

PSBlueSky Help¹

<https://jdhitsolutions.github.io>

v2.4.0 - 22 Jan 2025

¹<https://github.com/jdhitsolutions/PSBluesky>

Contents

PSBluesky	2
Installation	2
Module Information	3
Open-BskyHelp	3
Get-BskyModuleInfo	3
Authentication	4
Session and Tokens	4
Credentials	5
App Passwords	5
Rate Limits	7
Posting	7
Reposting and Quoting	8
Profiles	9
Followers	10
Searching for Users	11
Following	13
Blocking	14
Feed	15
Notifications	16
Timeline	16
API Logging	17
Ad-Hoc Scripting	19
Information and Troubleshooting	21
Other Module Features	21
Custom Formatting	21
Type Extensions	22
Custom Verbose Messaging	23
Preferences	23
Roadmap	25

PSBluesky



This module is a set of PowerShell functions designed to let you interact with the Bluesky API from a PowerShell session. Technically, the module commands are wrappers around the [AT protocol](#). The module is written for **PowerShell 7**, although it might work as written in Windows PowerShell with minimal changes if you wish to fork the GitHub repository. Commands *have not* been thoroughly tested for cross-platform compatibility, so please post an Issue if you encounter a problem.

The commands in this module are **not** intended to provide complete coverage of the Bluesky API or user experience. Instead, the module focuses on the most common tasks you might want to do with Bluesky from a PowerShell prompt.

For best results, you should run this module in a PowerShell console that supports emojis and ANSI formatting like Windows Terminal.

Installation

You can install the module from the PowerShell Gallery.

```
Install-Module -Name PSBluesky
```

Or using Install-PSResource from the [Microsoft.PowerShell.PSResourceGet](#) module.

```
Install-PSResource -Name PSBluesky -Repository PSGallery -TrustRepository
```

You might want to also install the following or related modules to securely store your Bluesky credentials:

- [Microsoft.PowerShell.SecretManagement](#)
- [Microsoft.PowerShell.SecretStore](#)

Module Information

After installing this module, you should end up with these PSBluesky commands:

Add-BskyImage	Block-BskyUser
Disable-BskyLogging	Enable-BskyLogging
Export-BskyPreference	Find-BskyUser
Get-BskyAccountDID	Get-BskyBlockedList
Get-BskyBlockedUser	Get-BskyFeed
Get-BskyFollowers	Get-BskyFollowing
Get-BskyLiked	Get-BskyLogging
Get-BskyModuleInfo	Get-BskyNotification
Get-BskyPreference	Get-BskyProfile
Get-BskySession	Get-BskyTimeline
New-BskyFollow	New-BskyPost
Open-BskyHelp	Publish-BskyPost
Remove-BskyFollow	Remove-BskyLogging
Remove-BskyPreferenceFile	Set-BskyLogging
Set-BskyPreference	Start-BskySession
Unblock-BskyUser	Update-BskySession

Open-BskyHelp

After importing the module you can run [Open-BskyHelp](#) which will open a PDF version of this document in the default application associated with PDF files. Note that not all emojis will render properly in the PDF. Or you can use the `-AsMarkdown` parameter to read this file using markdown formatting. Not all Markdown features may properly render in the console.

Get-BskyModuleInfo

You can use [Get-BskyModuleInfo](#) to get a summary of the module. The default output includes clickable links to online command help and the module's GitHub repository.

```
PS C:\> Get-BskyModuleInfo

Module: PSBlueSky [v2.4.0]

Name      Alias      Synopsis
----      -
Add-BskyImage          Upload an image to Bluesky.
Block-BskyUser          Block a Bluesky user account.
Disable-BskyLogging     Disable BlueSky API logging.
Enable-BskyLogging      Enable BlueSky API logging.
Export-BskyPreference    Export your PSBlueSky format ...
Find-BskyUser           bsu          Search for Bluesky user accou...
Get-BskyAccountDID      Resolve a Bluesky account nam...
Get-BskyBlockedList     bsbblocklist Get your subscribed blocked l...
Get-BskyBlockedUser     bsbblock     Get your blocked accounts.
```

Get-BskyFeed	bsfeed	Get your Bluesky feed.
Get-BskyFollowers	bsfollower	Get your Bluesky followers
Get-BskyFollowing	bsfollow	Get a list of Bluesky accounts ...
Get-BskyLiked	bsliked	Get your liked Bluesky posts.
Get-BskyLogging		Getting the current PSBlueSky ...
Get-BskyModuleInfo		Get a summary of the PSBlueSky ...
Get-BskyNotification	bsn	Get Bluesky notifications.
Get-BskyPreference		Get PSBlueSky formatting prefe...
Get-BskyProfile	bsp	Get a Bluesky profile.
Get-BskySession	bss	Show your current Bluesky session.
Get-BskyTimeline	bst	Get your Bluesky timeline.
New-BskyFollow	Follow-BskyUser	Follow a Bluesky user.
New-BskyPost	skeet	Create a Bluesky post.
Open-BskyHelp	bshelp	Open the PSBluesky help document.
Publish-BskyPost	Repost-BskyPost	Repost or quote a Bluesky post.
Remove-BskyFollow	Unfollow-BskyUser	Unfollow a Bluesky user.
Remove-BskyPreferenceFile		Delete the user's PSBlueSky pre...
Set-BskyLogging		Configure PSBlueSky API logging.
Set-BskyPreference		Set a PSBlueSky formatting pref...
Start-BskySession		Start a new Bluesky session.
Unblock-BskyUser		Unblock a Bluesky user account.
Update-BskySession	Refresh-BskySession	Refresh the Bluesky session token.

The output will include clickable links to the online help for each command.

