

How to Use Git with Your SAS Projects

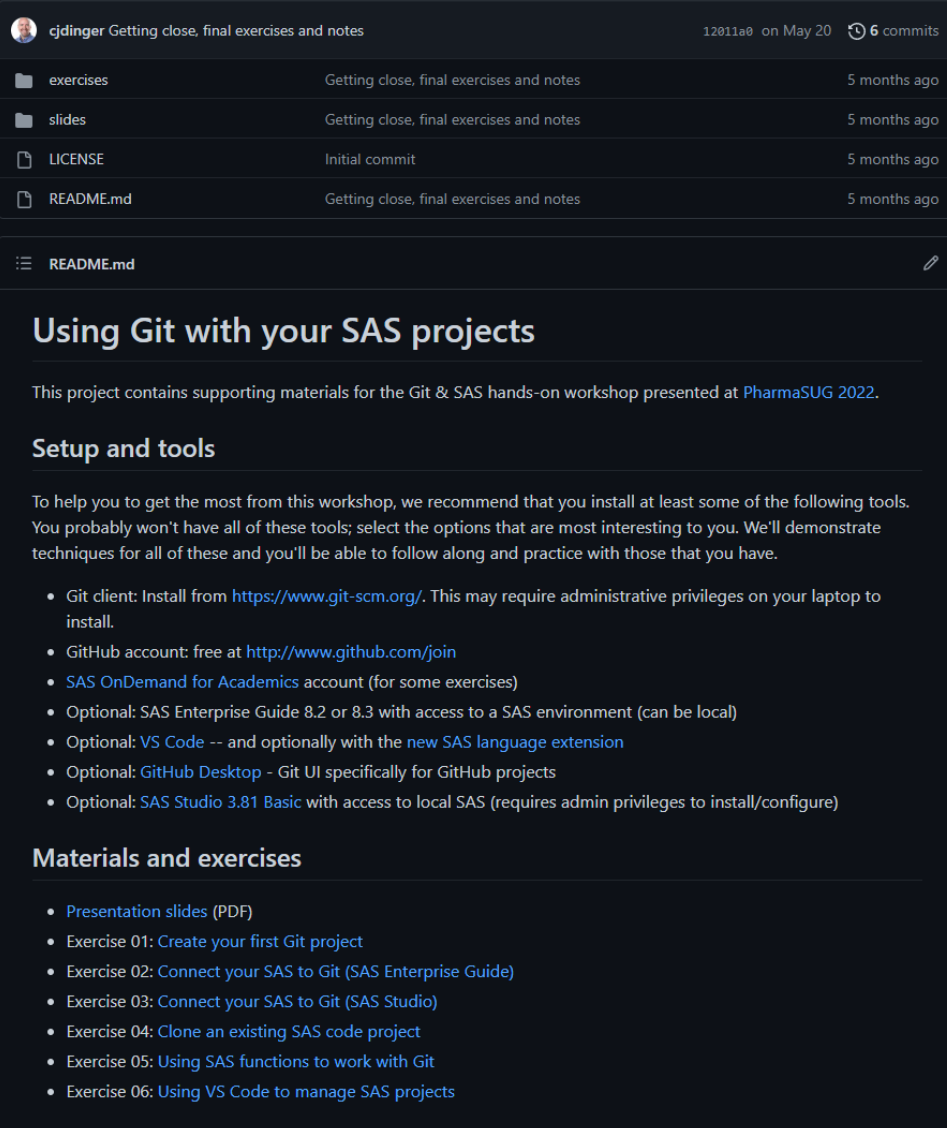


Chris Hemedinger is the Director of SAS User Engagement. His talented team looks after SAS online communities, SAS user groups, developer experience and GitHub, tech newsletters, expert webinars and tutorials.

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*.

Workshop available on GitHub

[github.com/sascommunities](https://github.com/sascommunities/git-workshop-pharmasug)
[git-workshop-pharmasug](https://github.com/sascommunities/git-workshop-pharmasug)



The screenshot shows the GitHub repository page for 'git-workshop-pharmasug' by user 'cjdinger'. The repository description is 'Getting close, final exercises and notes'. It has 12811a8 on May 20 and 6 commits. The file list includes 'exercises', 'slides', 'LICENSE', and 'README.md', all updated 5 months ago. The 'README.md' file is selected, showing the title 'Using Git with your SAS projects'. The content includes a description of the project's purpose, a 'Setup and tools' section with a list of recommended tools, and a 'Materials and exercises' section with a list of exercises.

Using Git with your SAS projects

This project contains supporting materials for the Git & SAS hands-on workshop presented at [PharmaSUG 2022](#).

Setup and tools

To help you to get the most from this workshop, we recommend that you install at least some of the following tools. You probably won't have all of these tools; select the options that are most interesting to you. We'll demonstrate techniques for all of these and you'll be able to follow along and practice with those that you have.

- Git client: Install from <https://www.git-scm.org/>. This may require administrative privileges on your laptop to install.
- GitHub account: free at <http://www.github.com/join>
- SAS OnDemand for Academics account (for some exercises)
- Optional: SAS Enterprise Guide 8.2 or 8.3 with access to a SAS environment (can be local)
- Optional: VS Code -- and optionally with the [new SAS language extension](#)
- Optional: GitHub Desktop - Git UI specifically for GitHub projects
- Optional: SAS Studio 3.81 Basic with access to local SAS (requires admin privileges to install/configure)

Materials and exercises

- [Presentation slides \(PDF\)](#)
- Exercise 01: Create your first Git project
- Exercise 02: Connect your SAS to Git (SAS Enterprise Guide)
- Exercise 03: Connect your SAS to Git (SAS Studio)
- Exercise 04: Clone an existing SAS code project
- Exercise 05: Using SAS functions to work with Git
- Exercise 06: Using VS Code to manage SAS projects



