

Search...

FAQ

Getting Started ▾

Authentication

Quick Start

Introduction >

TL >

Authentication

Signing in

Table of Contents

- [Getting API ID and API HASH](#)
- [Logging in as a Bot](#)
- [Logging in as a User](#)
- [Using MTProxies and Socks5 Proxies.](#)
- [Persistent Session](#)
- [String Session](#)
- [Store Session](#)

Getting API ID and API HASH

Before working with Telegram's API, you need to get your own API ID and hash:

1. [Login to your Telegram account](#) with the phone number of the developer account to use.
2. Click under API Development tools.
3. A *Create new application* window will appear. Fill in your application details. There is no need to enter any *URL*, and only the first two fields (*App title* and *Short name*) can currently be changed later.
4. Click on *Create application* at the end. Remember that your **API hash is secret** and Telegram won't let you revoke it. Don't post it anywhere!

Logging in as a Bot

Using GramJS you can use a bot token to log in. Doing this is simple

JavaScript

TypeScript

```
const { TelegramClient } = require('gramjs')
const { StringSession } = require('gramjs')

const stringSession = ""; // put your string session here
const BOT_TOKEN = ""; // put your bot token here

(async () => {
  const client = new TelegramClient(
    new StringSession(stringSession),
    apiId,
    apiHash,
    { connectionRetries: 5 }
  );
  await client.start({
    botAuthToken: BOT_TOKEN,
  });
  console.log(client.sessionId);
})();
```

Underneath this is just calling the RAW function `ImportBotAuthorization`. you can leave the string session empty for now.

Logging in as a User

Logging in as a user is a bit more complex because you'll need to provide callbacks for when you receive the code from telegram. you can use the [input](#) package to manage that on Node or [prompt](#) on the browser