

1. I created a private repository where I will read the Qlik Sense scripts:

The screenshot shows a GitHub repository page for 'ucianotk / qlik-sense'. The repository is marked as 'Private'. The file 'Test Script.qvs' is selected, showing its content. The file contains 29 lines of Qlik Sense script code, including various SET statements for formatting and a loop for testing.

```
1 SET ThousandSep='.';
2 SET DecimalSep=',';
3 SET MoneyThousandSep='.';
4 SET MoneyDecimalSep='.';
5 SET MoneyFormat='RS###,00;-RS###,00';
6 SET TimeFormat='h:mm:ss';
7 SET DateFormat='DD/MM/YYYY';
8 SET TimestampFormat='DD/MM/YYYY hh:mm:ss[.fff]';
9 SET FirstWeekDay=6;
10 SET BrokenWeeks=1;
11 SET ReferenceDay=0;
12 SET FirstMonthOfYear=1;
13 SET CollationLocale='pt-BR';
14 SET CreateSearchIndexOnReload=1;
15 SET MonthNames='jan;fev;mar;abr;mai;jun;jul;ago;set;out;nov;dez';
16 SET LongMonthNames='janeiro;fevereiro;março;abril;maio;junho;julho;agosto;setembro;outubro;novembro;dezembro';
17 SET DayNames='seg;ter;qua;qui;sex;sáb;dom';
18 SET LongDayNames='segunda-feira;terça-feira;quarta-feira;quinta-feira;sexta-feira;sábado;domingo';
19 SET NumericalAbbreviation='3;k;6;M;9;G;12;T;15;P;18;E;21;Z;24;Y;-3;m;-6;µ;-9;n;-12;p;-15;f;-18;a;-21;z;-24;y';
20
21 FOR vTeste = 0 TO 100
22   TRACE NÚMERO: $(vTeste);
23 NEXT;
24
25 EXIT SCRIPT;
26
27 LOAD
28   *
29 FROM TESTE;
```

2. Then I created a token to be able to authenticate and read my private repository (I just checked the option below):

The screenshot shows the GitHub Developer settings page, specifically the 'New personal access token' section. The 'Personal access tokens' option is selected in the left sidebar. The 'repo' scope is checked, indicating that the token will have full control of private repositories.

Settings / Developer settings

- GitHub Apps
- OAuth Apps
- Personal access tokens**

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

Qlik Sense Example

What's this token for?

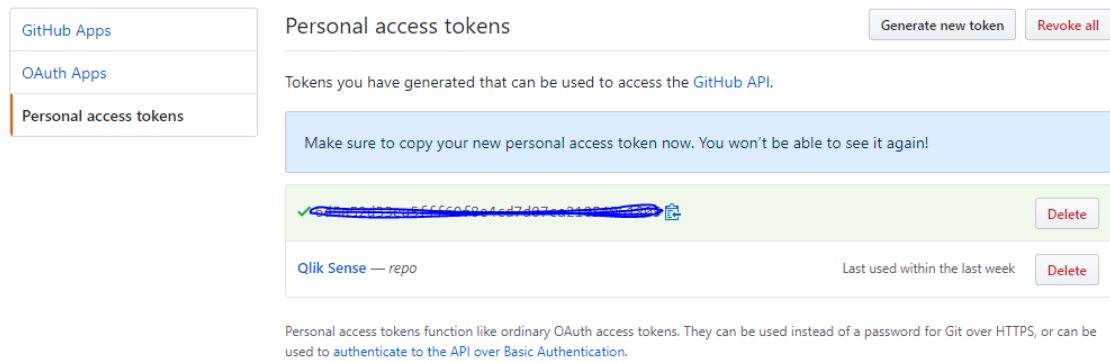
Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input type="checkbox"/> repo:status	Access commit status
<input type="checkbox"/> repo_deployment	Access deployment status
<input type="checkbox"/> public_repo	Access public repositories
<input type="checkbox"/> repo:invite	Access repository invitations
<input type="checkbox"/> write:packages	Upload packages to github package registry

