



# The Boston Area SAS<sup>®</sup> Users Group

Gathering SAS Users Since 1983

---

## How Do You Use SAS<sup>®</sup> to Access Data and APIs From the Web?



**Chris Hemedinger** is the Director of SAS User Engagement. His talented team looks after SAS online communities, SAS user groups, developer experience and GitHub, tech newsletters, expert webinars and tutorials. Chris is a recovering software developer who helped build popular SAS products such as SAS Enterprise Guide. Inexplicably, Chris is still coasting on the limited fame he earned as an author of SAS For Dummies. You can follow Chris on Twitter as [@cjdingler](#).

# How Do You Use SAS to Access Data and APIs From the Web?

For BASUG

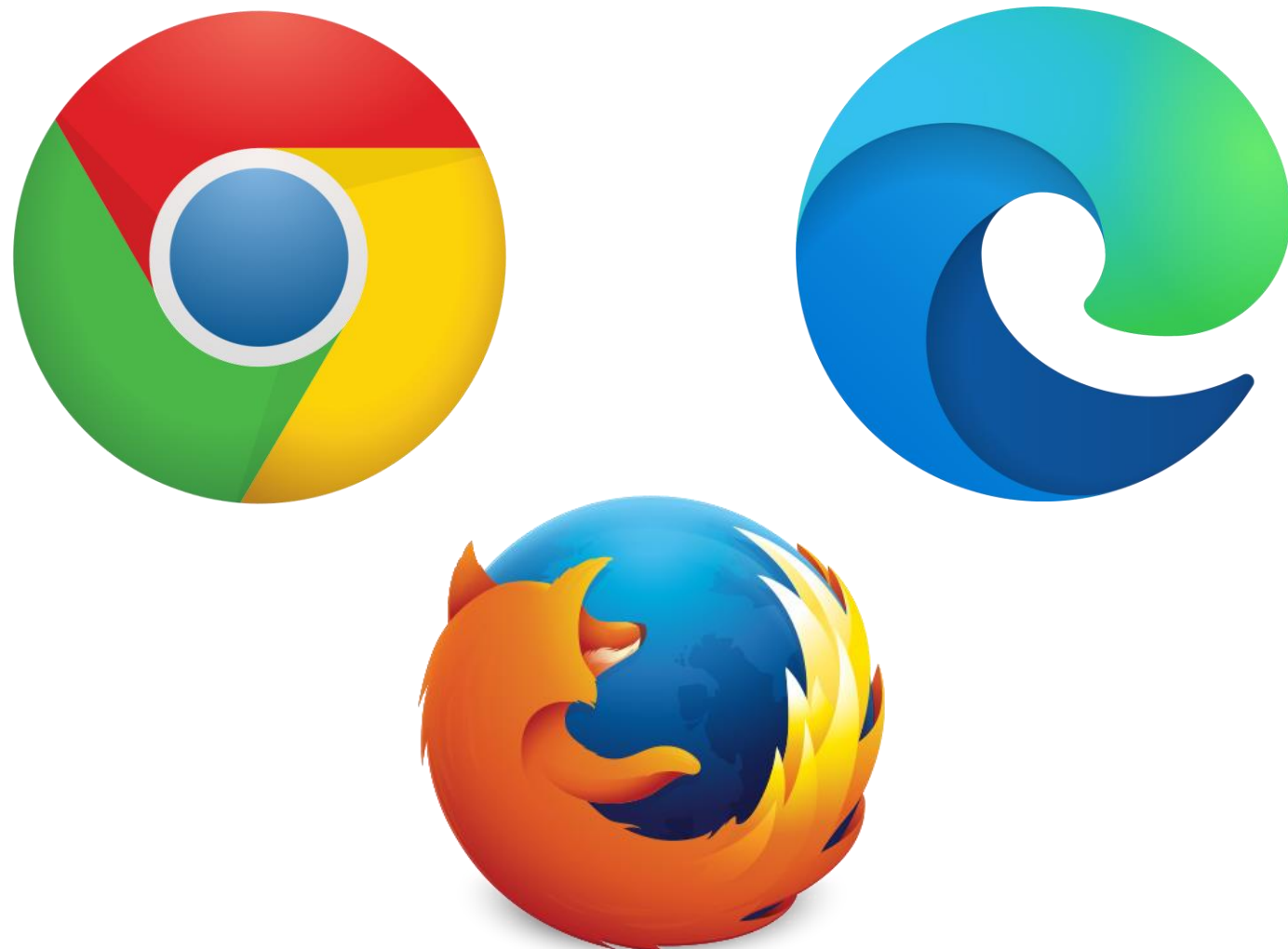
Chris Hemedinger, Director of SAS User Engagement



