Using SAS with Microsoft 365
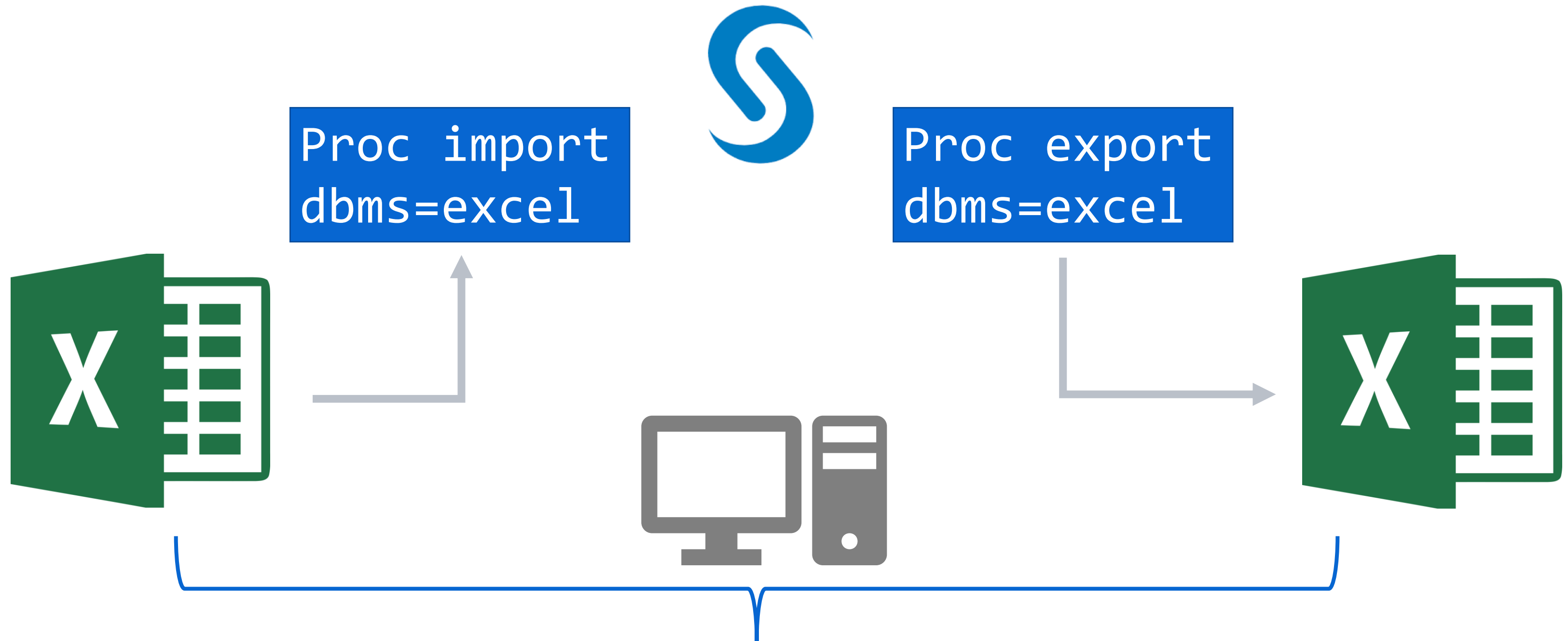
A Programming Approach

Chris Hemedinger, Director of SAS User Engagement
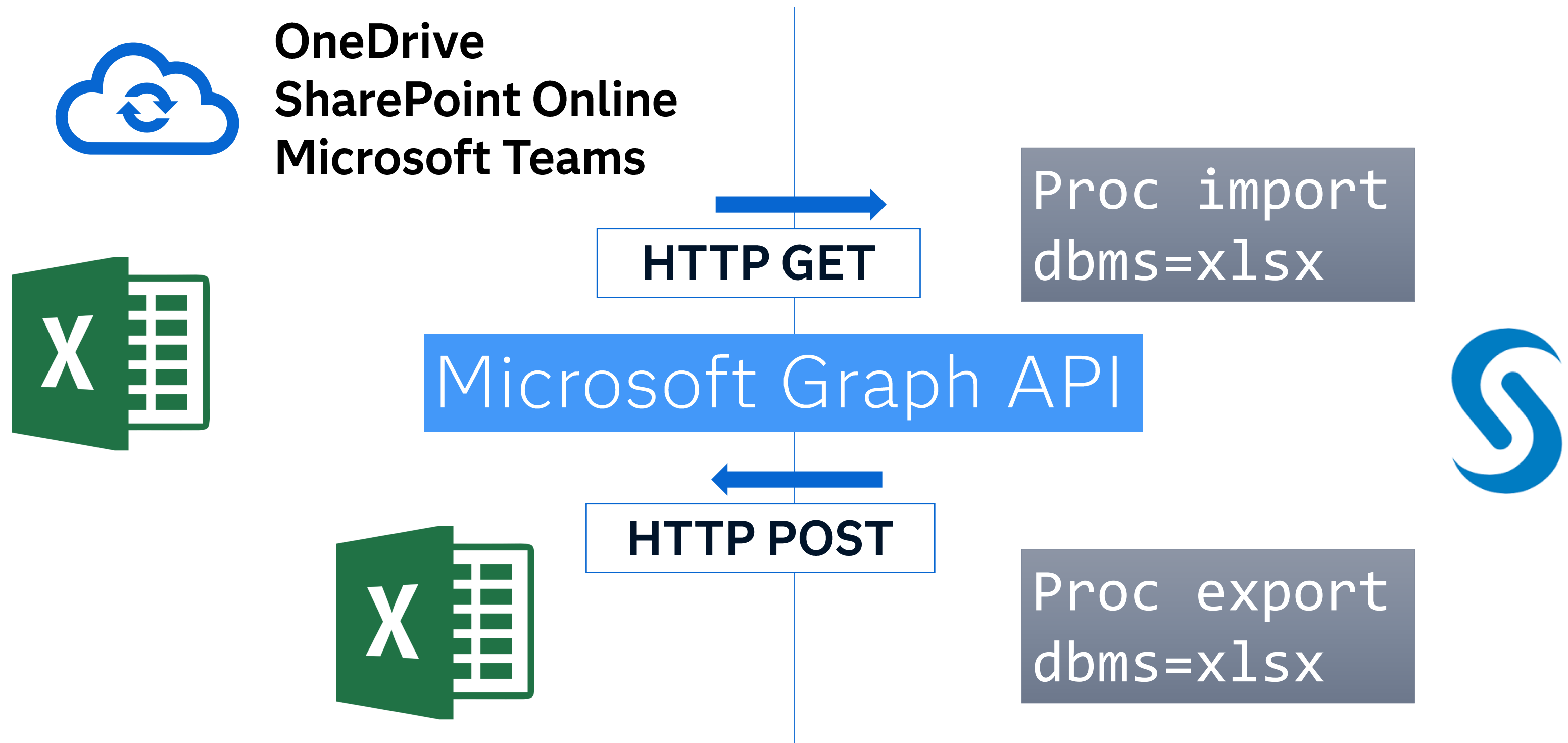
§sas

