

Mastodon Domain Knowledge

This document contains information about Mastodon that we encountered via trial and error

Accounts on Server X can only see posts on servers federated with Server X

This is the crux of the project. If one wants to view posts on unfederated servers, one will have to create a new account on those servers.

Clients (Bots) only work for servers that they were created on

Clients are created via creating an application on a server ([Obtaining client app access - Mastodon documentation \(joinmastodon.org\)](#)). If you create a client on server X, it can *only* operate with Server X via API. So, if you want to interact via API with 100 servers, you need to create 100 client applications, one for each server. This can thankfully be done automatically.

Don't get confused by the types of access tokens

There are 3 types of access tokens:

- 1) Application Access Token: The access token (alongside the ID and secret) used to identify and authorize your application to make API calls via generating a *Client (for the application)*. Think of it as the access token for your bot.
- 2) User Authorization Code: This is *not* used to generate a client. Rather, this is the code that a server generates when the user authorizes your application to access their account. *This code is one-time use only.*
- 3) User Access Token: With the user authorization code, you can use your *Application client* (see above) to generate a *User Access* token. Save this token. With this token, you can generate another *client* (for the user) that will be able to operate on behalf of the user (pulling timeline data etc.).

Not all Mastodon servers respond equally to API calls

Some servers return a correct json only when the correct header is given. It would be prudent to have the following as part of the header for every request:

```
headers = {  
    "User-Agent": "YourApp/1.0",  
    "Accept": "application/json",  
}
```

Some servers return incorrect data. For instance, social.vivaldi.net is a Mastodon server, but in the “domain” field of the data returned when querying the `api/v2/instance` endpoint is “vivaldi.net”, which is inaccurate (api calls need to be made to social.vivaldi.net, and not vivaldi.net). Developers beware!