Authentication

Session and Tokens



In order to send data, you must authenticate. Version 2.0.0 of this module introduced a new session model. After you import the module, you **must run** `Start-BskySession` to initialize the module and setup module-scoped variables.

```
Start-BskySession -credential $cred
```

The credential must be a `PSCredential` object representing your Bluesky username (handle) and password. (See below)

This command will create a hidden session object that will be called from other module commands to get the necessary authentication token for the `Invoke-ResetMethod` header. The access token has a limited lifetime unless it is refreshed. Beginning with version 2.4.0, the module will refresh the token every 60 minutes through a background runspace using a synchronized hashtable. If you remove the module, the runspace will be removed as well.

Run `Get-BskySession` to see your current session information.

```
PS C:\> Get-BskySession
```

```
User: jdhitsolutions.com

Active AccessToken RefreshToken Age
-----
True eyJ0eXAiOiJhdCtqd3Qi... eyJ0eXAiOiJyZWZyZXNo... 00:05:41.5115796
```

If you want to manually test the Bluesky API in your own code, you can use the access and refresh token properties from this object.

```
PS C:\> $bskySession = Get-BskySession
PS C:\> $access = $bskySession.AccessToken
PS C:\> $refresh = $bskySession.RefreshToken
```

However, it is still possible you will encounter an expired token error message. If you do, you can run `Update-BskySession` to refresh the token.

```
Get-BskySession | Update-BskySession
```

If this also fails, remove the module, re-import and start a new Bluesky session, including re-running `Start-BskySession`.

Credentials

You will need to create a `PSCredential` object with your Bluesky username and password. **The username is case-sensitive.** For automation purposes, you can use the Secrets management module to store your credential. Write your own code to retrieve the credential and pass it to the module commands.

You might want to use `PSDefaultParameterValues` to set the credential for all commands.

```
$PSDefaultParameterValues['*-Bsky*:Credential'] = $BlueskyCredential
```

You should only need this credential for `Start-BskySession`.

This module does not support 2FA. You must use an app password.

App Passwords

While this module does not use your credential object for anything other than establishing a Bluesky session, you may elect to take a more secure approach and use an app password. This password can be revoked at any time without affecting your main account password. Follow these steps if you want to use an app password with this module.

In the Bluesky app, go to your profile and select `Settings`. Then select `App Passwords`.

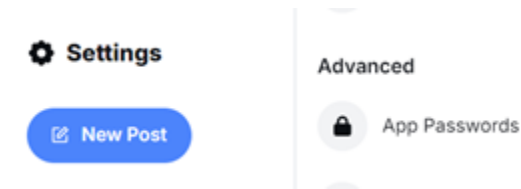


Figure 1: Bluesky app passwords

You will need to create a new app password. Give it a meaningful name.

Figure 2: Bluesky creating an app password

Click the button to create the app password.

Figure 3: Bluesky app password

You will only see the password once. Copy the password and save it in a secure location such as a secrets management vault.

Next, you need to create a credential object in PowerShell. The user name will be the DID (decentralized identifier) for your account. The password will be the app password you just created. You can run `Get-BskyAccountDID` to get your DID. This command does not require authentication.

```
$did = Get-BskyAccountDID jdhitsolutions.com
```

Now you can create a credential object.

```
PS C:\> $cred = Get-Credential $did

PowerShell credential request
Enter your credentials.
Password for user did:plc:ohgsqpfsbocaaxusxqlgfvd7:
```

Enter the app password. You will use this credential object with `Start-BskySession`. I recommend you save the credential in a secrets management vault and retrieve it when you need to start a Bluesky session.

```
$cred = Get-Secret -Name PSBlueskyCredential
$PSDefaultParameterValues['*-Sky*:Credential'] = $cred
```

You can revoke an app password at any time. If you do, you will need to create a new app password and update your credential object.



I recommend you use an app password with this module and protect your primary account password. If your Bluesky account is protected with 2FA, you *must* use an app password with this module.

Rate Limits

The commands in this module use the public Bluesky API which means there are [rate limits](#). If you exceed the rate limit, you will get an error message. You will need to wait until the rate limit resets.



There is a rate limit of 300 new sessions per day. This shouldn't be an issue for most people unless you are testing code or running some sort of high-volume automation.



Posting

Use `New-BskyPost`, or its alias `skeet`, to post a message to Bluesky. There are parameters to include an image. If you include an image, the `New-BskyPost` command will call `Add-BskyImage` to upload the image. It is **strongly recommended** that you included ALT text for the image.

```
$param = @{
    Message    = "Getting close to sharing my #PowerShell Bluesky code."
    ImagePath  = "C:\work\MsPowerShell.jpg"
    ImageAlt    = "Ms. PowerShell"
    Verbose    = $true
}
New-BskyPost @param
```

The output is a URL to the post.

If your message contains a URL, it will be converted to a clickable link. Make sure your link is surrounded by white space. Beginning with v1.0.0, you can post Markdown style links.


```
PS C:\> $m = "Testing multiple Markdown style links from my [#PowerShell
PSBluesky module](https://github.com/jdhitsolutions/PSBluesky) which you
can find on the [PowerShell Gallery](https://www.powershellgallery.com/
packages/PSBlueSky/0.6.0)"
PS C:\> skeet $m
```



Jeff Hicks @jdhitsolutions.com · 13m

Testing multiple Markdown style links from my [#PowerShell PSBluesky module](#) which you can find on the [PowerShell Gallery](#)



Figure 4: Markdown links in a Bluesky post

This example is using the `skeet` alias for `New-BskyPost`.