SAS and Microsoft Office

Simpler times, pre-Cloud

Proc import
dbms=excel

Proc export
dbms=excel

SAS and Microsoft 365
More collaboration, more moving parts

OneDrive
SharePoint Online
Microsoft Teams

HTTP GET

Microsoft Graph API

HTTP POST

Proc import
dbms=xlsx

Proc export
dbms=xlsx

More than Excel: Microsoft Graph APIs

Use the documentation and the Microsoft Graph Explorer tool

PROC HTTP and the JSON Libname engine

Your keys to using the Microsoft APIs from within SAS

```sas
filename resp temp;
proc http
url="https://graph.microsoft.com/v1.0/me"
  oauth_bearer="&access_token"
  out=resp;
run;


libname me JSON fileref=resp;
proc print data=me.root;
run;
```

Obs	ordinal_root	_odata_context	displayName	givenName	jobTitle	mail	mobilePhone	officeLocation	p
1	1	https://graph.microsoft.com/v1.0/ $metadata#users/$entity	Chris Hemedinger	Chris	Director, SAS User Engagement			U5132	

§sas

Connecting SAS to Microsoft 365

One time

| Register a new client "application" | → | Obtain authorization code |

Each time, automated
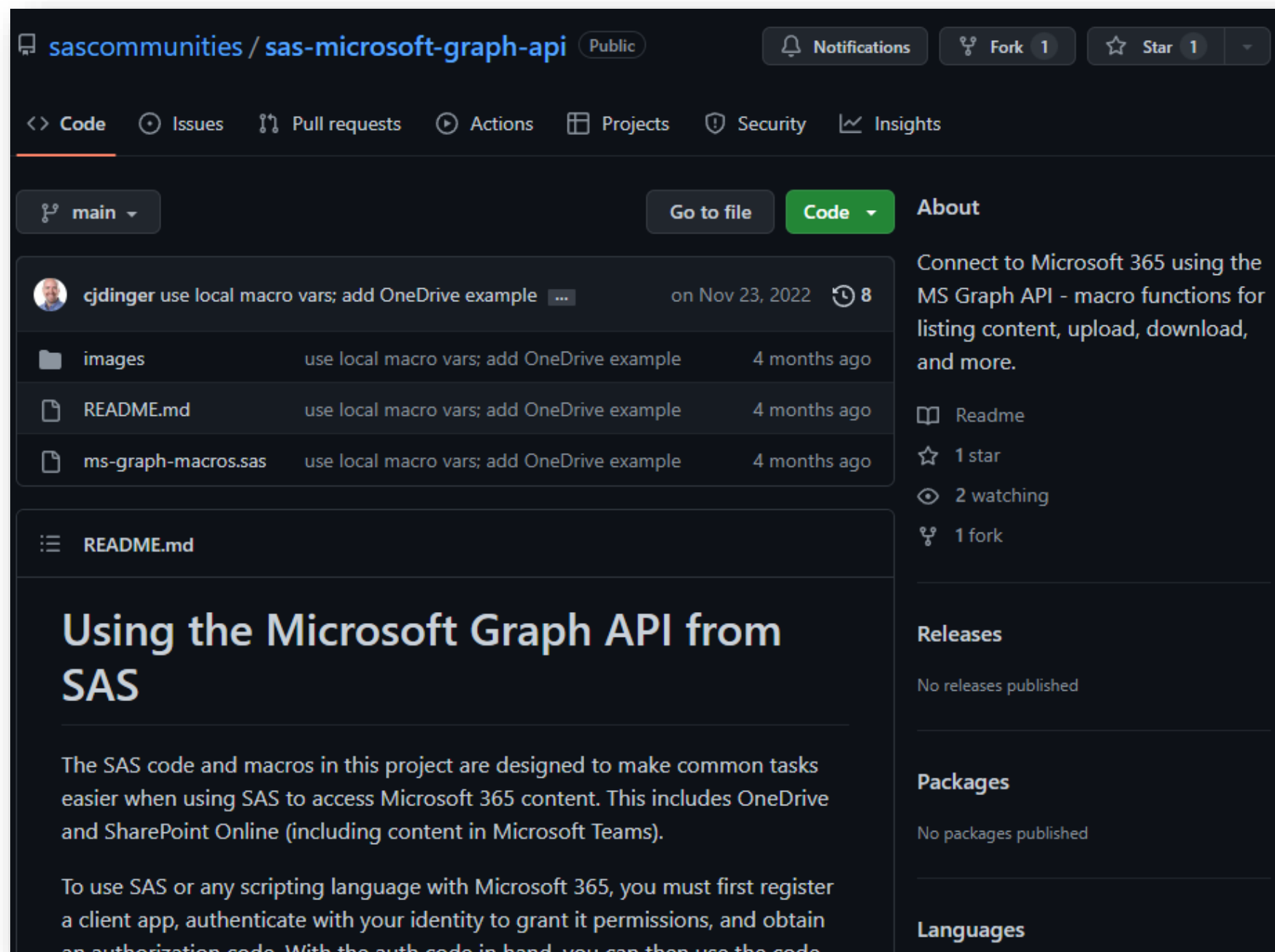
| Obtain access token (and refresh token) | → | Use Microsoft 365 APIs to navigate, retrieve, and publish files |

§sas

SAS macros to help

Find them at https://github.com/sascommunities/sas-microsoft-graph-api

- Help with auth code
- Obtain access token to connect
- List OneDrive root, folders, files
- List SharePoint site folders, files
- Download files
- Upload files (including large files)

1. Register a client app

1. Register a client app (New reg)

1. Register a client app (Name and account type)

1. Register a client app (Properties)

1. Register a client app (redirect URI)

Platform configurations

Depending on the platform or device this application is targeting, additional configuration may be required such as redirect URIs, specific authentication settings, or fields specific to the platform.