# The Internet: Not just for web pages

And content is not just for web browsers

## For end users



## For automated processes

`curl://`



**FILENAME URL**  
**PROC HTTP**

Any application that connects to a web site/service with HTTP is a web client

# Today's topics

- Intro to FILENAME URL and PROC HTTP
- Use case: Download a file and read it with SAS
- Use case: Download a file with user/password
- Use case: Download and read a ZIP file in SAS
- Use case: Post data to a site via web form
- Use case: "Scrape" data from a web page
- Use case: Call REST API with JSON response



# FILENAME URL

Simple. But limited.

```
filename data URL "https://sitewithdata.com/path-to-data";
```

Example:

```
filename data URL  
"https://www.federalreserve.gov/paymentsystems/files/coin_currircvolume.txt";
```

- One line!
- Supports HTTP GET method only
- Supports basic options: user/password, proxy\*, and a few others
- No control over where SAS stores the data that's fetched



# PROC HTTP

More lines of code. More power.

```
proc http
  method="method"
  url="http://sitewithdata.com/path"
  out=fileref;
```

Example:

```
filename data "/local/path/save-the-data-file";
```

```
proc http
  method="GET"
  url =
    "https://www.federalreserve.gov/paymentsystems/files/coin_currircvolume.txt"
  out=data;
```

```
run;
```

# PROC HTTP

**More lines. More power.**

- Supports HTTP GET, POST, PUT, and any other method
- Provides (requires) control for location of output
- Many options for authentication, caching, cookie behavior – aspects of a full HTTP client
- Rich DEBUG support
- SAS language analogy to cURL

```
proc http  
method="method"  
url="http://sitewithdata.com/"  
out=fileref;
```

**DEMO**

# Use case: Download a data file

Data from CMS.gov

COVID-19 Nursing Home Data

Information on COVID-19 reported by nursing homes to the CDC's National Healthcare Safety Network (NHSN) COVID-19 Long Term Care Facility Survey

Data update frequency  
Weekly

Latest data available  
October 9, 2022

Data source

Control and

Access API for COVID-19 Nursing Home Data

The Open Data API (ODA) provides programmatic access to this dataset including the ability to filter, query, and aggregate data.

TIME PERIOD	API ENDPOINT
Latest	JSON https://data.cms.gov/data-api/v1/data <a href="#">Copy</a>

[API Docs for the Dataset](#)

Download

Summary

Resources

```
filename nh temp;
proc http
  url="https://data.cms.gov/data-api/v1/dataset/137f90cb-
ac53-4b3d-8358-e65cf64e03d3/data"
  method="GET"
  out=nh;
run;

options validvarname=v7;
libname cms json fileref=nh;

data records;
  set cms.root;
run;

title; footnote;
proc freq data=records order=freq;
  table provider_name;
run;
```



# Download and import: Result

Log Output Data (1)

< Where | Query Builder | Tasks ▾

RECORDS

	ordinal_root	week_ending	federal_provider_number	provider_name	provider_address	provider_city	provider_state
1	1	2020-05-24	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
2	2	2020-05-31	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
3	3	2020-06-07	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
4	4	2020-06-14	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
5	5	2020-06-21	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
6	6	2020-06-28	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
7	7	2020-07-05	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
8	8	2020-07-12	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
9	9	2020-07-19	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL
10	10	2020-07-26	015009	BURNS NURSING HOME, INC.	701 MONROE STREET NW	RUSSELLVILLE	AL

Results (1)

