PSBlueSky Help[1]

https://jdhitsolutions.github.io

v2.2.0

Contents

PSBluesky

This module is a set of PowerShell functions designed to let you interact with Bluesky API from PowerShell. 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 modules or related modules to securely store your Bluesky credentials:

- Microsoft.PowerShell.SecretManagement

- Microsoft.PowerShell.SecretStore

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

- Add-BskyImage
- Find-BskyUser
- Get-BskyAccountDID
- Get-BskyBlockedList
- Get-BskyBlockedUser
- Get-BskyFeed
- Get-BskyFollowers
- Get-BskyFollowing
- Get-BskyLiked
- Get-BskyModuleInfo
- Get-BskyNotification
- Get-BskyProfile
- Get-BskySession
- Get-BskyTimeline
- New-BskyPost
- Open-BskyHelp
- Publish-BskyPost
- Start-BskySession
- Update-BskySession

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. Or you can use the `-AsMarkdown` parameter to read this file using markdown formatting. Not all markdown features may properly render in the console.

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 [v2.2.0]


Name                  Alias          Synopsis
----                  -----          --------
Add-BskyImage                        Upload an image to Bluesky.
Find-BskyUser         bsu            Search for Bluesky user acco...
Get-BskyAccountDID                   Resolve a Bluesky account na...
Get-BskyBlockedList   bsblocklist    Get your subscribed blocked ...
Get-BskyBlockedUser   bsblock        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 account ...
Get-BskyLiked                        Get your liked Bluesky posts.
```

```
Get-BskyModuleInfo                         Get a summary of the PSBlueSk ...
Get-BskyNotification bsn                   Get Bluesky notifications.
Get-BskyProfile      bsp                   Get a Bluesky profile.
Get-BskySession      bss                   Show your current Bluesky ses ...
Get-BskyTimeline     bst                   Get your Bluesky timeline.
New-BskyPost         skeet                 Create a Bluesky post.
Open-BskyHelp        bshelp                Open the PSBluesky help docum ...
Publish-BskyPost     Repost-BskyPost       Repost or quote a Bluesky post
Start-BSkySession                          Start a new Bluesky session.
Update-BskySession   Refresh-BskySession Refresh the Bluesky session t ...
```

Authentication

Session and Tokens

🏛 In order to send data, you must authenticate. Version 2.0.0 of this module introduces 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 should 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 1.2.0, the module will refresh the token every 15 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 use 2FA at this time. 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`.
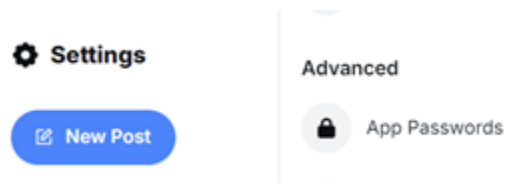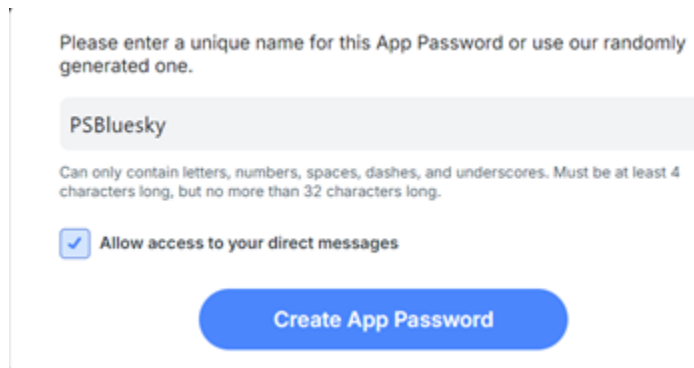

Figure 1: Bluesky app passwords

You will need to create a new app password. Give it a meaningful name.
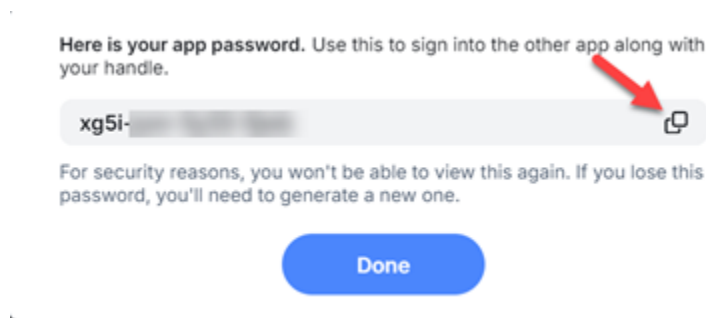
Figure 2: Bluesky creating an app password

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

♡ 1

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`.

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"
```

Profiles

The module has a command to retrieve a Bluesky profile.