+ Add a platform

Mobile and desktop applications   Quickstart   Docs↗   🗑

Redirect URIs

The URIs we will accept as destinations when returning authentication responses (tokens) after successfully authenticating users. The redirect URI you send in the request to the login server should match one listed here. Also referred to as reply URLs. Learn more about Redirect URIs and their restrictions↗

✓ https://login.microsoftonline.com/common/oauth2/nativeclient 📄

☐ https://login.live.com/oauth20_desktop.srf (LiveSDK) 📄

☐ msala73c4ef7-ad2e-47d9-8d42-46b0217846cf://auth (MSAL only) 📄

Add URI

1. Register a client app (Permissions)

Configured permissions

Applications are authorized to call APIs when they are granted permissions by users/admins as part of the consent process. The list of configured permissions should include all the permissions the application needs. Learn more about permissions and consent

+ Add a permission   ✓ Grant admin consent for SAS

API / Permissions name	Type	Description	Admin consent req...	Status	
∨ Microsoft Graph (3)					•••
Files.ReadWrite.All	Delegated	Have full access to all files user can access	No		•••
Sites.ReadWrite.All	Delegated	Edit or delete items in all site collections	No		•••
User.Read	Delegated	Sign in and read user profile	No		•••

To view and manage consented permissions for individual apps, as well as your tenant's consent settings, try Enterprise applications.

NOTE: Your organization might require Admin consent to register an app and its permissions!

§sas

Store your App ID, Tenant ID in a config file


```json
{
    "tenant_id": "7ee95434-72a2-4469-891e-aa584912124e",
    "client_id": "c777c2bc-5d9b-4955-806d-a9d26755c65e",
    "redirect_uri": "https://login.microsoftonline.com/common/oauth2/nativeclient",
    "resource" : "https://graph.microsoft.com"
}
```

2. Obtain an auth code (auth URL)

```sas
%let codepath = C:\Projects\GitHub\sas-microsoft-graph-api;
%include "&codepath./ms-graph-macros.sas";


%let cfg = u:\Projects\ms365;
%initConfig(configPath=&cfg);



%generateAuthUrl();
```

```
642          %generateAuthUrl();
Paste this URL into your web browser:
-- START -------
https://login.microsoft.com/b1c14d5c-3625-45b3-a430-9552373a0c2f/oauth2/authorize?client
om
---END ---------
```

§sas

2. Obtain an auth code (grant permission, copy code)
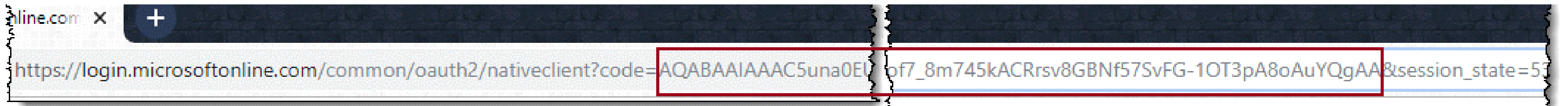
2. Obtain an auth code (generate token.json)

```sas
/* Note: this code can be quite long -- 700+ characters. */
%let auth_code=PASTE-YOUR-AUTH-CODE-HERE;

/*
  Now that we have an authorization code we can get the access token
  This step will write the token.json file that we can use in our
  production programs.
*/
%get_access_token(&auth_code.);
```

§sas

Managing your credentials
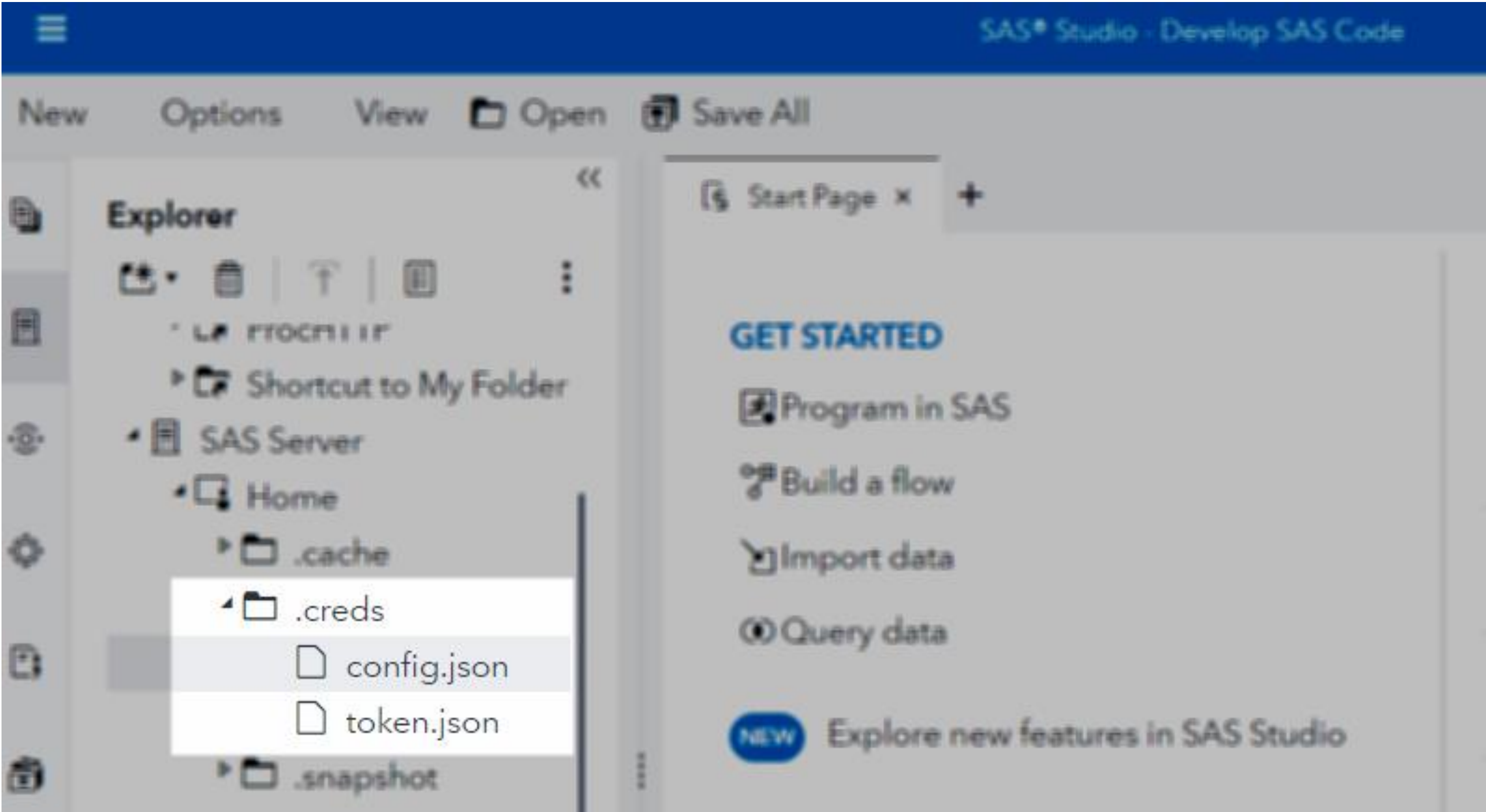
Use helper macros for common tasks