>

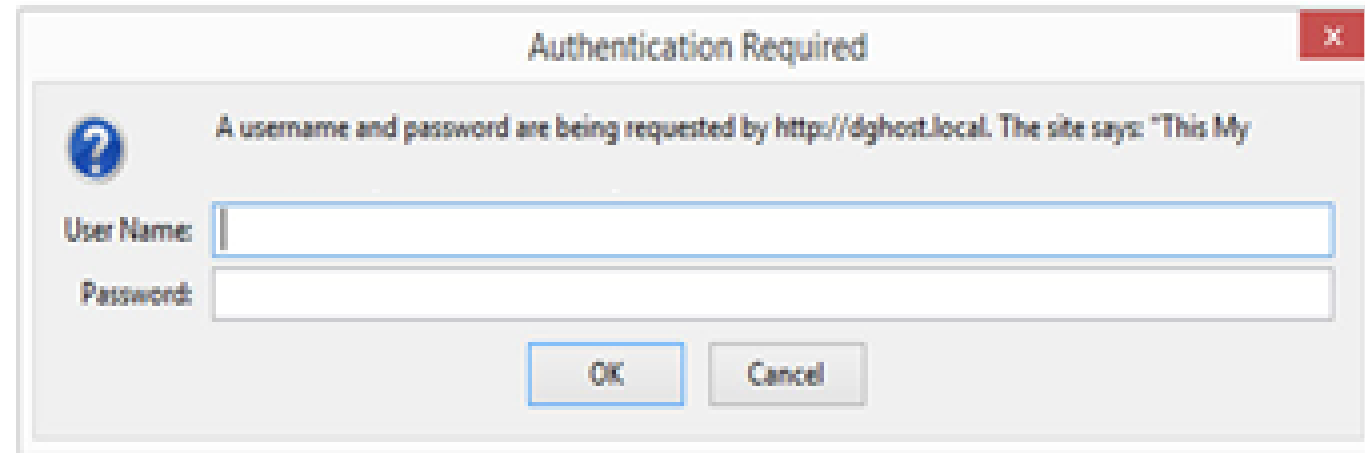
The FREQ Procedure

provider_name	Frequency	Percent	Cumulative Frequency	Cumulative Percent
ATHENS HEALTH AND REHABILITATION LLC	125	12.50	125	12.50
BURNS NURSING HOME, INC.	125	12.50	250	25.00
COOSA VALLEY HEALTHCARE CENTER	125	12.50	375	37.50
EASTVIEW REHABILITATION & HEALTHCARE CENTER	125	12.50	500	50.00
HATLEY HEALTH CARE INC	125	12.50	625	62.50
HIGHLANDS HEALTH AND REHAB	125	12.50	750	75.00
MERRY WOOD LODGE	125	12.50	875	87.50
PLANTATION MANOR NURSING HOME	125	12.50	1000	100.00



# Use case: Download a file with user/password

Access a site with Basic authentication



```
proc http
  url="www.secured-site.com"
  out=resp
  method="GET"
  WEBUSERNAME="user"
  WEBPASSWORD="pass"
  AUTH_BASIC
  AUTH_NEGOTIATE;
run;
```

# Example of AUTH\_BASIC

```
filename resp temp;
proc http
  url="http://httpbin.org/basic-auth/chris/pass123"
  method="GET"
  AUTH_BASIC
  out=resp
  webusername="chris"
  webpassword="pass123"
  ;
run;

data _null_;
  rc = jsonpp('resp', 'log');
run;
```

```
{
  "authenticated": true,
  "user": "chris"
}
```

Test methods using  
<http://httpbin.org/>

# Tip: Using httpbin.org to test logic

```
filename resp "%sysfunc(getoption(WORK))/stream.json";  
proc http  
  url="https://httpbin.org/stream/1"  
  method="GET"  
  out=resp;  
run;
```

```
/* Supported with SAS 9.4 Maint 5 */
```

```
%put HTTP Status code = &SYS_PROCHTTP_STATUS_CODE. : &SYS_PROCHTTP_STATUS_PHRASE.;
```

```
data _null_;  
  rc = jsonpp('resp', 'log');  
run;
```

```
HTTP Status code = 200 : OK
```

```
{  
  "url": "https://httpbin.org/stream/1",  
  "args": {  
  
  },  
  "headers": {  
    "Host": "httpbin.org",  
    "X-Amzn-Trace-Id": "Root=1-5f3bdd40-542722e0066832f012490708",  
    "User-Agent": "SAS/9",  
    "Accept": "*/*"  
  },  
  "origin": "199.199.188.99",  
  "id": 0  
}
```