Today's topics

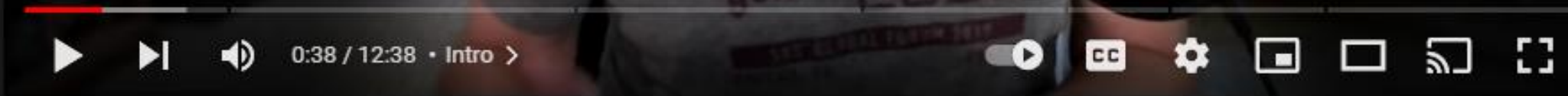
- What is Git™?
- Using Git with SAS tools
- Git functions in SAS programming language
- DevOps opportunities: collaboration and continuous integration
- Where to learn more

Git and the Git logo are either registered trademarks or trademarks of Software Freedom Conservancy, Inc., corporate home of the Git Project, in the United States and/or other countries.

youtube.com/sasusers

VERSION

CONTROL



#SASUsers #LearnSAS

SAS Tutorial | How to use Git with SAS



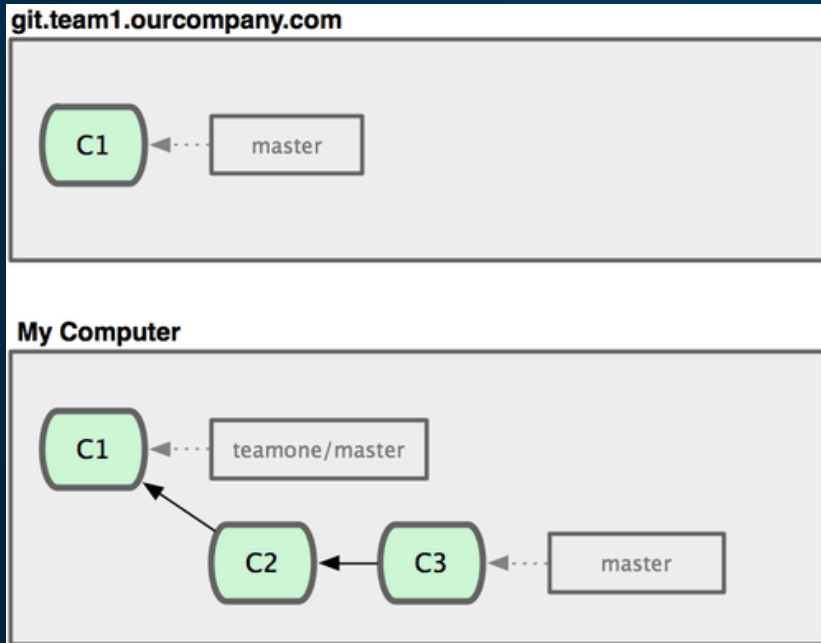
What is Git?

Git is a **distributed version control system** that supports several collaborative and standalone workflows

- **Distributed** – each developer has a “clone” of the code repository
- **Workflows** – feature-based, experiment branches, prod/test/dev, patch/cherry-pick
- **Open source** – Git software is free and open source.

Several commercial systems (GitHub, GitLab, Bitbucket, and more) add features and enterprise-readiness.

Source: <https://git-scm.com>



Benefits of using Git with SAS

Version control

Author: Mike Henderson | Date: 4/29/2020 15:54:21 | Commit hash: a059de1248793e17b9ff00ea8b194ecfb81f384e

flexible number of ISOChangeDate and SocialDistancingChange

- ISOChangeDate is now a string delimited by : that can handle none, sing
- SocialDistancingChange is now a string delimited by : that can handle no
- Break - If the file run_scenarios.csv has a column added it will not be use

CCF/COVID19.sas

```
7
8 %macro EasyRun (Scenario,Incubatio
9 - SocialDistancing,
10 - ISOChangeDate3, So
11 + SocialDistancing,
12 MarketShareParam
```

DevOps/automation

GitHub Maintainers > sassoftware.org ...

Conversations Files Wiki +

GitHub 5/1 12:36 PM

Pull request closed: "fix a bug in load_weights\(\)"

kesmitt13
5/1/2019 12:36 PM (UTC-05:00)

The push fixes a bug in load_weights(). The issue happens when load caffemodel.h5 as a model and model weights. The path option is a server-side path. 'if os.path.isfile(path):' is intended to check if the file exists but it is operating on client-side. I don't think we have a method to check server-side file by client python, so I remove the outer if statement.

Repository: sassoftware/python-dlpy

Pull request #: 133

Add a comment

View in GitHub

Reply

Collaboration

Start Page covid-19-sas x				
Commit (95) History				
Graph	Message	Author	Date	Comm
	new cc templates	Mike Henderson	4/30/2020 15:10:2	9161d
	Merge pull request #6 from sassoftware/Social-Distanc	Mike Henderson	4/30/2020 13:17:2	d4e09
	origin/Social-Distancing-Shifts Adjusted to handle	Mike Henderson	4/30/2020 13:05:1	6ec13
	Added supporting documentation	Chris Hemedinger	4/30/2020 11:45:2	8980c
	rebuild: change in Sigma default in keyword paramete	Mike Henderson	4/30/2020 10:48:3	65969f2
	Add SAS programs to fetch public COVID-19 data	Chris Hemedinger	4/30/2020 10:37:3	ee08ba9
✓ master	Add SAS code to fetch public COVID-19	Chris Hemedinger	4/30/2020 10:11:1	d2e382c
	progress	Mike Henderson	4/30/2020 09:46:2	379c816
	flexible number of ISOChangeDate and SocialDistancir	Mike Henderson	4/29/2020 15:54:2	a059de1
	Update orchestrate.sas	Andrew Williams	4/29/2020 11:04:3	9be4870
	remove extra variables from model_final	Mike Henderson	4/28/2020 11:30:3	e5ae023

Branch: — 2s No changes
Commit: — a few seconds ago Started by user Gate Demo078



Git has its own lingo

Clone

- Create a local copy of a repository

Fork

- Make a local copy of a repo where you **don't have push access**

Pull request

- **Open a discussion** with proposed code changes to be merged to upstream repo

Branch

- A logical location to divert from the “main line” of code to stage changes

Blame

- Annotated code listing with revision/author for each line

Commit

- **Record a set of changes to the local repo**

Commit History

- Log that shows chronological list of changes to the repo

Push / Pull

- Update a remote repo with local changes, or sync/merge local repo with remote changes