All within ms-graph-macros.sas

List OneDrive contents	%listMyDrives %listFolderItems
List SharePoint folders	%listSiteLibraries %listFolderItems
Upload a file from SAS to SharePoint	%uploadFile
Download a file from SharePoint to SAS	%downloadFile

§sas

DEMO

§.sas.
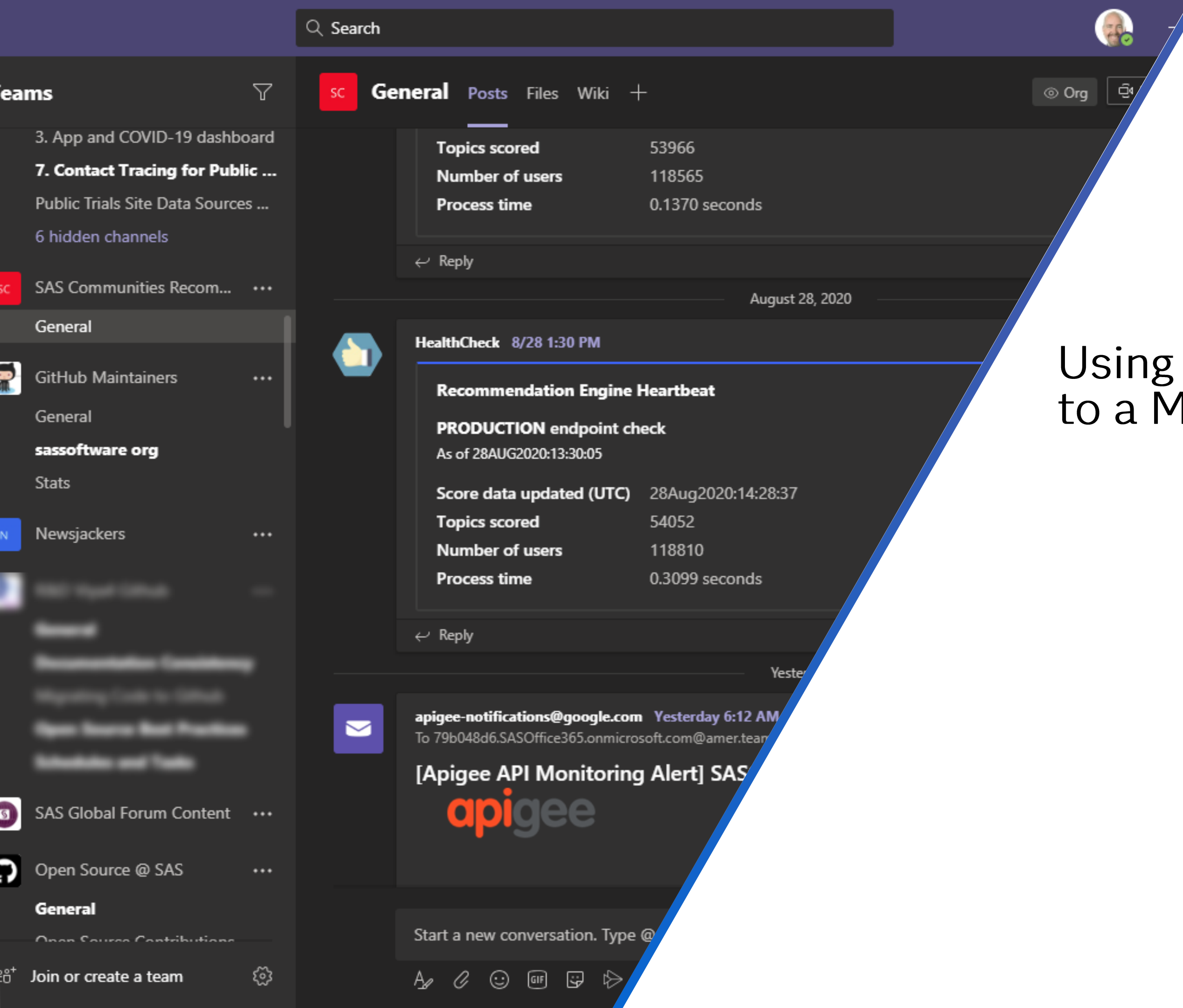
Using SAS to Send Messages to a Microsoft Teams Channel

