Sending Auth tokens to Sigma AppScript

Intended for people who use develops Sigma Apps, and need to authenticate with e.g. OAuth2.

# Background

If Sigma need to authenticate with a server using e.g. OAuth2, it may be necessary to launch a web browser for the user to sign in. After the sign in, the web browser will direct with a token to a callback site, defined by the app developer. However, the token need to be transferred back to Sigma. This is similar to mobile Apps, that need to authenticate.

For this purpose, Sigma (from version 7), installs itself in Windows with a protocol registration, that allows websites to pass information directly to Sigma.

# Step-by-step

## Direct the user to a website

This is possible, using the AppScript unit called “System.Help”, and the HelpService class, with the method DisplayURL(string, string).

## Implement a callback server

The next thing you need is a callback server, that you control. Configure the app on the 3rd party OAuth server, to use your server for callback. Depending on the authentication method, you can use your server to handle things such as token exchange (if you receive a code that must be exchanged), token refresh, etc. Once you have the token for the Sigma App, redirect it to:

SigmaAuthToken://[your\_data]

An example of [your\_data] could be:

SigmaAuthToken://?SigmaApp=MySigmaAppName&token=xyz

The user will experience the browser (Chrome, IE, Edge, …) to ask the user if it allows opening Sigma. If the user accepts, the entire URL is passed to Sigma, and can be retrieved in AppScript.

## AppScript implementation

The following code is a minimal example of receiving the token, by listening for the notification of new AuthTokens. Notice, that more Sigma Apps could be subscribing to the Auth event, which is why we check that it was intended for our App.

**unit TestAuthToken;**

**interface**

**implementation**

**uses**

**System.Net.HTTP.Client,**

**Sigma.Document;**

**procedure NewAuth(App: TSigmaApplication; NotificationType: TSigmaApplicationNotification);**

**begin**

**if NotificationType = anAuthToken then**

**begin**

**var url := TUri.Create(Application.AuthTokenData);**

**if url.Params <> '' then**

**begin**

**var params := TStringList.Create;**

**params.Delimiter := '&';**

**params.DelimitedText := url.Params;**

**if (params.Values['SigmaApp'] = 'MySigmaAppName') then**

**print( params.Values['token'] ); //Should be stored and used...**

**params.Free;**

**end;**

**end;**

**end;**

**initialization**

**Application.OnNotify := NewAuth;**

**finalization**

**Application.OnNotify := nil;**