SAS tools that integrate with Git

SAS Enterprise
Guide

SAS Studio

SAS
programming

SAS Data
Integration
Studio

And more!
SAS Viya products add
more Git integration

SAS Enterprise Guide and Git

Supports Git **internal** to project and/or with **external** repos

Project file (EGP)

.git repo

- EmbeddedProgram1.sas
Revisions / history
- EmbeddedProgram2.sas
Revisions / history
- EmbeddedProgram3.sas
Revisions / history

Project file (EGP) | No project

- ↗ LinkedProgram1.sas
- ↗ LinkedProgram2.sas
- ↗ LinkedProgram3.sas

File system



.git repo

- LinkedProgram1.sas
Revisions / history
- LinkedProgram2.sas
Revisions / history
- LinkedProgram3.sas
Revisions / history

SAS Enterprise Guide with external repo

The screenshot displays the SAS Enterprise Guide (MIS 9.4) interface with the 'sas-youtube-api' repository selected. The left sidebar shows 'Git Repositories' and 'Servers'. The main window is divided into several panes:

- Git Repositories:** Lists 'sas-youtube-api' as the current repository and other repositories like 'sas-dummy-blog', 'sas-netflix-git', 'learn-sas', and 'sas-zip-macros'.
- Start Page:** Shows the repository name 'sas-youtube-api' and the current branch 'main'. It includes buttons for 'Pull' and 'Push'.
- Commit History:** A table showing the commit history for the 'main' branch.
- Graph:** A visual representation of the commit history.
- Commit Details:** Shows the author 'saschr', date '8/12/2020 21:29:51', and commit hash 'e5d63c7994af2dbdbaba66af63f3f6c78a6d361b'. The commit message is 'Catch up work'.
- Files:** A list of files in the repository, including 'adhoc_plots.sas', 'adhoc_plots2.sas', 'bc_prep.sas', 'bc_sasgf_monthly.sas', 'youtube.sh', 'yt_sasgf_meta.sas', 'yt_sasuserschannel_buildvatable.sas', and 'yt_sasuserschannel_countries.sas'.
- Diff View:** Shows the difference between the current branch and the 'main' branch. The diff includes a new file 'b/adhoc_plots.sas' and a change to 'a/adhoc_plots.sas'.

The diff view shows the following changes:

```
1 diff --git a/adhoc_plots.sas b/adhoc_plots.sas
2 new file mode 100644
3 index 0000000..8d0a130
4 --- /dev/null
5 +++ b/adhoc_plots.sas
6 ----- Difference (@@ -0,0 +1,27 @@) -----
7 +ods html5(id=eghtml) gtitle;
8 +title "SAS User YouTube views";
9 +proc sgplot data=yt.youtube_sasuser_va;
10 + vbar month / response=views dataskin=matte;
11 + where level="CHANNEL";
12 +
13 +
14 +
15 +run;
16 +
```

All history maintained within Git repo available to **any** tool that integrates with Git.

Git integration with SAS Studio

The screenshot displays the SAS Studio interface with the following components:

- Top Bar:** SAS Studio - Develop SAS Code
- Menu Bar:** New, Options, View, Open, Save All
- Git Repositories Sidebar:**
 - Clone, Add
 - Current Repository: sas-netflix-git (master)
- Main Area:**
 - Repository: sas-netflix-git
 - Branch: master
 - Buttons: Pull master (Last pulled: Feb 9, 2022, 3:5...), Push master (Last pushed: Jan 19, 2021, 10:28:...)
 - Commit History Table:
- Commit History Table:**

Commit	Message	Author	Date	Commit ID
✓ master	origin/master update viewing histories	Chris Hemedinger	Jan 1, 2021, 12:17:49 PM	6f96051
	Update program to get my Netflix data	Chris Hemedinger	Nov 18, 2020, 12:25:50 PM	8b572fa
	refresh Netflix account history	Chris Hemedinger	Nov 13, 2020, 4:48:16 PM	a243b10
	Updating files for SUGA demo	cjdinger	Mar 4, 2020, 3:30:59 PM	2608f59
	Commit my comment	Chris Hemedinger	May 8, 2019, 2:21:57 PM	8daab35
		cjdinger	May 8, 2019, 11:34:39 AM	6ee1461
- Diff View:**
 - Author: Chris Hemedinger
 - Date: Jan 1, 2021, 12:17:49 PM
 - Commit ID: 6f96051f0b9bd7dc5b23f7957e67c11f46b2af0f
 - Message: update viewing histories
 - Parent: 8b572fa
 - Diff: diff --git a/NetflixData/NetflixViewingHistory_Dadasaurus.csv b/NetflixData/NetflixViewingHistory_Dadasaurus.csv index 0e3...@@ -1,15 +1,125 @@Title,Date+"Chilling Adventures of Sabrina: Part 4: Chapter Twenty-Nine: The Eldritch Dark", "1/1/21"+"The Office (U.S.): Season 6: The Chump", "12/31/20"+"Indiana Jones and the Raiders of the Lost Ark", "12/31/20"

Available as of v3.8 (SAS 9.4) and SAS Viya 3.5

Complete support for local and remote repo management

New Git functions in SAS language

Functions added in SAS 9.4 Maint 6 and SAS Viya

New SAS functions that mirror most of the Git commands that users are familiar with.

