intending to be the sole user (only your Twitter account will access the API) then you can bypass the requirement for authentication.

Within the application details page for your new application on <http://dev.twitter.com>, select the 'My Access Token' option from the right-hand menu.



**Figure 3 – Obtaining the access details from your application page**

Copy the access token and access token secret (the oauth tokens) from this screen, and use these along with your Twitter account screen name in the init() method when instantiating the monkehTweets component, as seen below:

```
<cffunction name="OnApplicationStart"
    access="public"
    returntype="boolean"
    output="false">
    <cfscript>
        application.objMonkehTweet =
            createObject('component',
                'com.coldfumonkeh.monkehTweet')
            .init(
                consumerKey      = '< enter your consumer key >',
                consumerSecret   = '< enter your consumer secret >',
                oauthToken       = '< enter the access token >',
                oauthTokenSecret = '< enter the access secret >',
                userAccountName   = '< enter your twitter screen name >',
                parseResults      = true
            );
        return true;
    </cfscript>
</cffunction>
```

This essentially bypasses the need for OAuth requests and redirects as we now have the access details for your account. You are ready to use monkehTweets instantly.

Lucky you. ☺

# Limitations

## Remaining functionality

There are a few methods currently missing from this release, namely:

getRewteets  
retweetedBy  
rewteetedByIDs  
geoSearch  
geoSimilarPlaces  
geoAddPlace  
addMemberToList  
deleteListMember  
getListMembersByID  
getListSubscriberByID

These will be revised and included in the next major release.

## Testing

monkehTweets has been tested on the Adobe ColdFusion platform.  
It has yet to be tested fully on Railo or OpenBlueDragon.