Reposting and Quoting

You can repost or quote a Bluesky post with `Publish-BskyPost`. You will need the CID and AT Uri (URI) of the post you want to repost or quote. These values should be part of items you can get with commands like `Get-BskyFeed` or `Get-BskyTimeline`.

```
PS C:\> Get-BskyTimeline -Limit 1 | Select-Object *

Author       : joeydantoni.com
AuthorDisplay : Joey D'Antoni
Date         : 1/9/2025 12:11:13 PM
Text         : I would likely denormalize the data a little bit, and
              have a non-vector table with ref. data like date that
              you could filter on, before doing the heavier lifting
              of the vector query. Almost like a fact/dim join in a
              DW. learn.microsoft.com/en-us/sample... cc
              @mauridb.bsky.social for a better example
Liked        : 0
Reposted     : 0
Quoted       : 0
URL          : https://bsky.app/profile/did:plc:3dvxs53kzjez5oxcbucfsa
              cf/post/3lfd6cfv7hs2o
URI          : at://did:plc:3dvxs53kzjez5oxcbucfsacf/app.bsky.feed.post
              /3lfd6cfv7hs2o
CID          : bafyreih7yplygqrno2n3n5mhqs35fbwqovxsvc6xrictt4afzgnwrt7
              epi
```

The easiest way to repost or quote a post is to pipe the object to `Publish-BskyPost`.

```

PS C:\> $tl[-1] | Publish-BskyPost -Verbose -WhatIf
VERBOSE: [12:14:47:7437 BEGIN ] Publish-BskyPost-> Starting PSBluesky module command
VERBOSE: [12:14:47:7440 BEGIN ] Publish-BskyPost-> Running under PowerShell version 7.4.6
VERBOSE: [12:14:47:7442 BEGIN ] Publish-BskyPost-> Using PowerShell Host ConsoleHost
VERBOSE: [12:14:47:7464 BEGIN ] Publish-BskyPost-> Running under Operating System Microsoft Windows 10.0.26100
VERBOSE: [12:14:47:7466 BEGIN ] Publish-BskyPost-> Using module PSBluesky version 2.2.0
VERBOSE: [12:14:47:7470 PROCESS] Publish-BskyPost-> Reposting the message with AT Uri at://did:plc:nsy2e4vupxndly4adhszebcy/app.bsky.feed.post/3lfbhddas6s25
What if: Performing the operation "Publish to Bluesky" on target "at://did:plc:nsy2e4vupxndly4adhszebcy/app.bsky.feed.post/3lfbhddas6s25".
VERBOSE: [12:14:47:7476 END ] Publish-BskyPost-> Ending PSBluesky module command
PS C:\>

```

Figure 5: Reposting a Bluesky post

This example assumes that `$tl` is an array of objects from `Get-BskyTimeline`. If you don't specify a quote, the command will repost the original message. If you do specify a quote, the command will quote the original message.

```
$f[-4] | Publish-BskyPost -Quote "Testing quoting with a PSBluesky command"
```

The command has an alias of `Repost-BskyPost`. If you include quote text, the post will be a quote, otherwise it will show as a repost from you.

Profiles

The module has a command to retrieve a Bluesky profile.

```
Get-BskyProfile jdhitsolutions.com
```

The module uses a custom format file.

```

PS C:\> Get-PSBlueskyProfile jdhitsolutions.com

Jeff Hicks [jdhitsolutions.com]

PowerShell Author/Teacher/MVP/Guide 🍌
Prof. PowerShell Emeritus 🦉
Grizzled and grumpy IT Pro - https://jdhitsolutions.github.io/
🎵 Amateur composer - https://musescore.com/user/26698536
Wine drinker 🍷🐶 and dog lover

Created          Posts  Followers  Following  Lists
-----
5/21/2023 2:44:48 PM    533         317        172         0

```

Figure 6: A Bluesky user profile

The user's profile name should be a clickable link.

The object has been customized with aliases and script properties.

```
PS C:\> $jeff = Get-BskyProfile
PS C:\> $jeff | Select-Object *

Username      : jdhitsolutions.com
Display       : Jeff Hicks
Created       : 5/21/2023 10:44:48 AM
Description   : PowerShell Author/Teacher/MVP/Tour Guide 📖
               Prof. PowerShell Emeritus 📖
               Grizzled and grumpy IT Pro - https://jdhitsolutions.github.io/
               📖Amateur composer - https://musescore.com/user/26698536
               Wine drinker 🍷 and dog lover
Avatar        : https://cdn.bsky.app/img/avatar/plain/did:plc:ohgsqpfsbocaaxus
               xqlgfvd7/bafkreifdfahcjmytu3iw2aj2d3howu6q7twkta3h23qmlve2d2mv
               o5sily@jpeg
Posts         : 1060
Followers     : 1904
Following     : 383
Lists         : 2
URL           : https://bsky.app/profile/jdhitsolutions.com
DID           : did:plc:ohgsqpfsbocaaxusxqlgfvd7
Name          : jdhitsolutions.com
Age           : 596.06:49:09.1731955
```

Note the the Followers property may not be 100% accurate. This counter may not properly reflect accounts that have stopped following you. The way that Bluesky is handling bot accounts may also affect this count.



Followers

You can retrieve a list of your followers. You can specify a number of followers between 1 and 100. The default is 50.

```
PS C:\> Get-PSBlueskyFollowers -Limit 2

Display      Username      Description
-----
Jonathan Lank jpluk.bsky.social
Libby Brown   trubludevil.social Product manager @ SpruceID. Mom. Digital Wallet and Passl
```

Figure 7: Bluesky followers

The custom formatting includes a clickable link to the follower's profile if running in Windows Terminal or a console that supports hyperlinks.