Select functions - All documented by searching [“Git functions” on support.sas.com.](https://support.sas.com)

GITFN_CLONE	Clones a Git repository (for example, from GitHub) into a directory on the SAS server.
GITFN_COMMIT	Commits staged files to the local repository
GITFN_DIFF	Returns the number of diffs between two commits in the local repository and creates a diff record object for the local repository.
GITFN_PUSH	Pushes the committed files in the local repository to the remote repository.
GITFN_NEW_BRANCH	Creates a Git branch

Name change: GITFN* → GIT* in SAS Viya

Demo

Connecting SAS to Git

SSH or HTTPS

SAS Studio

Supports SSH or HTTPS

The screenshot shows the 'Edit Profile' dialog box in SAS Studio. At the top, there are two tabs: 'SSH' (selected) and 'HTTPS'. The form contains the following fields: 'Profile name: *' with the value 'Chris (GitHub)', 'User name: *' with the value 'cjdingner', 'Email: *' with a redacted email address, 'Public SSH file path:' with the value '/.ssh/id_ed25519.pub' and a file icon, and 'Private SSH file path:' with the value '/.ssh/id_ed25519' and a file icon. At the bottom are 'OK' and 'Cancel' buttons.

SAS Enterprise Guide 8.2

Supports HTTPS

The screenshot shows the 'Manage Git Connections' dialog box in SAS Enterprise Guide. It has a tree view on the left with 'Git Repositories' expanded, showing 'Profiles' and 'Repositories'. 'Profiles' is selected, and 'GitHub' is checked. The 'Edit Profile' sub-dialog is open, showing fields for 'Profile name: *' (GitHub), 'User name: *' (cjdingner), 'Email: *' (redacted), and 'Password: *' (masked with dots). 'OK' and 'Cancel' buttons are at the bottom.

SAS GIT* functions

Supports SSH or HTTPS

```
/* Fetch latest code from GitHub */  
data _null_;  
  rc = gitfn_clone(  
    "git@github.com:sascommunities/sas-dummy-blog.git",  
    "&repoPath.",  
    "cjdingner", ""  
    "/u/sasrh/.ssh/id_rsa.pub",  
    "/u/sasrh/.ssh/id_rsa");  
  put rc=;  
run;
```

Generating an SSH key

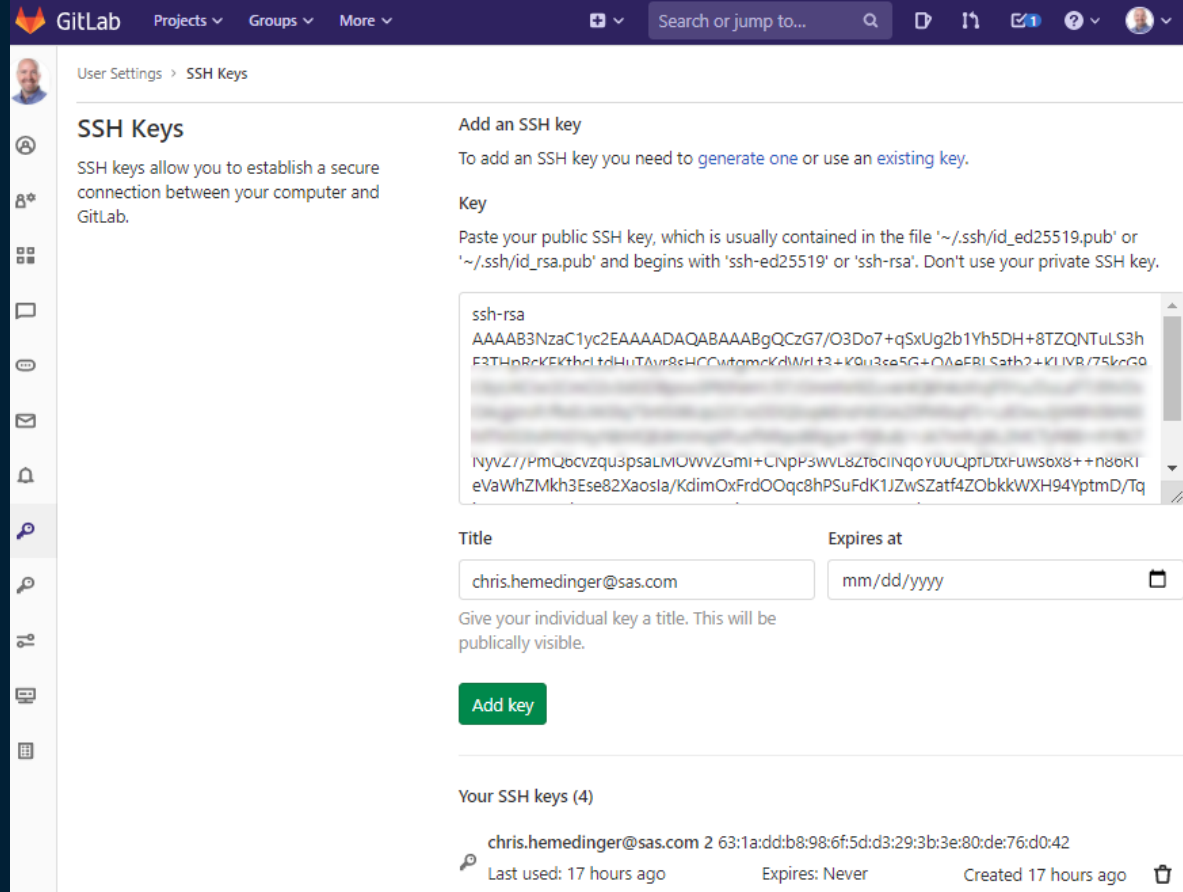
SSH for passwordless access

- To generate a key for use with SAS Studio or SAS GIT functions:
`ssh-keygen -t ecdsa -b 521 -C <email>`
- Will generate two files – a public and private key
 - Ex: id_rsa.pub and id_rsa
- Register the **public** key file with Git service
- **Protect** the **private** key file (readable by your account only)

Register SSH key

GitLab example

GitLab settings



The screenshot shows the GitLab 'User Settings > SSH Keys' page. It includes a sidebar with navigation icons, a main heading 'SSH Keys', and a description of SSH keys. Below this is a form to 'Add an SSH key' with a text area for the key, a 'Title' field, and an 'Expires at' date field. A green 'Add key' button is at the bottom. A section titled 'Your SSH keys (4)' lists existing keys with details like title, last used time, expiration, and creation time.

GitLab Projects Groups More

User Settings > SSH Keys

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

To add an SSH key you need to [generate one](#) or use an [existing key](#).