Two methods: Email and Incoming Webhook
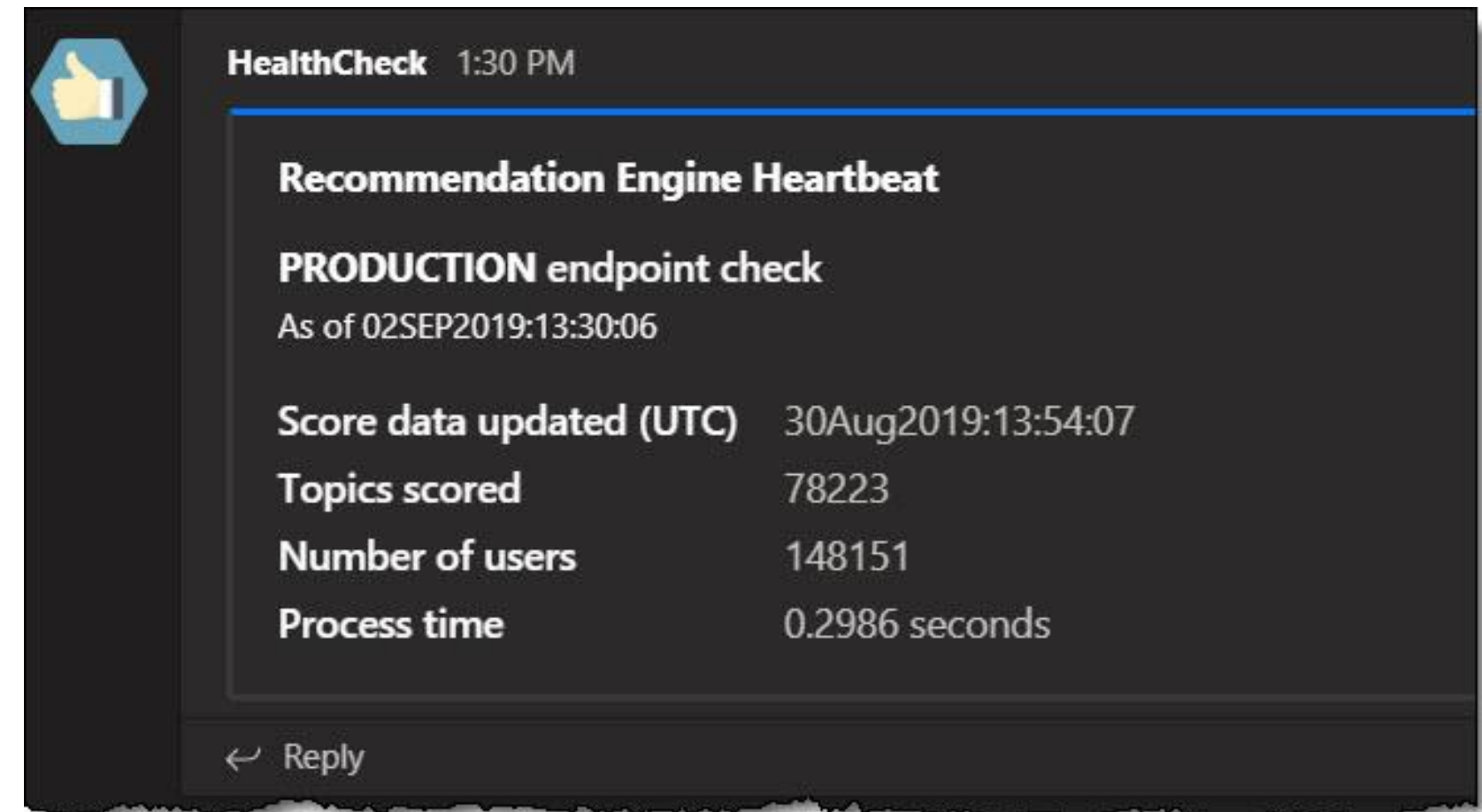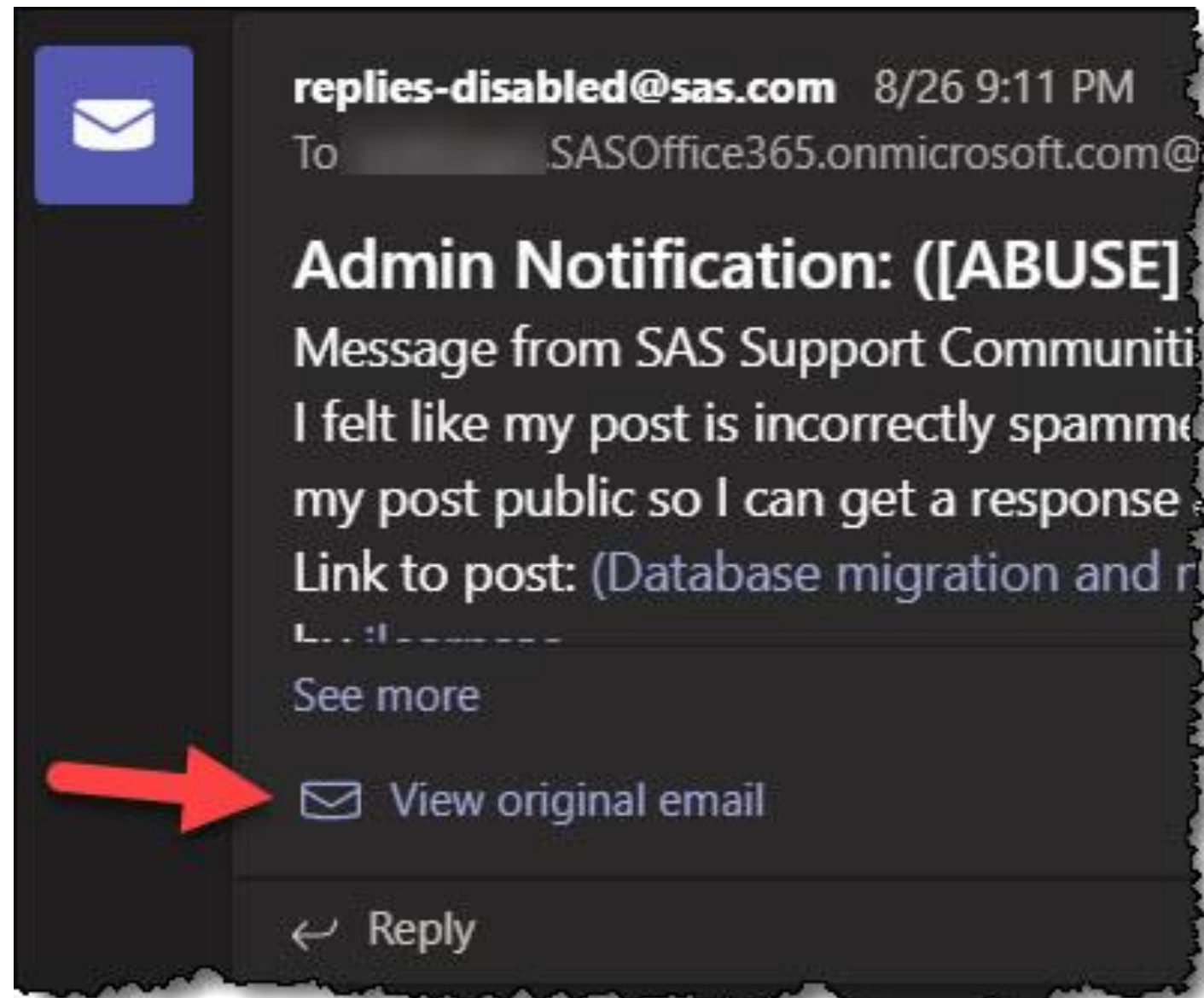
E-mail to a channel from SAS
Using FILENAME EMAIL
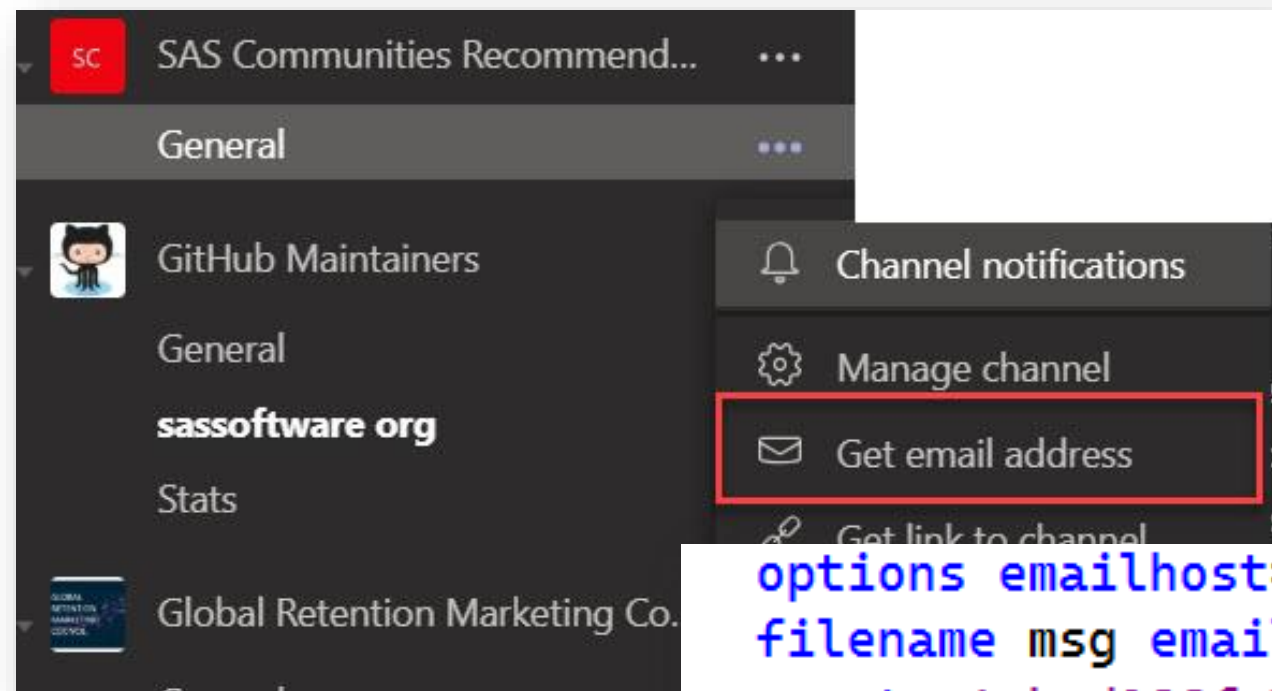

```
options emailhost='mailhost.company.com' emailsys=smtp;
filename msg email
    to='abcd123f.TenantOffice365.onmicrosoft.com@amer.teams.ms'
    FROM = "Communities Bot <communities@sas.com>"
    subject="&group. members: &toaddcount. INVITED on &SYSDATE."
    content_type='text/html'
;
ods html(id=email)
 file=msg (title="&group. report")
 style=dove;

   title "The following new users were INVITED to the &group. Group (&SYSDATE.)";
   proc print data=candidates;
   run;
ods html (id=email) close;
```