You can pipe the follower object to `Get-BskyProfile` to retrieve more information.

```
PS C:\> $f= Get-BskyFollowers
PS C:\> $f[12] | Get-BskyProfile

Jess Pomfret [jpomfret.bsky.social]

Database Engineer with a passion for automation, proper football
and fitness. She/Her.
```

Created	Posts	Followers	Following	Lists
8/14/2023 3:58:44 PM	125	236	157	1

The default behavior is to retrieve between 1 and 100 followers. Or you can use the `-All` parameter to retrieve all followers.



Searching for Users

You can search for Bluesky users with `Find-BskyUser`. The default behavior is to search for a user by name. The search is case-insensitive.

```
Find-BskyUser -UserName "jeff h" -Limit 3
```

The default output includes clickable links to the user's profile. This object too has been customized with aliases and script properties.

```
PS C:\> Find-BskyUser -UserName "jeff h" -Limit 3 |
Select-Object *Name,Description,Created,Age

DisplayName : Jeff (no, the other one)
UserName    : jeff-notheotherone.bsky.social
Description  : When I use a word, Humpty said, it means just what I choose it to
              mean - neither more nor less.

              The question is, said Alice, whether you can make words mean so
              many different things.

              H: The question is which is to be Master - that's all.
Created      : 8/11/2023 6:27:08 AM
Age          : 514.11:04:04.2416645

DisplayName : F-0 Loignon
UserName    : foloignon.bsky.social
Description  : « Un p'tit gosseux » - MBC
              « Une caricature » - H. Buzetti
              « L'osti de tapette communiste woke » - Un pirate
```

```

        Je suis pas mal l'amalgame de tout ce que Jeff Fillion haït :
        prof, artiste, écolo, socialiste, LGBT, féministe, antiraciste,
        intello, etc.
Created      : 10/5/2023 8:09:28 PM
Age         : 458.21:21:43.9869033

DisplayName : Jeff H
UserName    : jhorowitzmd.bsky.social
Description : Division Chief of Pulm, Crit Care and Sleep Med at OSU. Lung
            fibrosis investigator. Cubs. Star Wars, GoT, LoTR, Avengers. Dad,
            husband, son and brother. RT's do not mean endorsement.
            #horoblast.
Created      : 11/13/2024 6:53:06 AM
Age         : 54.10:38:05.3790593

```

The value you specify for the user name will also search the user's description property. This is a handy way of finding users with similar interests.

```
Find-BskyUser powershell
```

```

Username: Dave Carroll [thedavecarroll.com]

Created      Description
-----
4/29/2023 6:52:59 AM #PowerShell #Developer and enthusiast with a side of
                    #DevOps, #Python, infrastructure, and #RetroComputing.
                    Lifelong #cinophile and #punster.

                    #Aspie #INTJ #9w1
                    He/Him

                    https://thedavecarroll.com

Username: Sandro [kharec.info]

Created      Description
-----
7/24/2023 3:46:02 AM F R /U S SysOps turned #DevOps 🧑🏻‍💻
                    Python/Go/C++/Perl developer 🎧🎧 JSON/YAML/HCL lover 📺
                    Bash/PowerShell eater 🐛 repressed DBA ⚙️ pop culture nerd
                    📺📺📺📺📺

Username: Frode [frodeflaten.com]

```

Figure 8: Find Bluesky users via their description

You can pipe the search results to `Get-BskyProfile` to retrieve more information.

```

PS C:\> $p = Find-BskyUser powershell -Limit 10
PS C:\> $p | Get-BskyProfile

```

```
Chrissy LeMaire [funbucket.dev]
```

```
Dual Microsoft MVP, GitHub Star, creator of dbatools, author  
http://dbatools.io/book ☐☐
```

```
Totally into PowerShell, SQL Server and AI.
```

```
☐ North of France
```

Created	Posts	Followers	Following	Lists
-----	-----	-----	-----	-----
4/24/2023 12:12:00 PM	688	1435	614	0
...				

Following

Likewise, you can get a list of all the accounts you are following.

```
PS C:\> get-bskyFollowing -limit 3
```

Display	Username	Description
-----	-----	-----
Kevin Cunningham	dolearning.dev	Developer educator Helping dev teams keep their JS/TS, Python and PHP sharp (sometimes Java and Golang too) https://kevincunningham.co.uk
Richard M. Hicks	richardhicks.com	📍 Northern Ireland
Last Week Tonight with John Oliver UPDATE	lastweektonight1.bsky.social	/ video / info / resources / Last Week Tonight with John Oliver ~ unaffiliated

Figure 9: Bluesky following

As with followers, the default behavior is to retrieve between 1 and 100 accounts you are following. Or you can use the `-All` parameter to retrieve all accounts you are following.

Starting with version 2.3.0, you can follow and unfollow Bluesky users with [New-BskyFollow](#) and [Remove-BskyFollow](#). These commands can also be referenced by their aliases `Follow-BskyUser` and `Unfollow-BskyUser`.

To follow, specify the account's user name or handle.

```
PS C:\> Follow-BskyUser andrewpla.tech  
https://bsky.app/profile/did:plc:xrspiwserax6shgskcj7grgg
```

The output is a clickable link to the followed user's profile.

To unfollow, you can pipe a user object to `Remove-BskyFollow`

```
PS C:\> Remove-BskyFollow andrewpla.tech -WhatIf
What if: Performing the operation "Remove-BskyFollow" on target "andrewpla.tech".
```

Or you can pipe a user object to Remove-BskyFollow.

```
PS C:\> Get-BskyProfile andrewpla.tech | Remove-BskyFollow -Passthru

cid                                     rev
---                                     ---
bafyreibe4a3ewrh2oty2jercit75q42wdnuuip22b3tq6jybwamdz7daq4 3lfnjoouju62k
```

The followed user will receive a notification when you follow them but not when you unfollow them.