Key

Paste your public SSH key, which is usually contained in the file '~/.ssh/id_ed25519.pub' or '~/.ssh/id_rsa.pub' and begins with 'ssh-ed25519' or 'ssh-rsa'. Don't use your private SSH key.

```
ssh-rsa
AAAAAB3NzaC1yc2EAAAADAQABAAQGCzG7/O3Do7+qSxUg2b1Yh5DH+8TZQNTuLS3h
E3THpRcKEKthcltdHuITAvr8sHCCwfmKdM/dt3+K9u3se5G+QAeFRI Sath2+K1UVR/75kcG9
...
InyvZ7/PmQbcvzqu3psaLMUwvZGmi+CNpP3wvL8Zt6ciNqoY0UQptDtxFuwsx8++n8oRT
eVaWhZMkh3Ese82Xaosla/KdimOxFrDOOqc8hPSuFdK1JZwSZatf4ZObkkWXH94YptmD/Tq
```

Title **Expires at**

chris.hemedinger@sas.com mm/dd/yyyy

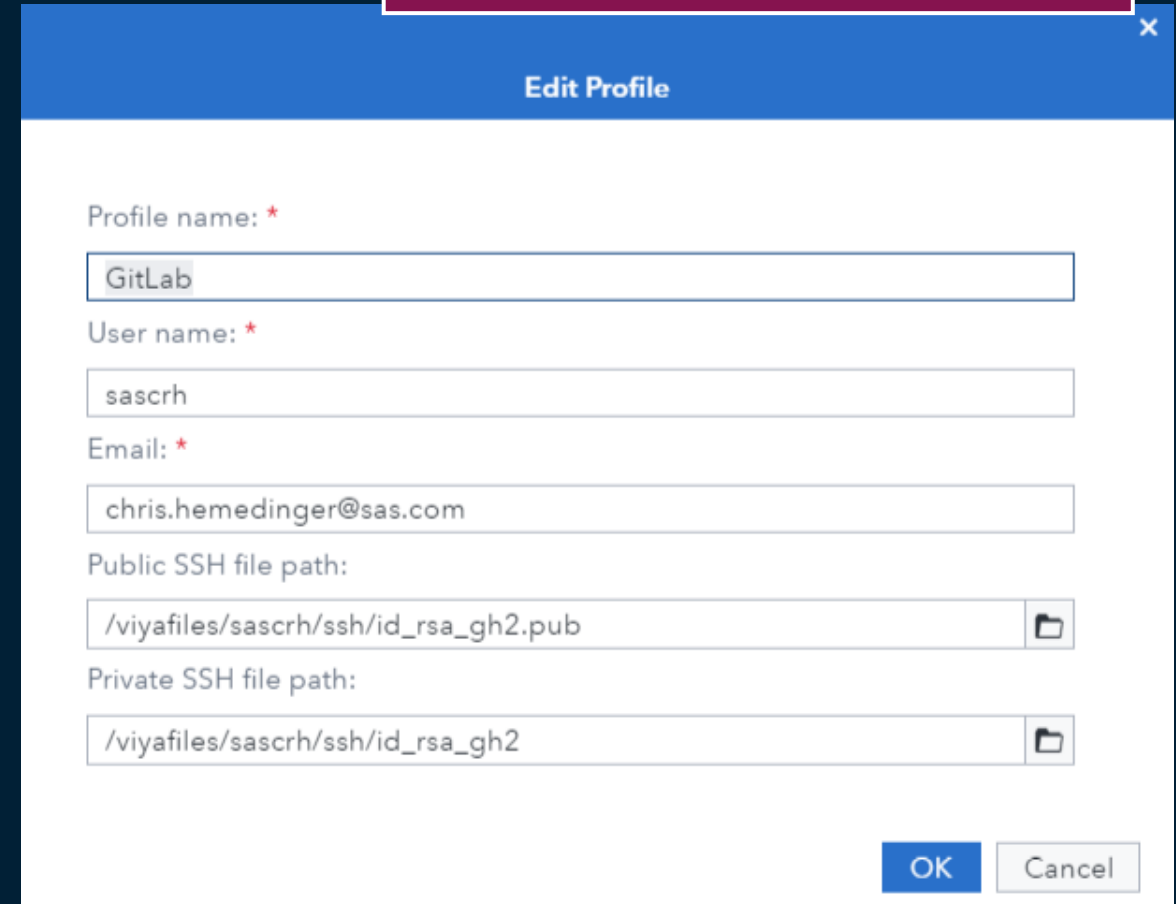
Give your individual key a title. This will be publically visible.

Add key

Your SSH keys (4)

chris.hemedinger@sas.com	2 63:1a:dd:b8:98:6f:5d:d3:29:3b:3e:80:de:76:d0:42
Last used: 17 hours ago	Expires: Never Created 17 hours ago

SAS Studio settings



The screenshot shows the 'Edit Profile' dialog in SAS Studio. It contains fields for 'Profile name', 'User name', 'Email', 'Public SSH file path', and 'Private SSH file path'. Each field has a text input and a folder selection icon. At the bottom are 'OK' and 'Cancel' buttons.

Edit Profile

Profile name: *

GitLab

User name: *

sasrh

Email: *

chris.hemedinger@sas.com

Public SSH file path:

/viyafiles/sasrh/ssh/id_rsa_gh2.pub

Private SSH file path:

/viyafiles/sasrh/ssh/id_rsa_gh2

OK Cancel

Generate an access token for HTTPS

Required for single-signon (SSO) or 2-factor auth

- HTTPS access requires username and password
- Instead of account password, use access token
- Access tokens convey specific permissions (**api**, **read_repository**, **write_repository**, etc.)
- **Protect your access token!**

Creating a personal access token

You should create a personal access token to use in place of a password with the command line or with the API.

Personal access tokens (PATs) are an alternative to using passwords for authentication to GitHub when using the [GitHub API](#) or the [command line](#).

