



PENCILDATA

# API Quickstart



PENCILDATA

This quickstart covers the four easy steps to  
using PencilDATA:

Hash, Authenticate, Register, Verify

# Step 1: The Hash

Create a digital fingerprint (“hash value”) of your file or data. We have sample code for the top 10 programming languages and platforms.

## Linux Example:

```
$ sha256sum sample-  
document.pdf
```

## Javascript Example:

```
crypto.createHmac('sha256',  
fileContents)
```

## Java Example:

```
hashString(fileContents, "SHA-2  
56");
```

The hash value generated for that specific file would be the same on every platform:

(example) ba3f474830169ddaece741cbbfbec13086139d88907104564b765c3935109d64

# Step 2: Authenticate

Submit your username and password to the tokens API. JSON will be returned with multiple values. You will need to extract the **accessToken** value.

## Example Request:

```
curl -s -X POST https://api.pencildata.com/token -H 'Content-Type: application/json' -d '{"username": "your-username", "password": "your-password"}'
```

## Example Response:

```
{"expiresIn": "3600",  
  "accessToken": "eyJraWQiOiJlTzRxQWFGb...",  
  "refreshToken": "...",  
}
```

# Step 3: Register

Submit your `accessToken` and hash to the register API. If successful, it returns an “entityId” unique to that registration and hash value. Save this `entityId`, as it will be used at a later time to verify the file.

The *storage* setting in our API can be *public* (uses Ethereum) or *private* (sawtooth/hyperledger). Private is faster and recommended for testing.

## Example Request:

```
curl -s -X POST https://api.pencildata.com/register/ -H 'Content-Type: application/json' -d '{"hash": "your-hash-value", "storage": "private"}' -H 'Authorization: Bearer your-accessToken-value'
```

## Example Response (the “entityId” you will save for later verification):

1526398776829

# Step 4: Verify

To authenticate and verify your file in the future, first create a fresh hash, using the same hashing formula from Step 1. Do not reuse the 'original' hash you used at registration, as that defeats the purpose.

Submit the fresh hash, the entityId you got from registration, and your accessToken to the verify API. It will return 'true' if your new hash matches what was originally submitted to us, or false to indicate the hash value does not match and something in your file or data has changed.

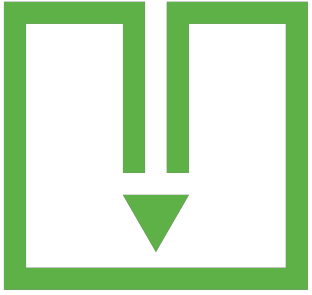
## Example Request:

```
curl -s -X GET 'https://api.pencildata.com/verify/your-entityId?hash=your-freshly-generated-hash-value&storage=private' -H 'Authorization: Bearer your-accessToken-value'
```

## Example Response (assuming your fresh hash matches the original for that entityId):

true

Note: You can test verification by altering the original file, hashing the altered version, and submitting that to the verify API with the original entityId. It will return false, since the hash of the altered file does not match what was originally registered with us.



PENCILDATA

Need help?

[support@pencildata.com](mailto:support@pencildata.com)