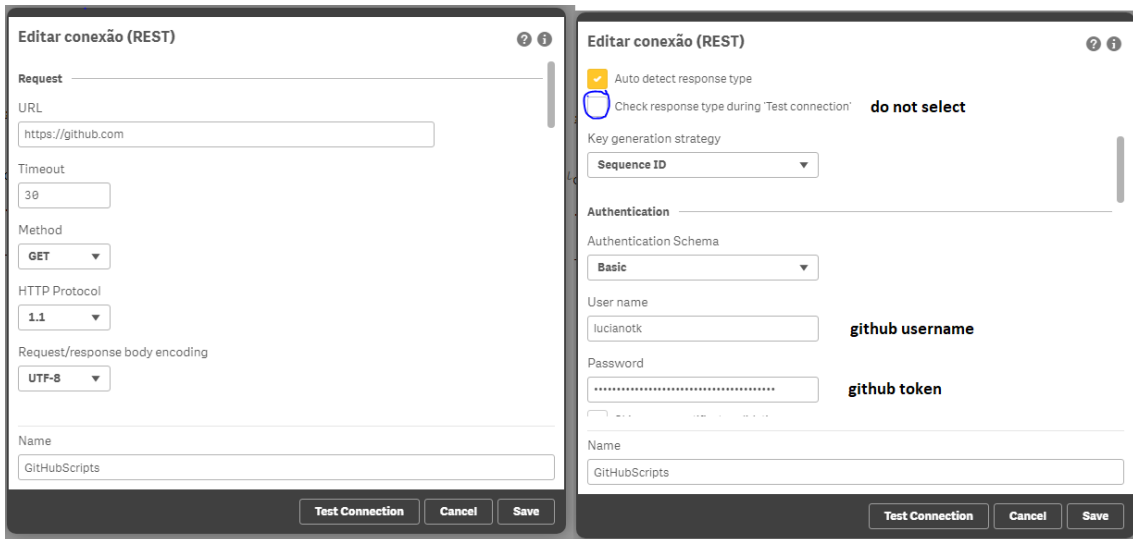
3. I copied the generated token and saved it somewhere safe:

Settings / Developer settings



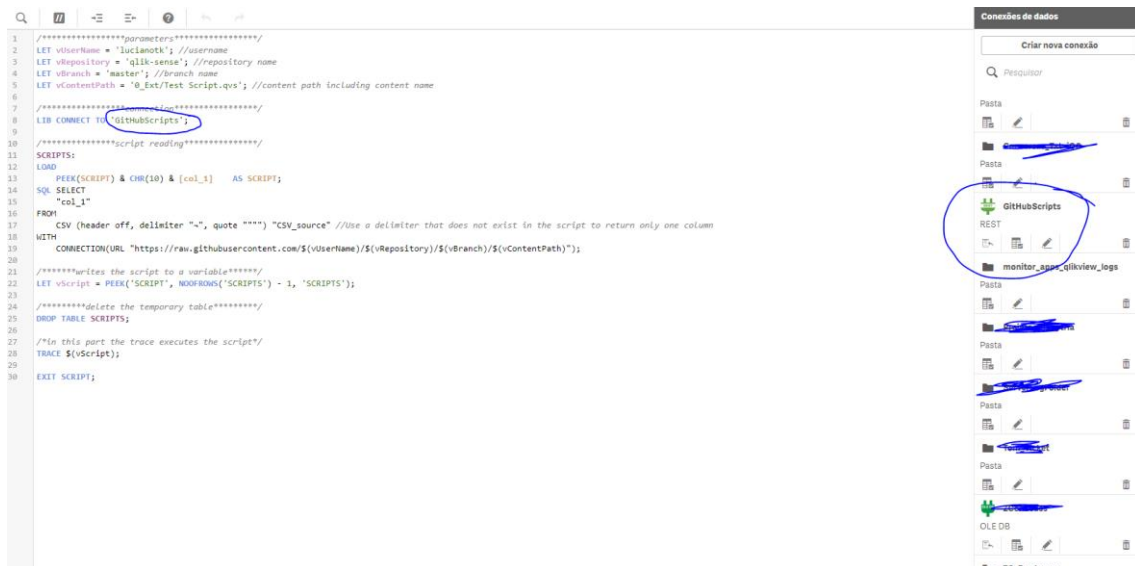
The screenshot shows the 'Personal access tokens' page in GitHub settings. On the left, there are navigation links for 'GitHub Apps', 'OAuth Apps', and 'Personal access tokens'. The main area has a 'Generate new token' button and a 'Revoke all' button. Below, a blue box contains the instruction: 'Make sure to copy your new personal access token now. You won't be able to see it again!'. A green box displays a token: 'ghp_1234567890abcdefghijklmnopqrstuvwxyz' with a copy icon. Below the token, it says 'Qlik Sense — repo' and 'Last used within the last week'. At the bottom, there is a note: 'Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.'

4. I created a new REST connection with the address 'https://github.com' and my user and created token (this connection is only for authenticating):



The image shows two side-by-side screenshots of the 'Editar conexão (REST)' dialog box. The left screenshot shows the 'Request' tab with fields for 'URL' (https://github.com), 'Method' (GET), 'HTTP Protocol' (1.1), and 'Request/response body encoding' (UTF-8). The right screenshot shows the 'Authentication' tab with 'Authentication Schema' set to 'Basic', 'User name' as 'luciano.tk', and 'Password' as 'github token'. It also has 'Key generation strategy' set to 'Sequence ID' and 'Name' as 'GitHubScripts'. Both screenshots have 'Test Connection', 'Cancel', and 'Save' buttons at the bottom.