# Use case: Download and read data in ZIP archive

Combine PROC HTTP and FILENAME ZIP

```
/* Download the ZIP file */
filename dl "%sysfunc(getoption(WORK))/ri130701_13dn01.zip";

proc http
  url="http://www.freddiemac.com/mbs/data/stacr/ri130701_13dn01.zip"
  method='GET'
  out=dl
  ct="application/zip";
run;

filename dl clear;

/* Crack it open and read with FILENAME ZIP */
filename inzip ZIP "%sysfunc(getoption(WORK))/ri130701_13dn01.zip";

data orig;
  infile inzip(ri130701_13dn01.txt);
...
```

# Use case: Post data to the web via form

Customer name:

Telephone:

E-mail address:

Pizza Size

Small

Medium

Large

Pizza Toppings

Bacon

Extra Cheese

Onion

Mushroom

Preferred delivery time:

Delivery instructions:


```
filename resp temp;
proc http
  url="http://httpbin.org/post"
  method="POST"
  in="custname=Chris&size=large&topping=cheese"
  out=resp;
run;

data _null_;
  rc = jsonpp('resp', 'log');
run;
```

```
"form": {
  "custname": "Chris",
  "size": "large",
  "topping": "cheese"
},
```



# Example of POST in action: SAS Hot fix notices



## How to stay current about SAS hot fixes

Use this space on the community to stay informed about the latest hot fixes to SAS software. You can subscribe to e-mail notifications, RSS feed, or take advantage of simple search tricks to find just the updates you need for your SAS products. [Read more](#) to learn how to use this space! Also read: [How to schedule and manage your SAS hot fixes](#)

< Previous
1
2
3
...
328
Next >

**Update available for SAS Viya 3.5**

An update is available for SAS Viya 3.5. Published 19Sep2023  
Issue(s) addressed: SAS Note 70368: You encounter an error when running the csCore\_set\_extrnDb\_configValues post-installation/post-onboarding script in SAS® Anti-Money...

0 0

**Update available for SAS Visual Investigator 10.8**

An update is available for SAS Visual Investigator 10.8. Published 19Sep2023 Issue(s) addressed: SAS Note 70371: Indexing fails with the error "no matching Note 70373: The alert recipient list in the

0 0

**Updates available for SAS Environment Manager 2.5\_M4 : Hot fix J9V012**

A new update is available for SAS Environment Manager , version 2.5\_M4 :Hot Fix: J9V012 - Published 19SEP2023 ,

**Updates available for SAS/ACCESS Interface to Impala 9.47 : Hot fix M5T002**

A new update is available for SAS/ACCESS Interface to Impala , version 9.47 :Hot Fix: M5T002 - Published 14SEP2023 , Download link for M5T002 Component name: SAS/ACCESS Interface to Impala Related SAS release: 9.4 Issues addresse...

0 0

**ALERT Updates available for SAS/STAT 15.1 : Hot fix I4X004**

A new update is available for SAS/STAT I4X004 - Published 13SEP2023 , Downlo Component name: SAS/STAT Related SA addressed in I4X004 SAS Note 70094.A

0 0

**Updates available for SAS Marketing Optimization 6.6 : Hot fix H8Q025**

A new update is available for SAS Marketing Optimization , version 6.6 :Hot Fix: H8Q025 - Published 12SEP2023 , Download link for H8Q025 Component name: SAS Marketing Optimization Related SAS release: 9.4 Issues addressed in...