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

The module uses a custom format file.

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
```

👫 Followers

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

```
Get-BskyFollowers -Limit 2
```

```
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 Passk
```

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.

10

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-O 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
```


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.

```
Get-BskyFollowing -limit 3
```


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.

Feed 🗞️

Use `Get-BskyFeed` to retrieve the latest posts from *your* feed. You can query for 1 to 100.

```
Get-BskyFeed -Limit 3
```

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 documenation 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 10: 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

Figure 11: Getting your Bluesky notifications

Long account names will be truncated in the formatted output.

Timeline 📅

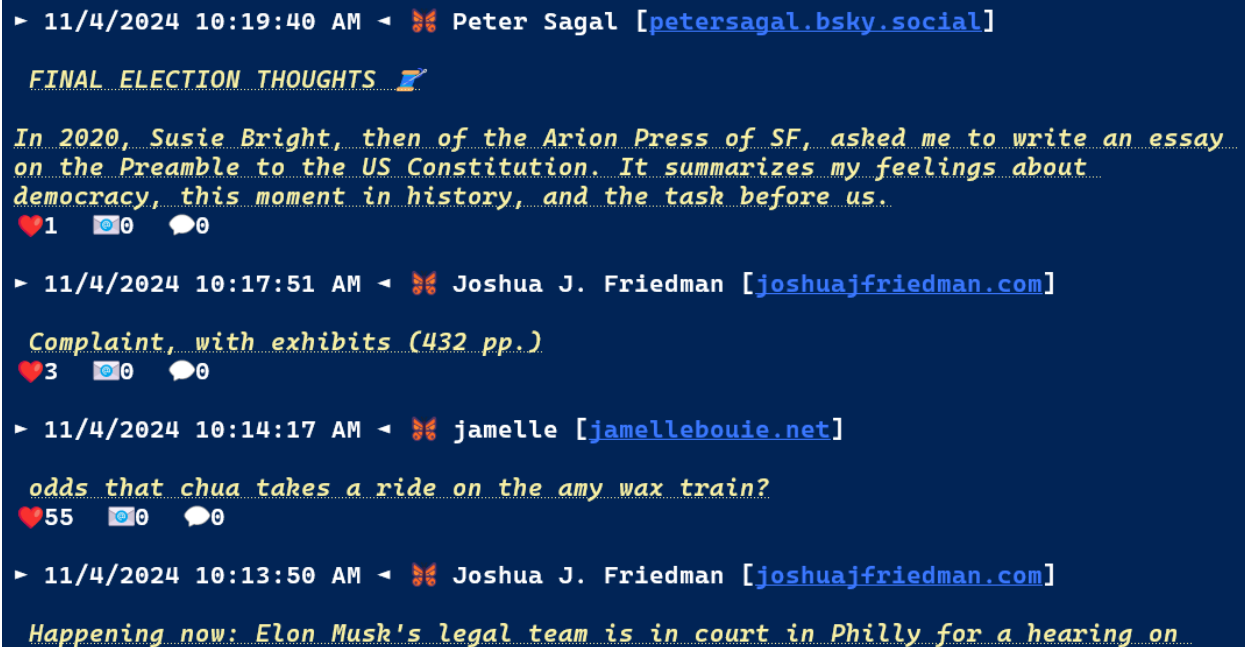If you want to view items from your timeline, use the `Get-BskyTimeline` command.

Figure 12: 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.

ℹ️ 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.

Figure 13: 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.

Figure 14: 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 #cinephile and #punster.


              #Aspie #INTJ #9w1
              He/Him


              https://thedavecarroll.com
```

```
Created        : 4/29/2023 6:52:59 AM
Age            : 569.04:06:51.6915207
```

Custom Verbose Messaging

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

Figure 15: 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.

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.