If you want to use a PAT to access resources owned by an organization that uses SAML SSO, you must authorize the PAT. For more information, see "[About authentication with SAML single sign-on](#)" and "[Authorizing a personal access token for use with SAML single sign-on](#)."

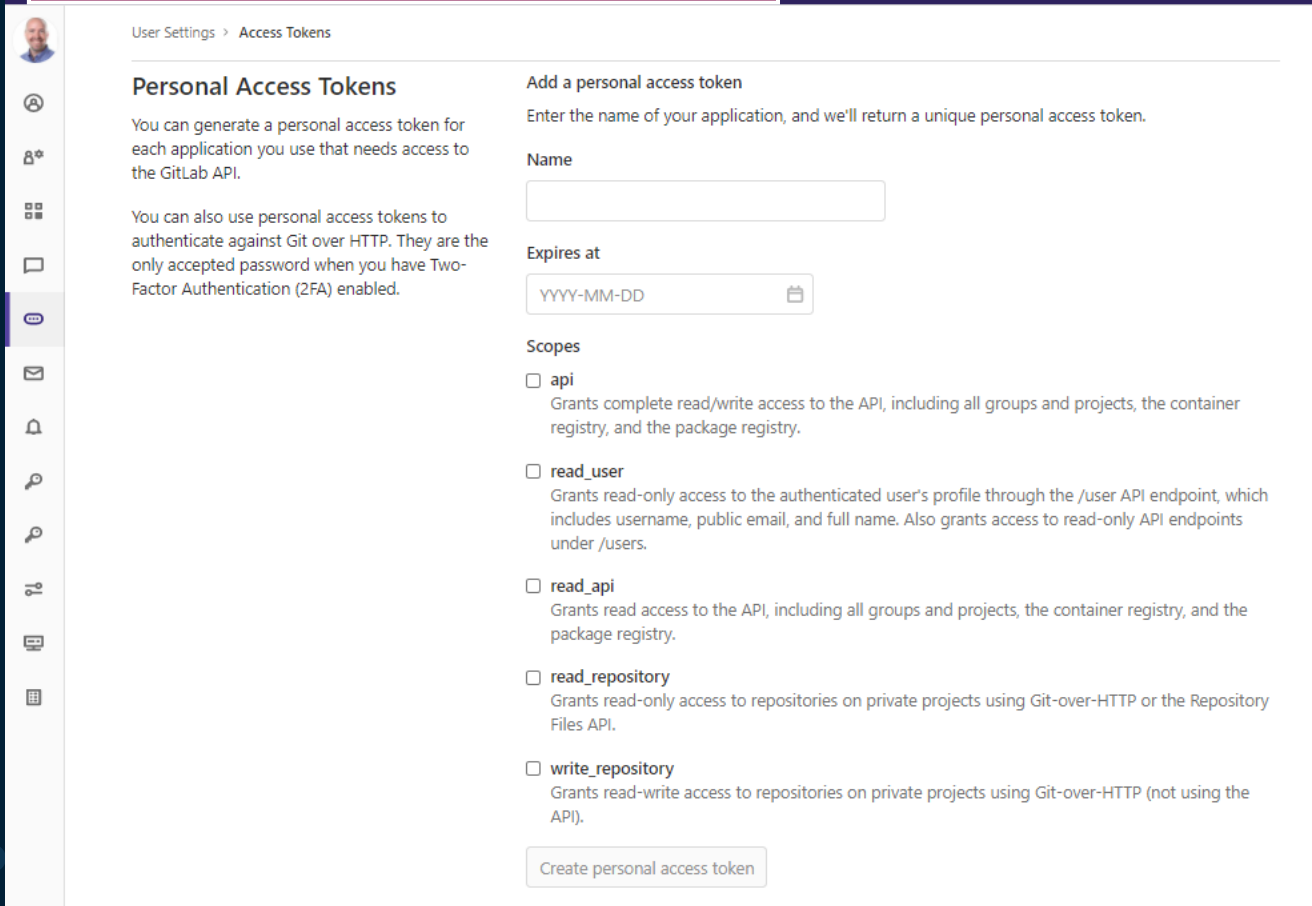
As a security precaution, GitHub automatically removes personal access tokens that haven't been used in a year.

From GitHub doc

Register access token

GitLab example

GitLab settings



The screenshot shows the 'User Settings > Access Tokens' page in GitLab. It includes a sidebar with navigation icons, a search bar, and a main content area. The 'Personal Access Tokens' section explains that tokens can be generated for applications or used for HTTP authentication. Below this, there's a form to 'Add a personal access token' with fields for 'Name' and 'Expires at' (YYYY-MM-DD). A 'Scopes' section lists several permissions: 'api' (checked), 'read_user', 'read_api', 'read_repository', and 'write_repository'. A 'Create personal access token' button is at the bottom.

User Settings > Access Tokens

Personal Access Tokens

You can generate a personal access token for each application you use that needs access to the GitLab API.

You can also use personal access tokens to authenticate against Git over HTTP. They are the only accepted password when you have Two-Factor Authentication (2FA) enabled.

Add a personal access token
Enter the name of your application, and we'll return a unique personal access token.