0 0

**Updates available for SAS Marketing Automation 6.6 : Hot fix H8P026**

A new update is available for SAS Marke version 6.6 :Hot Fix: H8P026 - Published Download link for H8P026 Component n Automation Related SAS release: 9.4 Iss

0 0

### SAS Hot Fix Announcements articles

By Date v ↑

**Today**

**TS\_HOTFIX** 1:50 PM

**Update available for SAS Viya 3.5**

An update is available for SAS Viya 3.5. Published 19Sep2023

**TS\_HOTFIX** 1:47 PM

**Update available for SAS Visual Investigator 10.8**

An update is available for SAS Visual Investigator 10.8. Published 19Sep2023

**Yesterday**

**SAS\_TSNEWS** Thu 6:00 AM

**Updates available for SAS Environment Manager 2.5\_M4 : Hot fix J9V012**

A new update is available for SAS Environment Manager , version 2.5\_M4 :

**Monday**

**SAS\_TSNEWS** Mon 6:00 AM

**Updates available for SAS/ACCESS Interface to Impala 9.47 : Hot fix M5T002**

A new update is available for SAS/ACCESS Interface to Impala , version 9.47 :

**Last Week**

**SAS\_TSNEWS** Fri 9/15

**ALERT Updates available for SAS/STAT 15.1 : Hot fix I4X004**

A new update is available for SAS/STAT , version 15.1 :

**Update available for SAS Viya 3.5**

## TS\_HOTFIX

Posted On Fri 9/22/2023 1:50 PM  
Feed SAS Hot Fix Announcements articles

An update is available for SAS Viya 3.5.

Published 19Sep2023

Issue(s) addressed:

SAS Note [70368](#): You encounter an error when running the csCore\_set\_extrnDb\_configValues post-installation/post-onboarding script in SAS® Anti-Money Laundering

For a complete list of SAS Viya updates available, visit the [SAS Viya Updates Page](#).

[View article...](#)



# Use case: Scrape data from the web

But first, an editorial

Web scraping is lossy, fragile process.

The information on the web page does not include data types, lengths, or constraints metadata.

And one tweak to the presentation of the web page can break any automated scraping process.

# Use case: Scrape data from the web

But if you must...

For this step	Use these features
Get the contents of the web page	<a href="#">PROC HTTP</a> or <a href="#">FILENAME URL</a>
Process/parse the web page contents	DATA step, with parsing functions such as FIND, <a href="#">SCAN</a> , and <a href="#">regular expressions via PRXMATCH</a> . Use SAS informats to convert text to native data types.
Repeat across subsequent pages	SAS macro language (%DO %UNTIL processing) or DATA step with <a href="#">CALL EXECUTE</a> to generate multiple iterations of the fetch/parse steps.



# Example of "web scraping"

National Notifiable Diseases Surveillance System (NNDSS)

CDC > NNDSS > Surveillance Case Definitions

Search Results for All Conditions

Search Conditions:  Search All Conditions

Notifiable Condition Lists: 2020

Infectious  Non-Infectious  Outbreaks

Name	Acronym	Common Name	Etiology	Notifiable From	Notifiable To
<a href="#">Acanthamoeba disease (excluding keratitis)</a>					
<a href="#">Acanthamoeba keratitis</a>					
<a href="#">Acquired immunodeficiency syndrome</a>	AIDS			2000	2008
<a href="#">Acute Flaccid Myelitis (AFM)</a>	AFM				
<a href="#">Amebiasis</a>			<i>Entamoeba histolytica</i>	1933	1994
<a href="#">Anaplasma phagocytophilum infection</a>				2008	Current
<a href="#">Anthrax</a>			<i>Bacillus anthracis</i>	1944	Current
<a href="#">Arboviral diseases, neuroinvasive and non-neuroinvasive</a>				2005	Current

```
<table class="text-centered table" style="margin-bottom:0 !important; width:100%; ">
  <thead>...</thead>
  <tbody>
    <tr>
      <td style="text-align:left;vertical-align:middle;">
        <a href="/nndss/conditions/acanthamoeba-disease-excluding-keratitis/" class="tp-link-po
          <i>Acanthamoeba</i>
            " disease (excluding keratitis)
            "
          <span class="print-only">...</span>
        </a>
      </td>
      <td class="tablet-hidden"></td>
      <td class="tablet-hidden"></td>
      <td class="tablet-hidden"></td>
      <td> </td>
      <td> </td>
    </tr>
  </tbody>
</table>
```

```
/* Get all of the nonblank lines */
filename CDC url "https://www.cdc.gov/nndss/conditions/search/";
data rep;
infile CDC length=len lrecl=32767;
input line $varying32767. len;
  line = strip(line);
  if len>0;
run;
filename CDC clear;

/* Parse the lines and keep just condition names */
/* When a condition code is found, grab the line following (full name of
condition) */
/* and the 8th line following (Notification To date)
*/
/* Relies on this page's exact layout and line break scheme */
data parsed (keep=condition_code condition_full note_to);
  length condition_code $ 40 condition_full $ 60;
  set rep;
  if find(line, "/nndss/conditions/") then do;
    condition_code=scan(line,4, '/');
    pickup= _n_+1 ;
    pickup2 = _n_+8;
    set rep (rename=(line=condition_full)) point=pickup;
    set rep (rename=(line=note_to)) point=pickup2;
    output;
  end;
run;
```