Incoming Webhook for rich messages
Using PROC HTTP
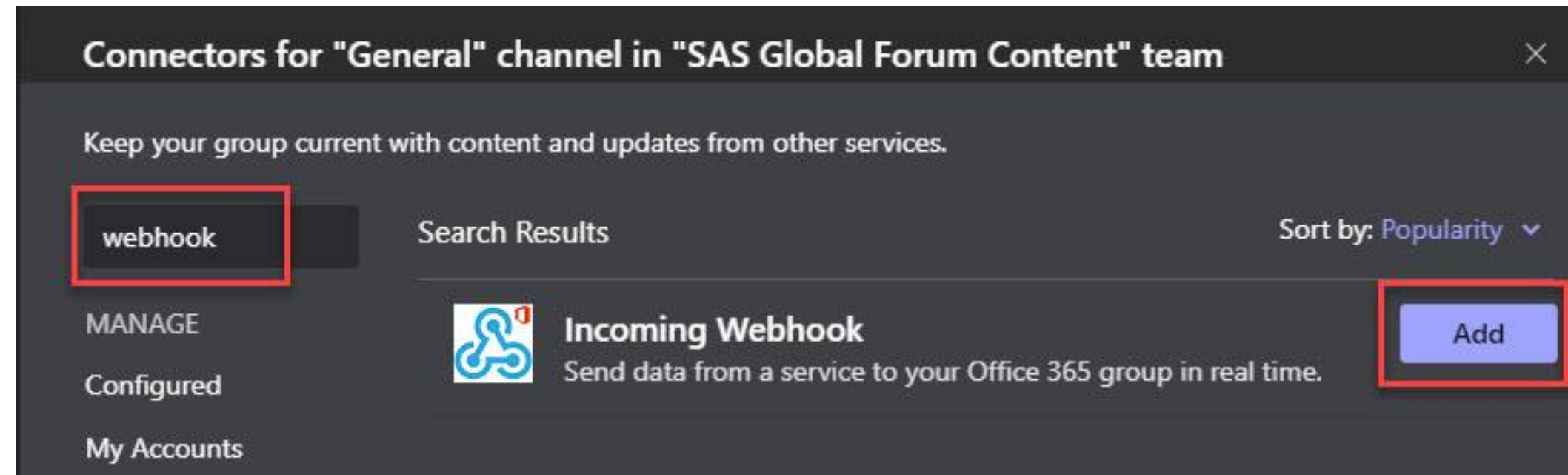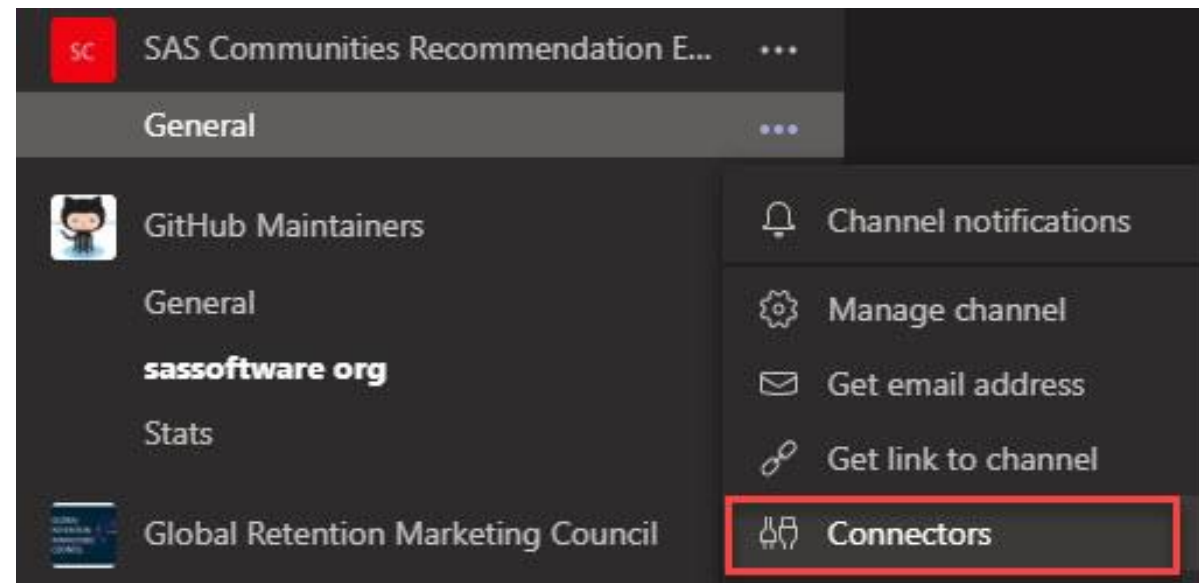
Sending a simple message

```
filename resp temp;
options noquotelenmax;
proc http
    /* Substitute your webhook URL here */
    url="https://outlook.office.com/webhook/your-unique-webhook-address-it-is-very-long"
    method="POST"
    in=
    '{
        "$schema": "http://adaptivecards.io/schemas/adaptive-card.json",
        "type": "AdaptiveCard",
        "version": "1.0",
        "summary": "Test message from SAS",
        "text": "This message was sent by **SAS**!"
    }'
    out=resp;
run;
```