Blocking

You can block a Bluesky user with Block-BskyUser. You will need the user's user name.

```
PS C:\> Block-BskyUser -UserName "PooBar1234.bsky.social"
```

You can also pipe follower or profile objects to the command.

```
PS C:\> $f[-1] | Block-BskyUser -WhatIf
What if: Performing the operation "Block-BskyUser" on target "thedavecarroll.com [did:plc:rlwd5iajr3]
PS C:\> Get-BskyProfile thedavecarroll.com | Block-BskyUser
```

Use Get-BskyBlockedUser to see a formatted display.

```
PS C:\> Get-BskyBlockedUser
```

Display	Username	Labels	Blocked	Description
Dave Carroll	thedavecarroll.com		1/21/2025 9:42:30 AM	#PowerShell #Developer and enthusiast with a side of #DevOps, #Python, infrastructure, and #RetroComputing. Lifelong #cinophile and #punster. #Aspie #INTJ #9w1 He/Him

Figure 10: Blocked Bluesky users

To unblock is just as easy.

```
Unblock-BskyUser thedavecarroll.com
```

If you subscribe to any block lists, use Get-BskyBlockedList to see the list.

```
PS C:\> Get-BskyBlockedList

Name       : Scammers \(catfish, crypto, etc.\)
Purpose    : Moderated list
Labels     :
Items      : 33
CreatedBy  : mkaney.bsky.social
Indexed    : 12/2/2024 9:50:00 PM
Description : catfish, crypto, etc.
```

Figure 11: Get Bluesky blocked list

The formatted output includes hyperlinks.

Feed

Use Get-BskyFeed to retrieve the latest posts from *your* feed. You can query for 1 to 100. The default output uses a custom format file. The current behavior is to get posts and replies.

```
PS C:\> Get-PSBlueskyFeed -Limit 3

Jeff Hicks [jdhitsolutions.com]
Getting close to pushing an update to my #PoewrShell bluesky module

Date                Liked Replied Reposted Quoted
-----
10/27/2024 1:43:06 PM      0      0      0      0

Jeff Hicks [jdhitsolutions.com]
Adding help documentation to my Bluesky #PowerShell module.

Date                Liked Replied Reposted Quoted
-----
10/27/2024 1:12:18 PM      1      0      0      0

Jeff Hicks [jdhitsolutions.com]
I'm using the #PowerShell InformationVariable to store the raw response from the Bluesky API.

Date                Liked Replied Reposted Quoted
-----
10/27/2024 12:05:15 PM      1      0      0      0
```

Figure 12: Getting your Bluesky feed



The formatted output includes clickable links to the the author, which might be different than you if reposting, and the post. For best results, run PowerShell 7 in Windows Terminal.

Notifications

You can retrieve your notifications with `Get-BskyNotification`. You can specify a limit of 1 to 100. The default is 50.

```
Get-BskyNotification -limit 10
```

The default formatted output includes clickable links to the author and the liked or reposted skeet.

-  Like
-  Follow
-  Repost
-  Reply



Date	Account	Subject
11/13/2024 1:36 PM	mscommunity.bsky.social	
11/13/2024 1:24 PM	Oscar	
11/13/2024 1:18 PM	dirteam.bsky.social	
11/13/2024 1:04 PM	Hoang Chu	
11/13/2024 1:01 PM	 Madhu PERERA	<u>I have published v1.2.0 of the PSBluesky module to the #PowerShell Gallery. Run Open-BskyHelp to read the help PDF</u>
	<div>Like</div>	
	<div>follow</div>	
11/13/2024 12:54 PM	medford	
11/13/2024 12:48 PM	tysonro.bsky.social	
11/13/2024 12:17 PM	sqljuju.bsky.social	
11/13/2024 11:47 AM	 Mark Oldham	<u>I'm on track to double my follower count in one week. I don't think I ever had this type of engagement in the upside down. Thank you.</u>
	<div>links</div>	
11/13/2024 11:38 AM	martyntolley.bsky.social	

Figure 13: Getting your Bluesky notifications

Long account names will be truncated in the formatted output.

Timeline

If you want to view items from your timeline, use the `Get-BskyTimeLine` command.