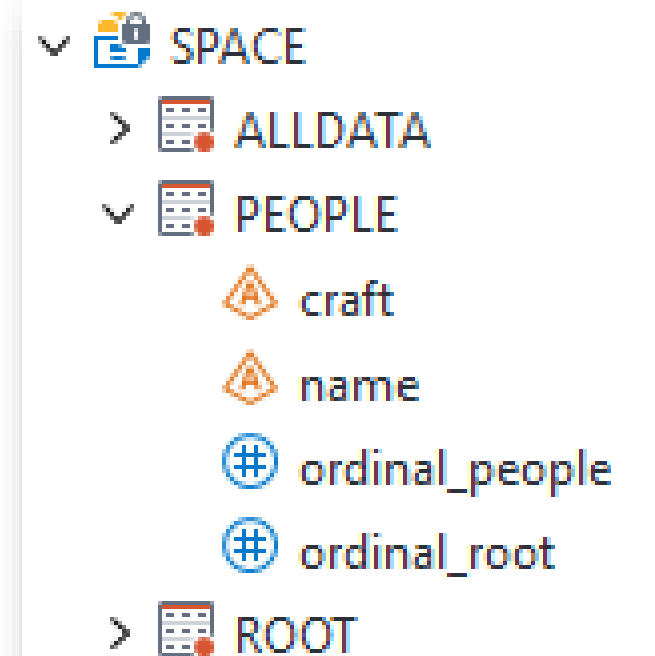
# Use case: REST API with JSON response

```
/* Neat service from Open Notify project */
filename resp temp;
proc http
  url="http://api.open-notify.org/astros.json"
  method= "GET"
  out=resp;
run;

data _null_;
  rc = jsonpp('resp', 'log');
run;

/* Assign a JSON library to the HTTP response */
libname space JSON fileref=resp;
```

```
{
  "people": [
    {
      "name": "Sergey Prokopyev",
      "craft": "ISS"
    },
    {
      "name": "Dmitry Petelin",
      "craft": "ISS"
    },
    {
      "name": "Frank Rubio",
      "craft": "ISS"
    },
    {
      "name": "Jing Haiping",
      "craft": "Tiangong"
    },
    {
      "name": "Gui Haichow",
      "craft": "Tiangong"
    },
    {
      "name": "Zhu Yangzhu",
      "craft": "Tiangong"
    },
    {
      "name": "Jasmin Moghbeli",
      "craft": "ISS"
    }
  ]
}
```



