Getting the keys you need to use the TD API

There are 5 different values that take part in accessing the API

1. Your Account ID, which never changes, and is about 9 digits
2. Your API key, which never changes, and is about 30 alphanumeric uppercase characters
3. An 'auth' token, which you retrieve via a step to be described and is used to create item(4), and is about 900+ characters long, and looks like this: "/vA8JPzCMMdQOXqj4+5vj5vkK". You will have to log in every 90 days to get a new auth token.
4. A ''refresh' token that's good for 90 days and is used to create item (5) and is also 900+ characters long and looks like item (3).
5. An 'access token' that's good for 30 minutes, and is also 900+ characters and looks like item (3)

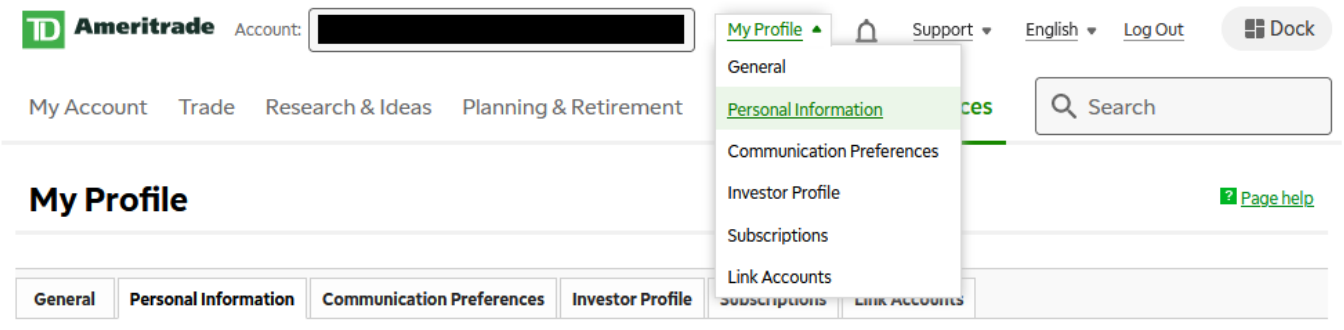
Getting your Account ID

To get your account id you go to the web site

`https://www.tdameritrade.com/`

And you log in to the account that you will be using to trade.

Go to My Profile/Personal information



The field 'Account ID' is on the personal information. Save that value.

Getting your API Key, aka ‘Consumer Key’

In order to get the API key you have to create an account on a different web site. Go to

`https://developer.tdameritrade.com/`

Click on ‘Register’ to create an account. This site has the concept of ‘apps’, which I interpret as being entry points into your account.

During the registration process you assign the account you want to an app, and an API key is created for you.

Now, it is not at all obvious at this point, but you *must create a callback url* to associate with the app, and it must be a url of the form http://localhost/something. This url is essential in getting to the next steps, even though its completely opaque as to why, and it must be localhost. This is confusing if you are developer because you will be thinking that there isn’t a web site on your system that is a call back location, but that’s ok.

So, you will leave the registration process with a brand new ‘Consumer key’ and having specified a callback url.

Getting an auth token via logging in

The next step is to retrieve an ‘auth’ token. The process is described pretty well here.

`https://developer.tdameritrade.com/content/simple-auth-local-apps`

You will need access to something to decode data in a url string. I did a google search for ‘decode url’ and found web pages that would do this online, you post data and decode it.

During this process you will get a pop up that will ask for your tdameritrade login credentials, and then it will confirm that you want to trade in this account via the API. You do, and then something interesting happens.

The page *redirects* to your callback url (remember it?). And, you get a 404 error because the http://localhost/whatever url isn’t on your system. But the important data is actually in the url bar. The url bar contains a code={url encoded auth token}

You copy the url encoded auth token, go to a web site to decode it, and THEN you are ready the next step, which is to create a *refresh* token. This will be described in the next step.

Unfortunately you will have to log in every 90 days to do this. The auth token, as best as I can tell, is good once to create a refresh token, and then it expires. However, you only need to make the call to get the refresh token once every 90 days. It’s a good idea to put the date into a calendar so you don’t forget, or your apps will start failing.