Name

Expires at

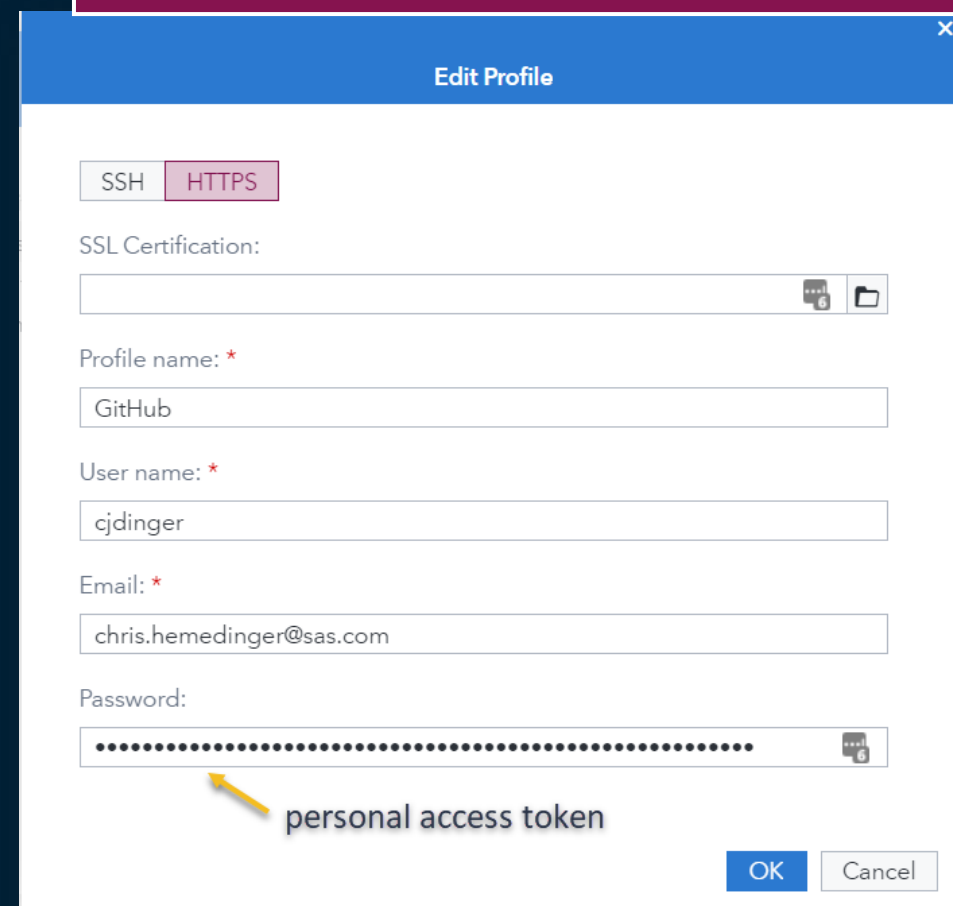
YYYY-MM-DD

Scopes

- ☒ **api**
Grants complete read/write access to the API, including all groups and projects, the container registry, and the package registry.
- ☐ **read_user**
Grants read-only access to the authenticated user's profile through the /user API endpoint, which includes username, public email, and full name. Also grants access to read-only API endpoints under /users.
- ☐ **read_api**
Grants read access to the API, including all groups and projects, the container registry, and the package registry.
- ☐ **read_repository**
Grants read-only access to repositories on private projects using Git-over-HTTP or the Repository Files API.
- ☐ **write_repository**
Grants read-write access to repositories on private projects using Git-over-HTTP (not using the API).

Create personal access token

SAS Studio settings



The screenshot shows the 'Edit Profile' dialog in SAS Studio. It has a blue header bar. Below it, there are tabs for 'SSH' and 'HTTPS'. The 'HTTPS' tab is selected. The 'SSL Certification' field is empty. The 'Profile name' field contains 'GitHub'. The 'User name' field contains 'cjdingner'. The 'Email' field contains 'chris.hemedinger@sas.com'. The 'Password' field is masked with dots. A yellow arrow points to the password field with the text 'personal access token'. At the bottom right are 'OK' and 'Cancel' buttons.

Edit Profile

SSH HTTPS

SSL Certification:

Profile name: *

User name: *

Email: *

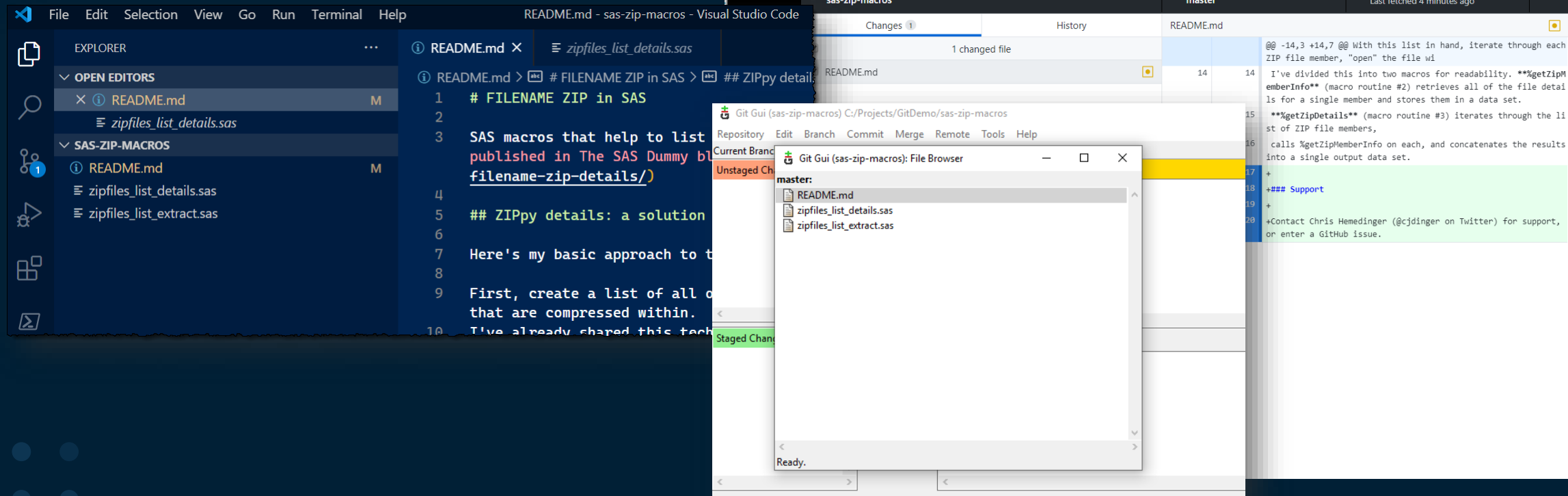
Password:

personal access token

OK Cancel

Use other tools alongside SAS

- Git Bash (CLI)
- Git GUI or GitHub Desktop
- VS Code



How to organize your SAS projects in Git

Read the blog article

- Don't create one huge repo
- Don't organize based on dept org structure
- Design with collaboration in mind
- Re-use 👍, Ownership 🗨️
- Work **with** Git features, not against them

How to organize your SAS projects in Git

By [Chris Hemedinger](#) on [The SAS Dummy](#) | November 10, 2020

Topics | [Programming Tips](#)

The code and data that drive analytics projects are important assets to the organizations that sponsor them. As such, there is a growing trend to manage these items in the source management systems of record. For most companies these days, that means Git. The specific system might be GitHub Enterprise, GitLab, or Bitbucket -- all platforms that are based on Git.