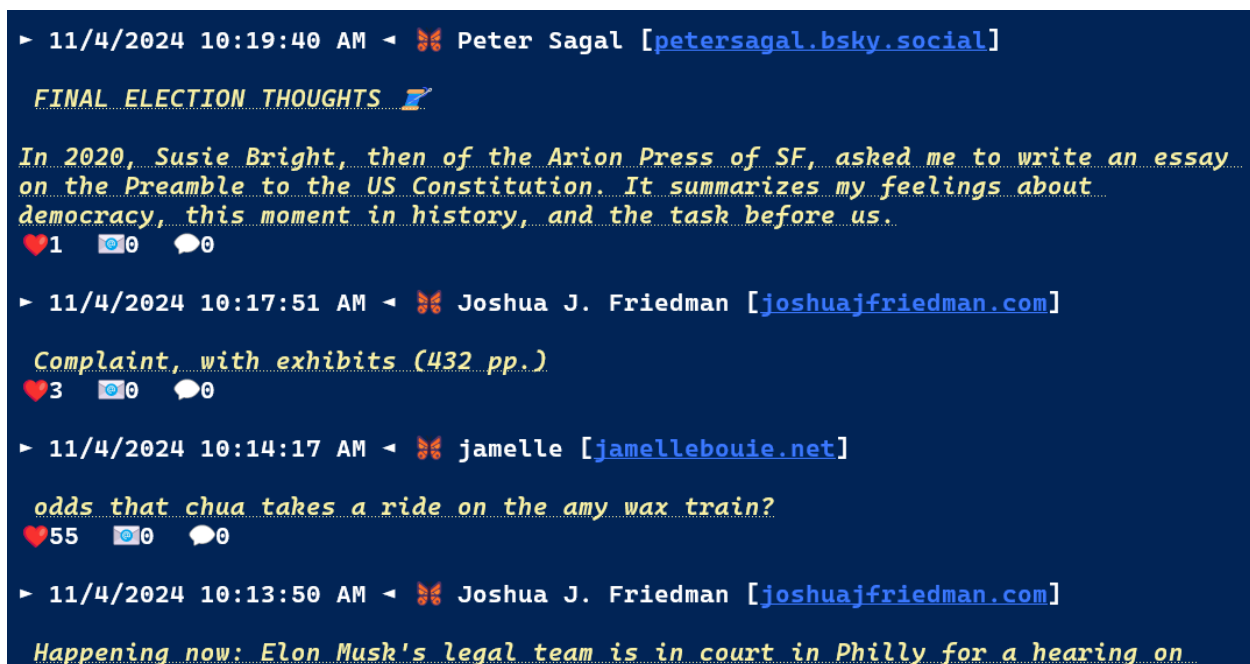


Figure 14: Getting your Bluesky timeline

You can specify a limit of 1 to 100. The default is 50.

```
Get-BskyTimeline -Limit 25
```

The command uses a custom format file. The default formatted output includes clickable links to the author and the post.

API Logging

Beginning with v2.4.0, you can enable API logging. This will allow you to keep track of what API endpoints you are using. You might want to know this information to gauge your usage against API rate limits. Logging is disabled by default when you import the module.

Run `Get-BskyLogging` to see the current logging status.

```
PS C:\> Get-BskyLogging

LoggingEnabled LogFile                               LogFileSize
-----
False C:\Users\Jeff\AppData\Local\Temp\bskyAPILogging.json 0
```

To enable logging, run `Enable-BskyLogging`.

```
PS C:\> Enable-BskyLogging -Passthru | Select LoggingEnabled

LoggingEnabled
-----
```

True

When an Bluesky API request is made, it will be logged to a JSON file.

```
PS C:\> Get-Content (Get-BskyLogging).LogFile
[
  {
    "Date": "2025-01-21T08:54:05.4764747-05:00",
    "Uri": "https://bsky.social/xrpc/com.atproto.server.createSession",
    "Endpoint": "com.atproto.server.createSession",
    "Name": "createSession",
    "Command": "_CreateSession"
  },
  {
    "Date": "2025-01-21T08:54:55.3206374-05:00",
    "Uri": "https://bsky.social/xrpc/app.bsky.actor.getProfile?actor=
jdhitsolutions.com",
    "Endpoint": "app.bsky.actor.getProfile",
    "Name": "getProfile",
    "Command": "Get-BskyProfile"
  }
]
PS C:\> Get-Content (Get-BskyLogging).LogFile | ConvertFrom-Json

Date      : 1/21/2025 8:54:05 AM
Uri       : https://bsky.social/xrpc/com.atproto.server.createSession
Endpoint  : com.atproto.server.createSession
Name      : createSession
Command   : _CreateSession

Date      : 1/21/2025 8:54:55 AM
Uri       : https://bsky.social/xrpc/app.bsky.actor.getProfile?actor=
jdhitsolutions.com
Endpoint  : app.bsky.actor.getProfile
Name      : getProfile
Command   : Get-BskyProfile
```

It should be easier to work with structured data to analyze usage patterns. The default log is in your %TEMP% directory. You can change the log file location Set-BskyLogging.

```
PS C:\> Set-BskyLogging c:\temp\api.json -WhatIf
What if: Performing the operation "Set-BskyLogging" on target "c:\temp\api.json".
```

You can disable logging with Disable-BskyLogging.

```
PS C:\> Disable-BskyLogging
```

If you want to clear the logging file or start over, you can easily delete it.

```
PS C:\> Remove-BskyLogging -WhatIf
What if: Performing the operation "Remove File" on target "C:\Users\Jeff\
AppData\Local\Temp\bskyAPILogging.json".
```

Changes you make to logging will persist for the duration of your PowerShell session. The next time you import the module, the default settings will be used and you will need to re-enable logging and/or change the log file location.

The assumption is that logging is a temporary feature to help you troubleshoot or monitor your API usage and thus logging settings are not stored as module preferences.

Logging information is stored in the `BskySession` object, but you should use the module functions to manage it.

Ad-Hoc Scripting

You can use the PSBluesky module in ad-hoc scripting projects using the [Bluesky API](#). One common requirement is to provide an authentication header. Since you can use the PSBluesky module to create an authenticated session, you can create a header from this information.

The session object has a script method you can use to create a header.

```
PS C:\> $headers = (Get-BskySession).CreateHeader()
PS C:\> $headers
```

Name	Value
Content-Type	application/json
Authorization	Bearer eyJ0eXAiOiJhdCtqd ...

During your scripting, if you get an error about an expired token, recreate the headers variable.