# Other functions to help

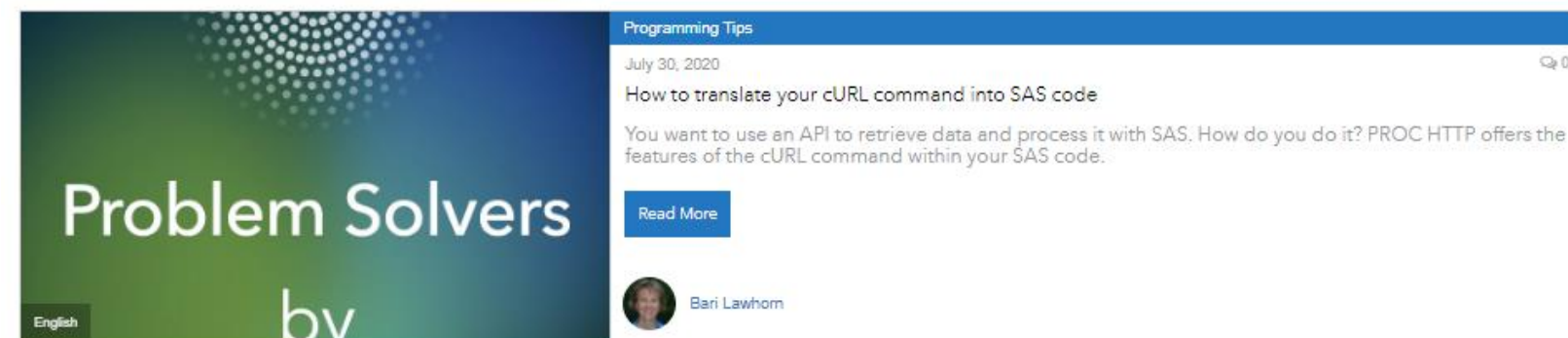
<a href="#"><u>HTMLDECODE Function</u></a>	Decodes a string that contains HTML numeric character references or HTML character entity references and returns the decoded string.
<a href="#"><u>HTMLENCODE Function</u></a>	Encodes characters using HTML character entity references and returns the encoded string.
<a href="#"><u>URLDECODE Function</u></a>	Returns a string that was decoded using the URL escape syntax.
<a href="#"><u>URLENCODE Function</u></a>	Returns a string that was encoded using the URL escape syntax. From: <code>urlencode(%str(ga:sessions,ga:pageviews,ga:users));</code> To: <code>ga%3Asessions,ga%3Apageviews,ga%3Ausers</code>
<code>%str(&amp;)</code>	Prevent SAS from interpreting “&” in a URL as a macro variable. <code>in="custname=Joe%str(&amp;)size=large%str(&amp;)topping=cheese"</code> Avoids " <code>WARNING: Apparent symbolic reference TOPPING not resolved.</code> "
JSONPP function	"Pretty print" a JSON response to a file or log



# Learn more

<https://blogs.sas.com/content/tag/proc-http/>

Tag: PROC HTTP



**Problem Solvers**

English bv


**Programming Tips**

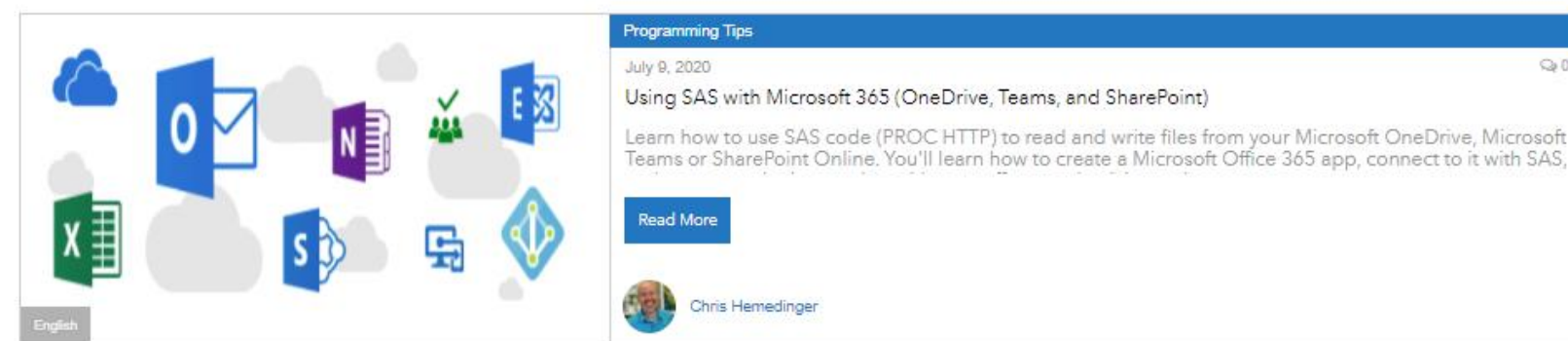
July 30, 2020

How to translate your cURL command into SAS code

You want to use an API to retrieve data and process it with SAS. How do you do it? PROC HTTP offers the features of the cURL command within your SAS code.