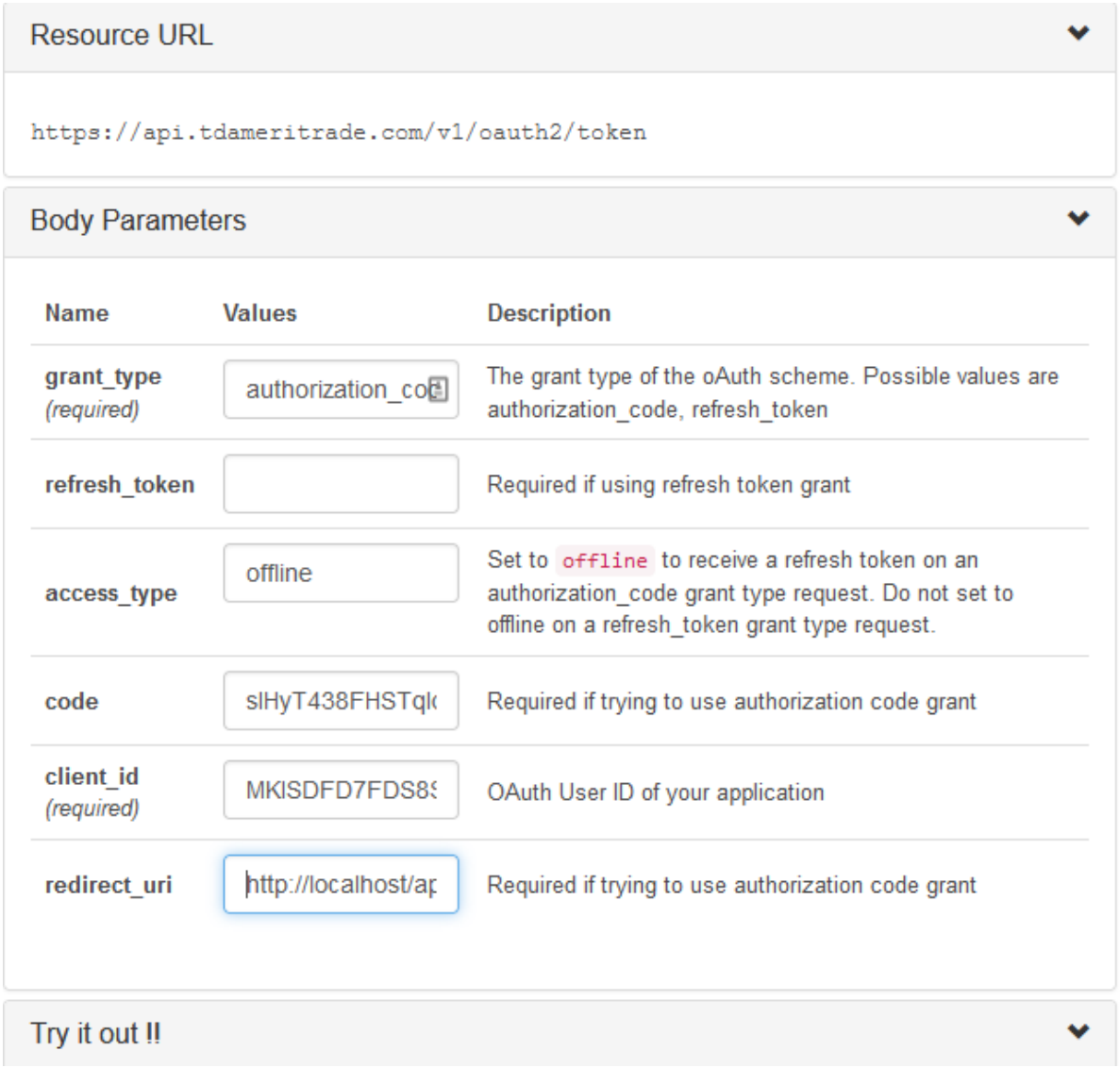
Its not at all clear why they don’t just give you a refesh token right away, but there you go. You have to create an auth token so you can create a refresh token, and then the auth token is no longer of any use.

Getting a refresh token from the auth token

The next step is to use your brand new auth token once and only once to retrieve an 'refresh' token. The refresh token is good for 90 days and you will save that value into a config file, as it will be used as part of an application. Go to this web site in the documentation

`https://developer.tdameritrade.com/authentication/apis/post/token-0`

There is a form that you will fill out, and SEND. In the 'code' field you use the url decoded value that you retrieved from the previous step. The 'client_id' is the same as the 'Consumer Key' that you got from the developer login, and the 'redirect_url' is also the same value you specified in the developer login, even though in this case its not going to do a redirect. Note that the values are truncated here, but the full values are used. 'authorization_code' is spelled out, for instance.



Getting a refresh token from the auth token

WARNING. This call is very particular, and I get lots of ‘bad request’ returns even when I think I’ve done everything right. The auth token is a beast of a string and making sure its pasted correctly is harder than it should be. The callback url should also be spelled the same, just to be sure.

You may need to retry multiple times, logging in again to get a new auth token and decoding it, before it works.

If it fails, go get a new auth token and review the spelling of each item.

If this works you get a nice green 200 OK, and a json body that includes a refresh_token AND an access_token.

Ignore the access_token, the library can fetch those. Copy that refresh token and save it in appkeys.json in the TD Lib build.

Getting an access token from the refresh token

Congratulations, you made it!

The access token is a short lived (30 minute) token that goes into the header of API calls.

Getting the access token is easy, there is a call in the library called GetAccessToken() which will return an access token.

You can examine the code in GetAccessToken() and see that its using the same url and form that we used to generate a refresh token, and in fact that same form generates both refresh tokens and access tokens, but access tokens are something that a running program uses so we can pull those as needed.

Why isn't there a library call to get a refresh token? Because you need an auth token to generate a refresh token, and the auth token is only generated via the login, and an auth token only works once.

The appkeys.json file

In order to place API calls a running program requires 4 things.

1. The account id, the 9 digit value you retrieved from logging in to tdameritrade.com
2. The 'Consumer Key', the 30 or so length alphanumeric string you retrieved from logging in to developer.tdameritrade.com
3. The refresh token, which was created from an auth token, and is used to generate access tokens
4. An access token, which the library will create for you and goes into the header of the API call

Since (4), the access token, is generated we only need to store values 1,2, and 3.

In the library source there is a file called appkeys.json, and this file is where you store values 1,2, and 3. When you've plugged in your account values it will look something like this:

```
{
  "ConsumerKey": "GO4KDIOEPROGJKI7IDHDJGKKKDJ3",
  "AccountNumber": "0496746365",
  "refresh_token": "900+ characters that look like qBD/gkjNP9Exvifft3+xQLMFfPelHI8CBD1TYlyKcfzvNDK"
}
```

The library class TDConnection loads and desterilizes this file when its created. You only need one of these objects for your application.

Every app will do these two steps at a minimum. Once you have the authtoken you will be able to access the API.

```
/*
 * Get a TD connection and authtoken for this operation
 */

TDConnection oTDA = new TDConnection();
string authtoken = oTDA.GetAccessToken();
```