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 atproto protocols. The module is written for PowerShell 7, although it might work as written in Windows PowerShell with minimal changes. Commands have not been tested for cross-platform compatibility.

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
- Get-BskyAccessToken
- Get-BskyFeed
- Get-BskyFollowers
- Get-BskyFollowing
- Get-BskyProfile
- Get-BskySession
- Get-BskyTimeline
- New-BskyPost
- Open-BskyHelp

Authentication

In order to send data, you must authenticate. The `Get-BskyAccessToken` function will retrieve an access token. You shouldn't need to call this command directly. The other module commands will call it and pass the authentication token as needed. Technically, the token has a time limit and it could be re-used. But it is just as easy to get a token with each request since it is assumed you will be using the module commands intermittently.

You will need to create a PSCredential object with your Bluesky username and password. 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
```

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. The module attempts to refresh and re-use Bluesky sessions. You can run `Get-BskySession` to see your current session information.

⚠️ There is a rate limit of 300 new sessions per day. If you reload the module you will end up creating a new session which could affect your rate limit.

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
```

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

Profiles

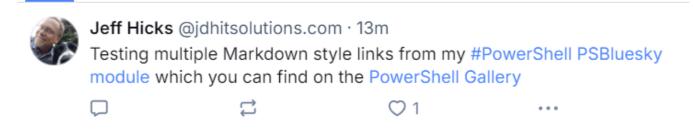The module has a command to retrieve a Bluesky profile.

Figure 1: Markdown links in a Bluesky post

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

The module uses a custom format file.

Figure 2: A Bluesky profile

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

👫 Followers

You can retrieve a list of your followers

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


Figure 3: 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
```

You can retrieve between 1 and 100 followers. I don't know if there is a way to enumerate or page through all followers.

Feed 📰

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

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

The object output has a custom format file.

```
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
```
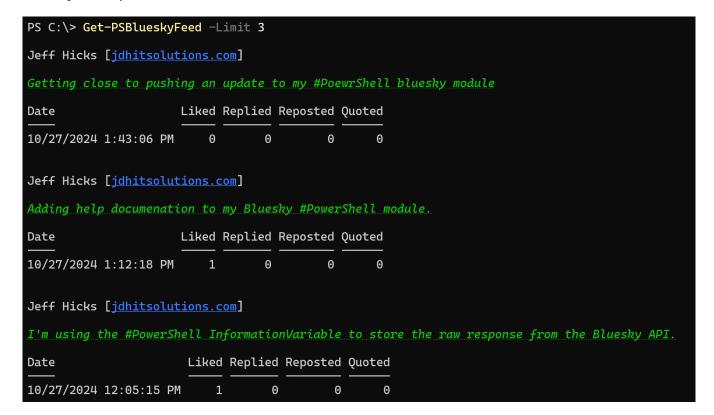
Figure 4: Bluesky feed

The output includes clickable links to the the author, which might be different than you if reposting,

and the post.

The current behavior is to get posts and replies.

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.

```
![Information stream](images/bsky-information.png)

The output will be an object.

```powershell
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"
```

Roadmap 🗺️

I have a short list of items to finish before this can be published to the PowerShell Gallery.

- support posting multiple images
- localized verbose and other messaging
- 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, please use the Discussions feature.