[Read More](#)

 Bari Lawhom




**Programming Tips**

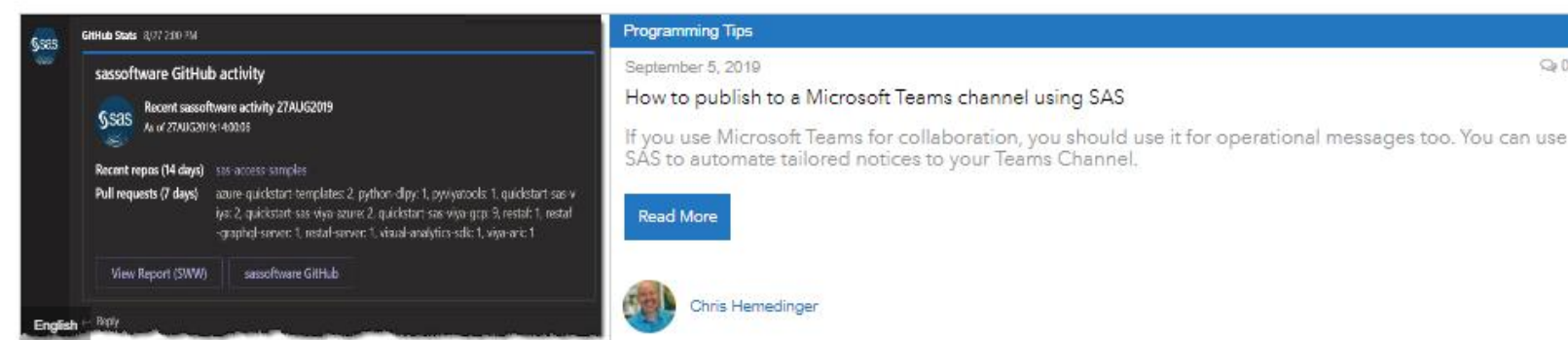
July 9, 2020

Using SAS with Microsoft 365 (OneDrive, Teams, and SharePoint)

Learn how to use SAS code (PROC HTTP) to read and write files from your Microsoft OneDrive, Microsoft Teams or SharePoint Online. You'll learn how to create a Microsoft Office 365 app, connect to it with SAS,

[Read More](#)

 Chris Hemedinger




**Programming Tips**

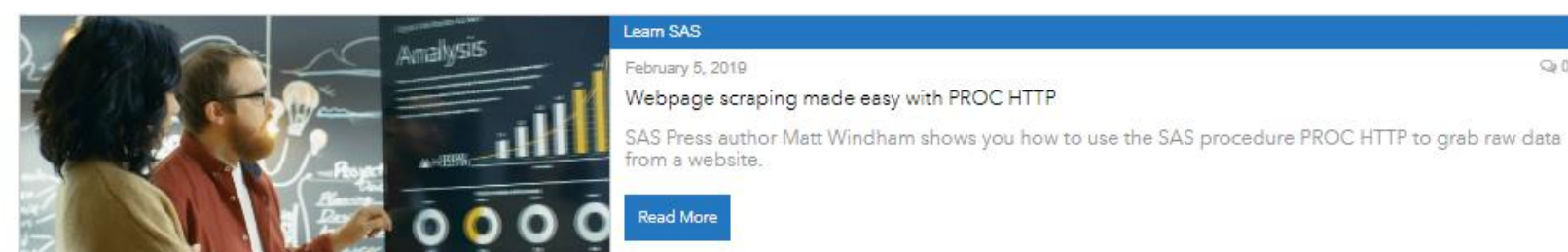
September 5, 2019

How to publish to a Microsoft Teams channel using SAS

If you use Microsoft Teams for collaboration, you should use it for operational messages too. You can use SAS to automate tailored notices to your Teams Channel.

[Read More](#)

 Chris Hemedinger



**Learn SAS**

February 5, 2019

Webpage scraping made easy with PROC HTTP

SAS Press author Matt Windham shows you how to use the SAS procedure PROC HTTP to grab raw data from a website.

[Read More](#)

SAS Conference papers

[ABCs of PROC HTTP](#)

[REST Just Got Easy with SAS and PROC HTTP](#)

Ask the Expert webinar  
[How Do You Use SAS® to Access Data and APIs From the Web?](#)



# BASUG proud moments

2023 User Group Leader award



Quentin McMullen

2023 Distinguished Educator of the Year



Michael Salé

**Happy 40<sup>th</sup> Birthday BASUG!**