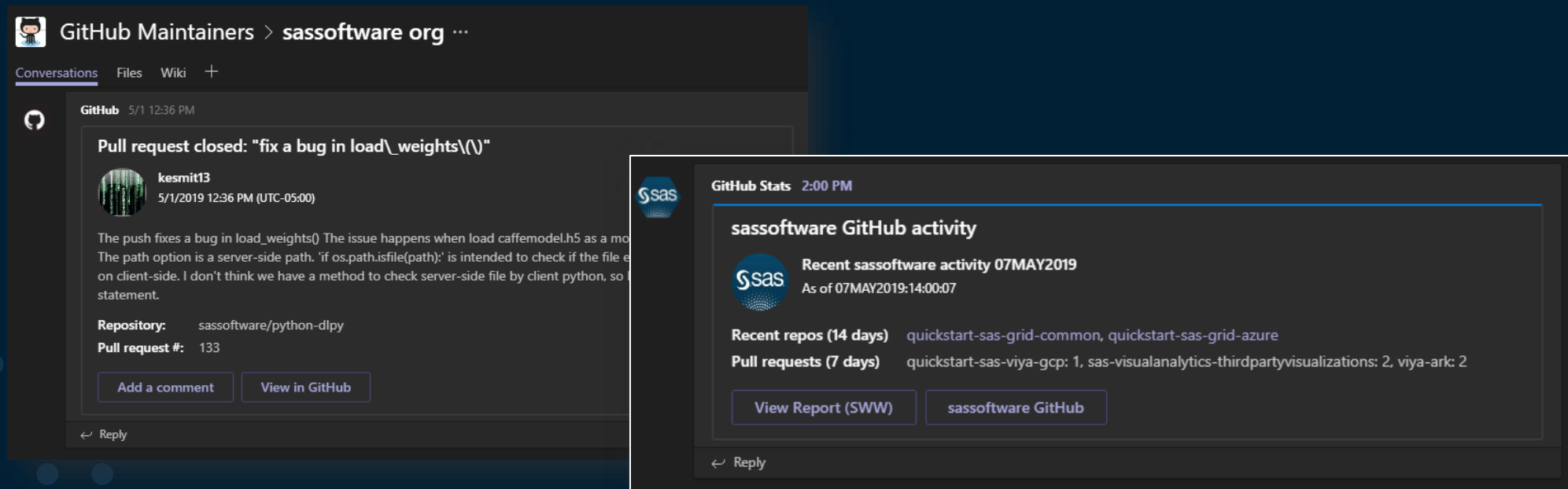
Many SAS products support direct integration with Git. This includes SAS Studio, SAS Enterprise Guide, and the SAS programming language. (That last one checks a **lot** of boxes for ways to use Git and SAS together.) While we have good documentation and videos to help you learn about Git and SAS, we often get questions around "best practices" -- what is **the best/correct way** to organize your SAS projects in Git?

<https://blogs.sas.com/content/sasdummys/sas-projects-git/>

Git integration is just the beginning

From there, trigger other operations

- Gerrit for code review
- Jenkins or Travis CI for build, deployment, and other continuous integration
- Trigger notifications/alerts in Slack, Microsoft Teams



The screenshot displays two overlapping GitHub interface elements. The background element is a pull request page for the repository 'sassoftware/python-dlpy', showing a pull request titled 'fix a bug in load_weights()' by user 'kesmit13'. The foreground element is a 'GitHub Stats' widget for 'sassoftware', showing recent activity as of 07MAY2019, including recent repositories and pull requests.

GitHub Maintainers > sassoftware org

Conversations Files Wiki +

GitHub 5/1 12:36 PM

Pull request closed: "fix a bug in load_weights(\)"

kesmit13
5/1/2019 12:36 PM (UTC-05:00)

The push fixes a bug in load_weights(). The issue happens when load_caffemodel.h5 as a model. The path option is a server-side path. 'if os.path.isfile(path):' is intended to check if the file exists on client-side. I don't think we have a method to check server-side file by client python, so I added a statement.

Repository: sassoftware/python-dlpy
Pull request #: 133

[Add a comment](#) [View in GitHub](#)

[Reply](#)

GitHub Stats 2:00 PM

sassoftware GitHub activity

Recent sassoftware activity 07MAY2019
As of 07MAY2019:14:00:07

Recent repos (14 days) quickstart-sas-grid-common, quickstart-sas-grid-azure
Pull requests (7 days) quickstart-sas-viya-gcp: 1, sas-visualanalytics-thirdpartyvisualizations: 2, viya-ark: 2

[View Report \(SWW\)](#) [sassoftware GitHub](#)

[Reply](#)

Learn more

- [Using Git with SAS](#) (SAS Users YouTube)
- [Git functions in SAS 9.4 and SAS Viya](#) (doc)
- [Using Git in SAS Enterprise Guide](#) (doc)
- [Git with SAS Studio and SAS Enterprise Guide](#) (video)
- [SAS Software on GitHub](#)
- [Using built-in Git operations in SAS](#) (blog)
- [How to organize your SAS projects in Git](#) (blog)
- [DevOps with SAS 9: SAS code, GitLab, and Jenkins](#) (community)
- [developer.sas.com](#) for SAS app development
- [Pro Git](#) by Scott Chacon and Ben Straub, free online book about Git

A series of horizontal bars of varying lengths and colors (teal, blue, and purple) are arranged on the left side of the slide, creating a modern, abstract background element.

Q & A

sas.com