5. I adapted the load to search the files according to parameters defined in variables



The image shows a script editor on the left and a 'Conexões de dados' (Data Connections) panel on the right. The script is a Kettle script with the following content:

```
1 //*****parameters*****
2 LET vusername = 'luciano.tk'; //username
3 LET vRepository = 'qlik-sense'; //repository name
4 LET vBranch = 'master'; //branch name
5 LET vContentPath = '0_Ext/Test_Script.qvs'; //content path including content name
6
7 //*****script reading*****
8 LIB CONNECT TO 'GitHubScripts';
9
10 //*****script reading*****
11 SCRIPTS:
12 LOAD
13 PEEK(SCRIPT) & CHR(10) & [col_1] AS SCRIPT;
14 SQL SELECT
15 "col_1"
16 FROM
17 CSV (header off, delimiter "~", quote "") "CSV_source" //Use a delimiter that does not exist in the script to return only one column
18 WITH
19 CONNECTION(URL "https://raw.githubusercontent.com/${vusername}/${vRepository}/${vBranch}/${vContentPath}");
20
21 //*****writes the script to a variable*****
22 LET vscript = PEEK("SCRIPT", NOFORN("SCRIPTS") - 1, "SCRIPTS");
23
24 //*****delete the temporary table*****
25 DROP TABLE SCRIPTS;
26
27 /*In this part the trace executes the script*/
28 TRACE ${vscript};
29
30 EXIT SCRIPT;
```

The right panel shows a list of data connections, with 'GitHubScripts' highlighted and circled in blue. The list includes 'Criar nova conexão', 'Pesquisar', and several 'Pasta' entries, with 'GitHubScripts' being the selected connection.

```
1 /*****parameters*****/
2 LET vUserName = 'luciano@k'; //username
3 LET vRepository = 'qlik-sense'; //repository name
4 LET vBranch = 'master'; //branch name
5 LET vContentPath = '0_Ext/Test Script.qvs'; //cont
6
7 /*****connection*****/
8 LIB CONNECT TO 'GitHubScripts';
9
10 /*****script reading*****/
11 SCRIPTS:
12 LOAD
13 PEEK(SCRIPT) & CHR(10) & [col_1] AS SCRIPT;
14 SQL SELECT
15 "col_1"
16 FROM
17 CSV (header off, delimiter "-", quote "") C
18 WITH
19 CONNECTION(URL "https://raw.githubusercontent.com
20
21 /*****writes the script to a variable*****/
22 LET vScript = PEEK('SCRIPT', NOFRONS('SCRIPTS')
23
24 /*****delete the temporary table*****/
25 DROP TABLE SCRIPTS;
26
27 /*in this part the trace executes the script*/
28 TRACE $(vScript);
29
30 EXIT SCRIPT;
```

Progresso da carga de dados

Carga de dados concluída.

Tempo decorrido 00:00:04

NÚMERO: 93
NÚMERO: 94
NÚMERO: 95
NÚMERO: 96
NÚMERO: 97
NÚMERO: 98
NÚMERO: 99
NÚMERO: 100

Criando índice de pesquisa
Criação do índice de pesquisa concluída com sucesso

Aplicativo salvo

Concluído com êxito
0 erro(s) corrigido(s)
0 chave(s) sintética(s)

Fechar quando concluído com êxito Fechar

Todas as alterações salvas

Code:

```
/******parameters******/
LET vUserName = 'lucianotk'; //username
LET vRepository = 'qlik-sense'; //repository name
LET vBranch = 'master'; //branch name
LET vContentPath = '0_Ext/Test Script.qvs'; //content path including content name

/******connection******/
LIB CONNECT TO 'GitHubScripts'; // replace with the name of your connection created in the
previous step, if necessary

/******script reading******/
SCRIPTS:
LOAD
    PEEK(SCRIPT) & CHR(10) & [col_1]    AS SCRIPT;
SQL SELECT
    "col_1"
FROM
    CSV (header off, delimiter "-", quote "") "CSV_source" //Use a delimiter that does
not exist in the script to return only one column
WITH
    CONNECTION(URL
"https://raw.githubusercontent.com/$(vUserName)/$(vRepository)/$(vBranch)/$(vContentPat
h)");

/******writes the script to a variable******/
LET vScript = PEEK('SCRIPT', NOOFROWS('SCRIPTS') - 1, 'SCRIPTS');

/******delete the temporary table******/
DROP TABLE SCRIPTS;

/*in this part the trace executes the script*/
TRACE $(vScript);
```

Note: It worked for me and met my need, but I don't know if it solves yours, it's just an example.