Say you want to use the [actor-get-preferences](#) API. You can define the uri according to the documentation.

```
$PDShost = "https://bsky.social"
$uri = "$PDShost/xrpc/app.bsky.actor.getPreferences"
```

Then you can use the `Invoke-RestMethod` cmdlet to make the API call.

```
$r = Invoke-RestMethod -Uri $uri -Method Get -Headers $headers
```

And then work with results.

```
PS C:\> $r.preferences
```

hideReplies	: False
hideRepliesByUnfollowed	: False
hideRepliesByLikeCount	: 2
hideReposts	: False
hideQuotePosts	: False

```

$type           : app.bsky.actor.defs#feedViewPref
feed            : home

nuxs : {@{id=TenMillionDialog; completed=True}, @{id=NeueTypography; completed
      =True}}
$type : app.bsky.actor.defs#bskyAppStatePref
...

PS C:\> $r.preferences[0] | Select hide*

hideReplies           : False
hideRepliesByUnfollowed : False
hideRepliesByLikeCount : 2
hideReposts           : False
hideQuotePosts        : False

```

Here's another example using the [get-known-followers](#) API.

```

PS C:\> $uri = "$PDSHost/xrpc/app.bsky.graph.getKnownFollowers?actor=
jdhitsolutions.com&limit=100"
PS C:\> $r = Invoke-RestMethod -Uri $uri -Headers $headers
PS C:\> $r.followers.count
100
PS C:\> $r.followers[-1]

did           : did:plc:h6cd7ej2sl4hswxfyjp6b4mx
handle        : harjit.bsky.social
displayName    : Harjit Dhaliwal [Hoorge]
avatar        : https://cdn.bsky.app/img/avatar/plain/did:plc:h6cd7ej2sl4hswxfyjp6b4mx/bafkreibb56a7ywrmmwqcfn5wj42a473wj375c4l33526xiitch2td3an4km@jpeg
associated    : @{chat=}
viewer        : @{muted=False; blockedBy=False;
                  following=at://did:plc:ohgsqpfsbocaaxusxqlgfv7/app.bsky.grap ...
                  followedBy=at://did:plc:h6cd7ej2sl4hswxfyjp6b4mx/app.bsky.gra ...
labels        : {}
createdAt     : 5/1/2023 4:35:34 AM
description    : Senior Product Marketing Manager for Cloud Endpoints at Microsoft,
                  Ex-Microsoft MVP, IT Pro, Social Media Strategist, Photographer,
                  International Speaker, Mentor.

                  https://about.me/HarjitDhaliwal
                  https://aka.ms/TacklingTech

                  #Windows #Microsoft #Copilot
indexedAt     : 11/29/2024 1:28:38 AM

```



Information and Troubleshooting

The commands in this module should write the raw response from the API request to the Information stream. Some commands might include additional information.

```

PS C:\> $me = Get-PSBlueskyProfile jdhitsolutions.com -InformationVariable v
PS C:\> $v | select *

MessageData      : @{did=did:plc:ohgsqpfsbocaaxusxqlgfvd7; handle=jdhitsolutions.com; displayName=Jeff;
                    avatar=https://cdn.bsky.app/img/avatar/plain/did:plc:ohgsqpfsbocaaxusxqlgfvd7/bafk
                    associated=; viewer=; labels=System.Object[]; createdAt=5/21/2023 2:44:48 PM; desc
                    Prof. PowerShell Emeritus 🦉
                    Grizzled and grumpy IT Pro - https://jdhitsolutions.github.io/
                    🎵Amateur composer - https://musescore.com/user/26698536
                    Wine drinker 🍷🐕 and dog lover; indexedAt=10/26/2024 3:45:40 PM;
                    banner=https://cdn.bsky.app/img/banner/plain/did:plc:ohgsqpfsbocaaxusxqlgfvd7/bafk
                    followersCount=322; followsCount=177; postsCount=543}
Source           : C:\Scripts\PSBluesky\functions\Get-PSBlueSkyProfile.ps1
TimeGenerated    : 10/27/2024 12:02:44 PM
Tags             : {raw}
User             : THINKX1-JH\Jeff
Computer        : ThinkX1-JH
ProcessId       : 20632
NativeThreadId   : 20936
ManagedThreadId : 17

```

Figure 15: Information stream

The output will be an object.

```

PS C:\> $v.MessageData | Select-Object did,handle,*count

did           : did:plc:ohgsqpfsbocaaxusxqlgfvd7
handle        : jdhitsolutions.com
followersCount : 322
followsCount   : 177
postsCount    : 543

```

You might want to set a default parameter value.

```
$PSDefaultParameterValues['*-Sky*:InformationVariable'] = "iv"
```

Other Module Features

Custom Formatting

Many of the commands in this module use custom formatting files to provide a more visually appealing output. The module uses emojis and ANSI formatting. The formatting is designed for Windows Terminal or a console that supports ANSI formatting and emojis.

Some commands will also have alternative format views.

```
PS C:\> $tl | sort liked -desc | select -first 10 | format-table -view liked
```

Liked	Date	Author	Text
690	1/9/2025 11:32:07 AM	Know Your Meme [knowyourmeme.com]	On this day 12 years ago, @kcg.bsky.social posted the "This is Fine" comic.
668	1/9/2025 12:35:47 PM	Aaron Rupar [atrupar.com]	Some of the responses we've seen to the CA fires illustrate the same basic political dynamic that helped Trump win last year. It's a given Republicans will crap on Dems when things go wrong. But it's also a given that the far left does it too, often in equally misleading ways.
397	1/9/2025 12:34:56 PM	NPR [npr.org]	Californians are all too familiar with flip-flopping weather extremes, and new research suggests that weather whiplash is increasing as the atmosphere warms due to human-caused climate change. via @kqednews.bsky.social
309	1/9/2025 12:34:10 PM	David Roberts [volts.wtf]	To an asshole, all virtue is "virtue signaling."
230	1/9/2025 12:42:44 PM	Aaron Rupar [atrupar.com]	People are pressing me for an example, so here you go. Far left social media ran wild yesterday with the narrative that Karen Bass cut the LA fire department. That's misleading at best, but is the exact sort of talking point that gets heavy play in rightwing media.
208	1/9/2025 8:02:54 AM	Steve Kenson [stevekenson.com]	Welcome to Bluesky! A few tips:

Figure 16: Liked custom view

Command	Format	ViewName
Get-BskyLiked	Table	Liked
Get-BskyTimeline	Table	Liked
Get-BskyTimeline	Table	Default
Get-BskyFeed	Table	Liked
Get-BskyModuleInfo	List	Default