SAS Updates 11:28 AM

This message was sent by **SAS!**

↵ Reply
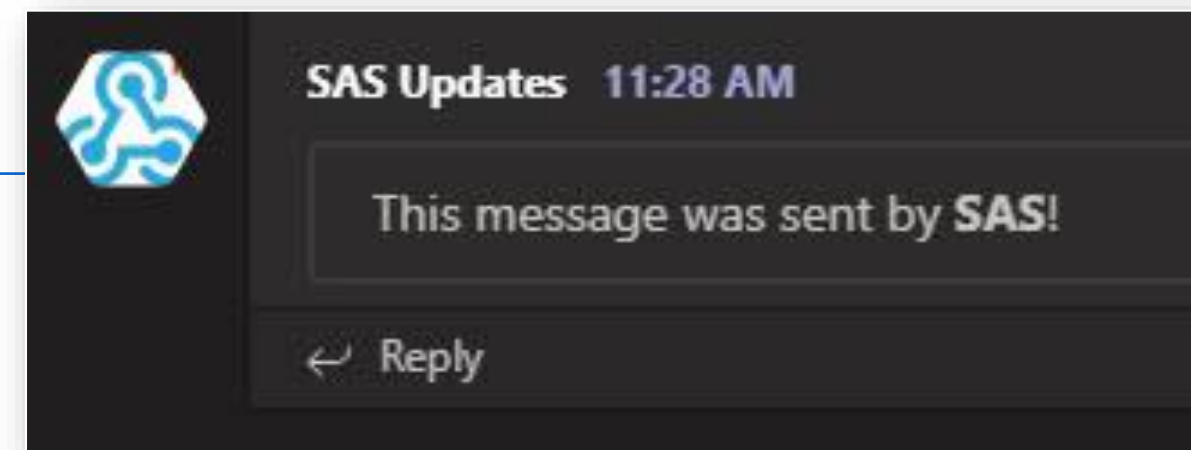
Sending a more complex message

Design a Message Card


```json
{
    "@type": "MessageCard",
    "@context": "https://schema.org/extensions",
    "summary": "Recommendation Engine Health Check",
    "themeColor": "0075FF",
    "sections": [
        {
            "startGroup": true,
            "title": "**Recommendation Engine Heartbeat**",
            "activityImage": "",
            "activityTitle": "**PRODUCTION** endpoint check",
            "activitySubtitle": "As of 05SEP2019:13:10:30",
            "facts": [
                {
                    "name": "Score data updated (UTC)",
                    "value": "05Sep2019:13:30:06"
                },
                {
                    "name": "Topics scored",
                    "value": "78378"
                },
                {
                    "name": "Number of users",
                    "value": "148686"
                },
                {
                    "name": "Process time",
                    "value": "2.0373 seconds"
                }
            ]
        }
    ]
}
```


HealthCheck 1:30 PM

Recommendation Engine Heartbeat

PRODUCTION endpoint check
As of 02SEP2019:13:30:06

Score data updated (UTC)	30Aug2019:13:54:07
Topics scored	78223
Number of users	148151
Process time	0.2986 seconds