Type Extensions

You are encourage to pipe command results to Get-Member to discover additional properties you might find useful.

```
PS C:\> Get-BskyProfile thedavecarroll.com |
Format-List Name,Description,Created,Age

Name           : thedavecarroll.com
Description    : #PowerShell #Developer and enthusiast with a side of #DevOps,
                #Python, infrastructure, and
                #RetroComputing. Lifelong #cinophile and #punster.

                #Aspie #INTJ #9w1
                He/Him

                https://thedavecarroll.com
Created        : 4/29/2023 6:52:59 AM
Age            : 633.07:27:20.9929789
```

Custom Verbose Messaging

This module uses ANSI formatting with localized string data and customized verbose messaging.

```
PS C:\> $f = Get-BskyFeed -Limit 10 -Verbose
VERBOSE: [10:31:47:2888 BEGIN ] Get-BskyFeed-> Starting PSBluesky module command
VERBOSE: [10:31:47:2891 BEGIN ] Get-BskyFeed-> Running under PowerShell version 7.4.6
VERBOSE: [10:31:47:2892 BEGIN ] Get-BskyFeed-> Using PowerShell Host ConsoleHost
VERBOSE: [10:31:47:2894 BEGIN ] Get-BskyFeed-> Running under Operating System Microsoft Windows 10.0.26100
VERBOSE: [10:31:47:2895 BEGIN ] Get-BskyFeed-> Using module PSBluesky version 2.0.0
VERBOSE: [10:31:47:2898 PROCESS] Get-BskyFeed-> Querying 10 feed items for jdhitsolutions.com
VERBOSE: [10:31:47:2899 PROCESS] Get-BskyFeed-> https://bsky.social/xrpc/app.bsky.feed.getAuthorFeed?actor=did:plc:ohg
sqpfsbocaaxusxqlgfvd7&limit=10&filter=posts_with_replies
VERBOSE: Requested HTTP/1.1 GET with 0-byte payload
VERBOSE: Received HTTP/1.1 response of content type application/json of unknown size
VERBOSE: Content encoding: utf-8
VERBOSE: [10:31:47:8473 END ] Get-BskyFeed-> Ending PSBluesky module command
PS C:\>
```

Figure 17: Custom verbose messaging

The color formatting the command names is not user-configurable at this time.

I am hoping to add localized strings for other languages as well as localizations to command help and the help PDF file.

Preferences

The module features that rely on custom formatting and ANSI colorization may not be compatible in all console environments or for all users. You may need to adjust settings to improve the display. All formatting preferences are stored in an exported hash table variable called `bskyPreferences`. While you can modify this variable directly, it is recommended that you use the related module commands to manage it.

Run [Get-BskyPreference](#) to see your current settings.


```
PS C:\> Get-BskyPreference

Setting                Style
-----
Username               `e[4;38;5;190m
ProfileDescription     `e[3;96m
Message                `e[3;93m
BlockedList            `e[4;92m
IsRepost               `e[36m
LikedLow               `e[38;5;48m
LikedMedium            `e[38;5;214m
LikedHigh              `e[38;5;105m
LikedHighest           `e[38;5;201m
InfoAlias              `e[3;38;5;220m
ActiveSession          `e[92m
InactiveSession        `e[91m
Add-BskyImage          `e[1;38;5;122m
Find-BskyUser          `e[1;38;5;111m
Get-BskyFeed           `e[1;96m
Get-BskyBlockedList    `e[1;38;5;165m
Get-BskyBlockedUser    `e[1;38;5;168m
Get-BskyFollowers      `e[1;38;5;182m
Get-BskyFollowing      `e[1;38;5;208m
Get-BskyLiked          `e[1;38;5;201m
Get-BskyNotification   `e[1;38;5;195m
Get-BskyProfile        `e[1;38;5;214m
Get-BskySession        `e[1;38;5;197m
New-BskyFollow         `e[1;38;5;147m
New-BskyPost           `e[1;38;5;159m
Publish-BskyPost       `e[1;38;5;180m
Remove-BskyFollow      `e[1;95m
Start-BskySession      `e[1;38;5;228m
DefaultCommand         `e[1;38;5;51m
```

Figure 18: Bluesky preferences

You can use [Set-BskyPreference](#) to change them.

```
Set-BskyPreference Username -style "`e[3;38;5;135m"
```

You can use any ANSI sequence or `$PSStyle` value. Any changes you make will only persist for the duration of your PowerShell session or until you remove the PSBluesky module. The next time you import the module, the default preferences will be used. To save your changes, run [Export-BskyPreference](#) to save your preferences to a JSON file under `$HOME`.

```
PS C:\> Export-BskyPreference -Passthru

Directory: C:\Users\Jeff

Mode                LastWriteTime         Length Name
----                -
-a---             1/13/2025   9:41 AM           1091 .psbluesky-preferences.json
```

The next time you import the module, if this file is present, the module will import it and apply your preferences.

To remove your preferences and restore the module defaults, you can use [Remove-BskyPreferenceFile](#) and then re-import the module. If you decide to uninstall the module, and have exported preferences, you should run this command before uninstalling. Otherwise, you will need to manually delete the file.

Roadmap

I have a short list of items on my wish list:

- Add support for posting video
- Add commands to work with direct messages, aka chat
- Maybe create a TUI-base reader for your timeline

If you are testing the module and think you've found a bug, please post an [Issue](#). For all other topics and questions, including feature requests, please use the repository's [Discussions](#) section. I am open to pull requests if you have a feature you'd like to add. I especially would love to see the Pester tests expanded and the pending tests completed.

Note: You can find me on Bluesky as jdhitsolutions.com.