↵ Reply
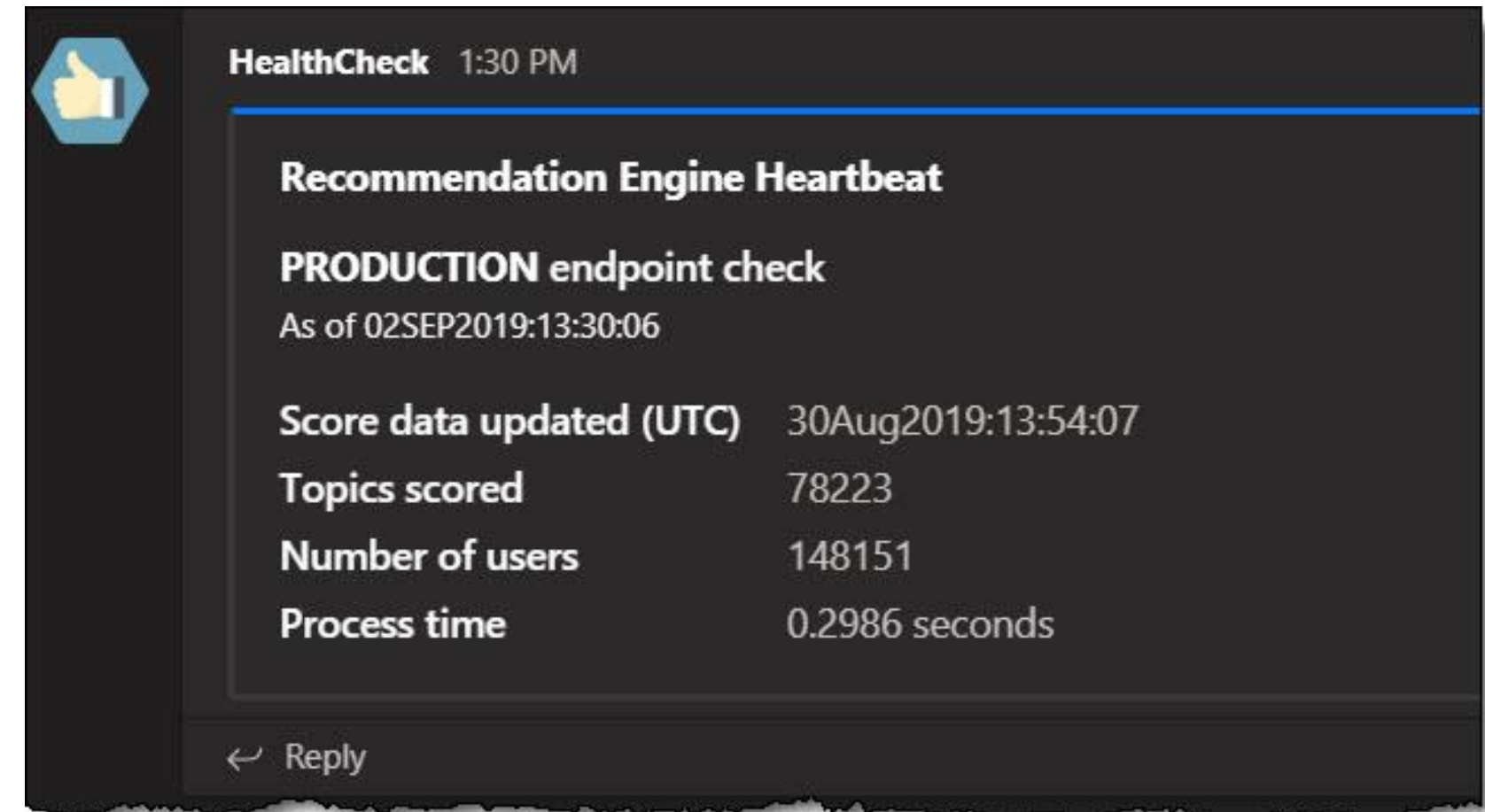
Generate with PROC JSON

§sas

Learning more about SAS and Microsoft 365

blogs.sas.com/sasdummy/tag/microsoft-office-365

Using SAS with Microsoft OneDrive and SharePoint Online

💬 33

By Chris Hemedinger on The SAS Dummy | November 29, 2018

Topics | Programming Tips

If your work environment is like ours here at SAS, you're seeing more of your data and applications move to the cloud. It's not yet a complete replacement for having local files on your desktop machine, but with cloud storage and apps -- like Microsoft OneDrive -- I can now access my work documents from any browser and any device, including my smartphone. I can update now my spreadsheets while waiting in the dentist office. Oh joy.

For those of us who use SAS to read and create Microsoft Excel documents, cloud-based files can add an extra wrinkle when we automate the process. It also adds some exciting possibilities! The Microsoft Office 365 suite offers APIs to discover, fetch, and update our documents using code. In this article, I'll show you how to use SAS programs to reach into your Microsoft OneDrive (or SharePoint Online) cloud to read and update your files. **Note:** All of this assumes that you already have a Microsoft Office 365 account -- perhaps provisioned by your IT support team -- and that you're using it to manage your documents.
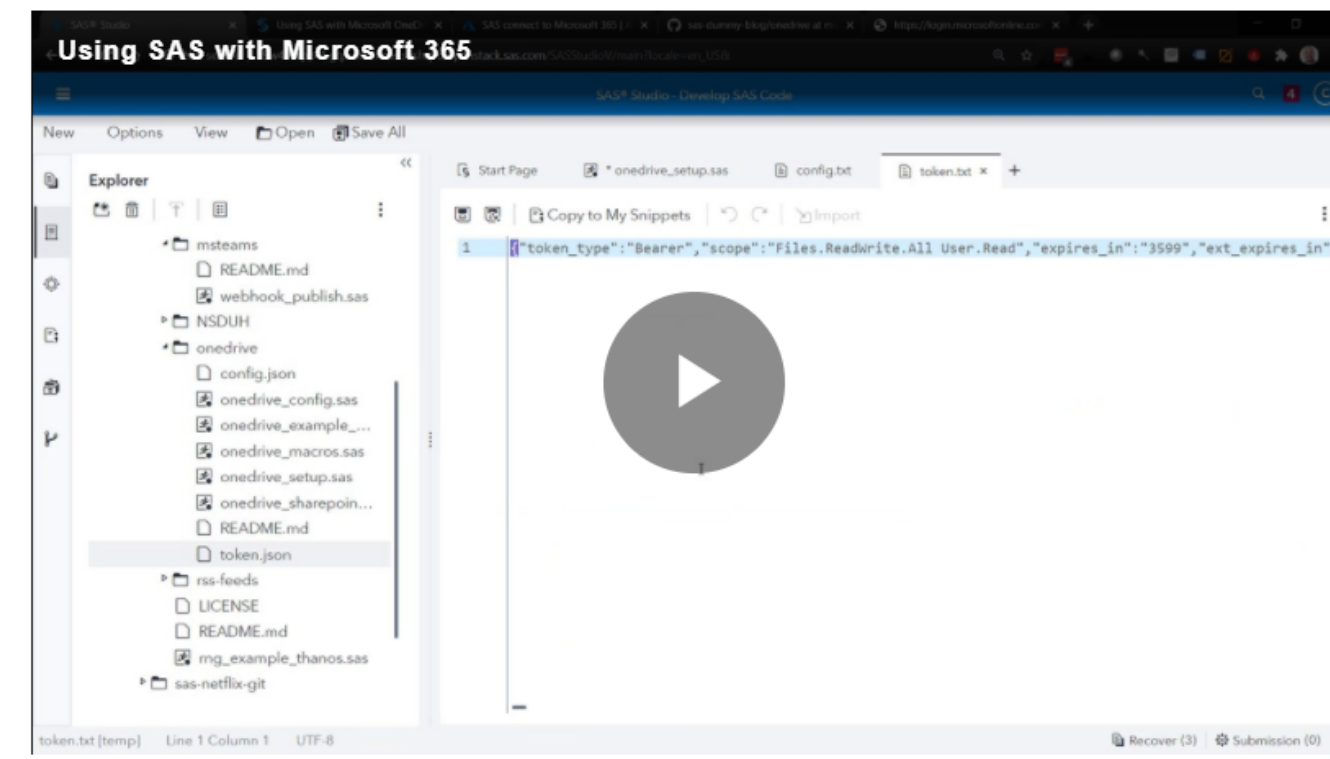
How to use SAS to access Microsoft 365

ChrisHemedinger

📅 Posted 07-08-2020 04:39 PM (2557 views)

Many of us are now using cloud-based productivity tools like Microsoft Office 365, hosted in Microsoft Azure. And some of us are using SAS software in the cloud, too. For those of us who use SAS to read and create Microsoft Excel documents, cloud-based files can add an extra wrinkle when we automate the process. It also adds some exciting possibilities!

The Microsoft 365 suite offers APIs to discover, fetch, and update our documents using code. In this video, I show you how to use SAS programs to reach into your Microsoft OneDrive, Microsoft Teams and SharePoint cloud to read and update your files. The approach relies on the REST APIs in Microsoft Office 365 and on the HTTP procedure in SAS.

Q & A

Chris.Hemedinger@sas.com

§sas