@maryemhadjwannes

Store & Access Files on IPFS

Start Tutorial

IPFS (InterPlanetary File System)

@maryemhadjwannes

What is IPFS?

IPFS (InterPlanetary File System) is a decentralized file storage system that allows for distributed storage of data across a peer-to-peer network.

Advantages of IPFS over Blockchain:

1. **Cost-Efficient:** **Storing large files on blockchain is expensive (gas fees). IPFS is a cheaper alternative.**
2. **Scalability:** **IPFS handles large files (images, videos) better than blockchain.**
3. **Decentralized:** **Like blockchain, IPFS is decentralized, ensuring data availability.**
4. **Faster:** **IPFS retrieves files from a distributed network, making it quicker than blockchain.**

When to Use IPFS?

Use IPFS for storing large files and application data linked to blockchain for verification.

Pinata API

@maryemhadjwannes

What is Pinata?

A platform to simplify working with IPFS: upload, pin, and manage files on IPFS.

Key Features:

1. **Pin JSON & Files:** **Upload data and files to IPFS with simple API calls.**
2. **Secure & Reliable:** **Use API keys for security and automatic pinning to ensure file availability.**
3. **Easy Integration:** **Fast, scalable, and easy-to-use APIs for developers.**

Why Pinata?

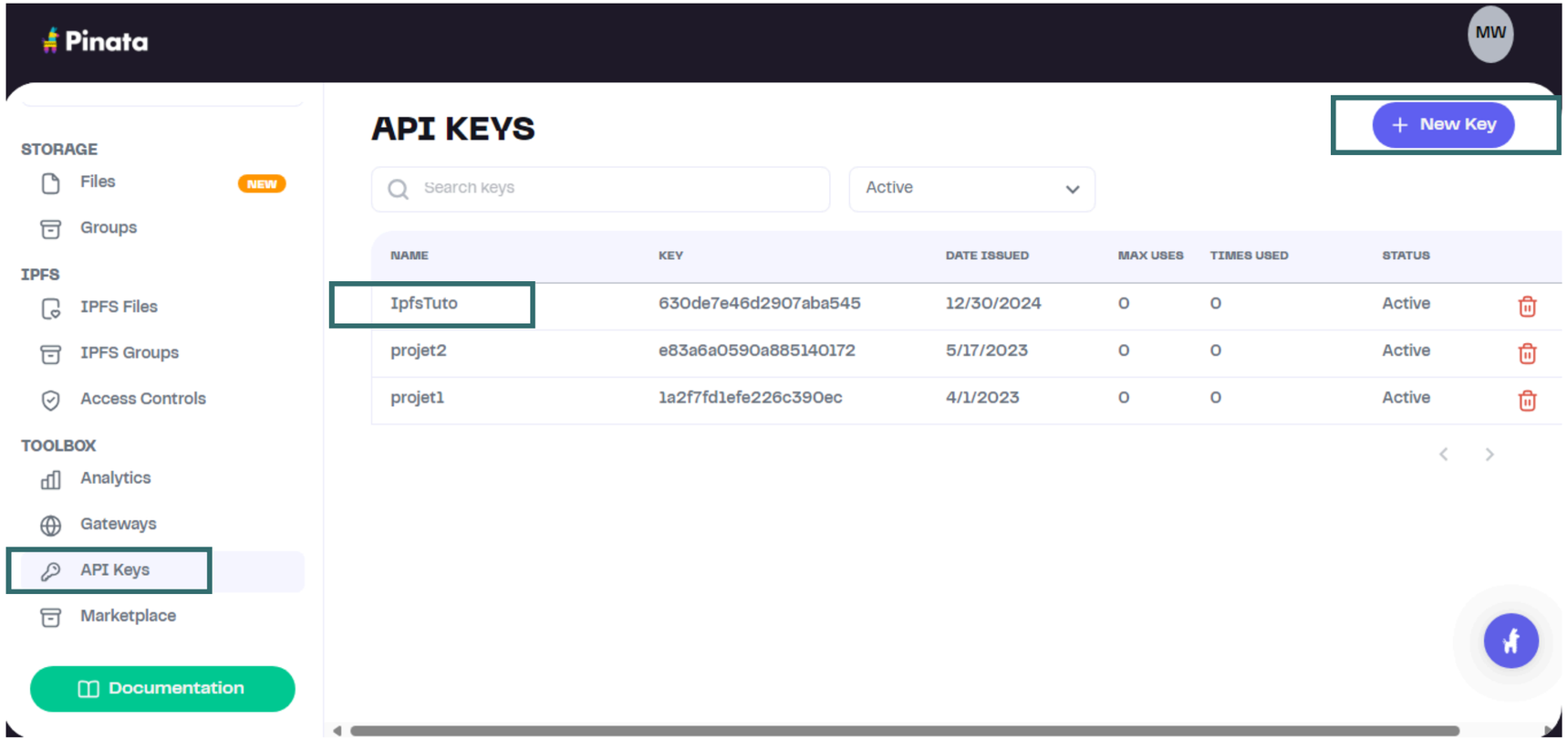
- **User-friendly: Simplifies IPFS integration for decentralized apps (dApps).**
- **Scalable: Ideal for both small and large decentralized applications.**

Setting Up Pinata

@maryemhadjwannes

1- Sign up on Pinata: Go to the [Pinata website](#) and sign up.

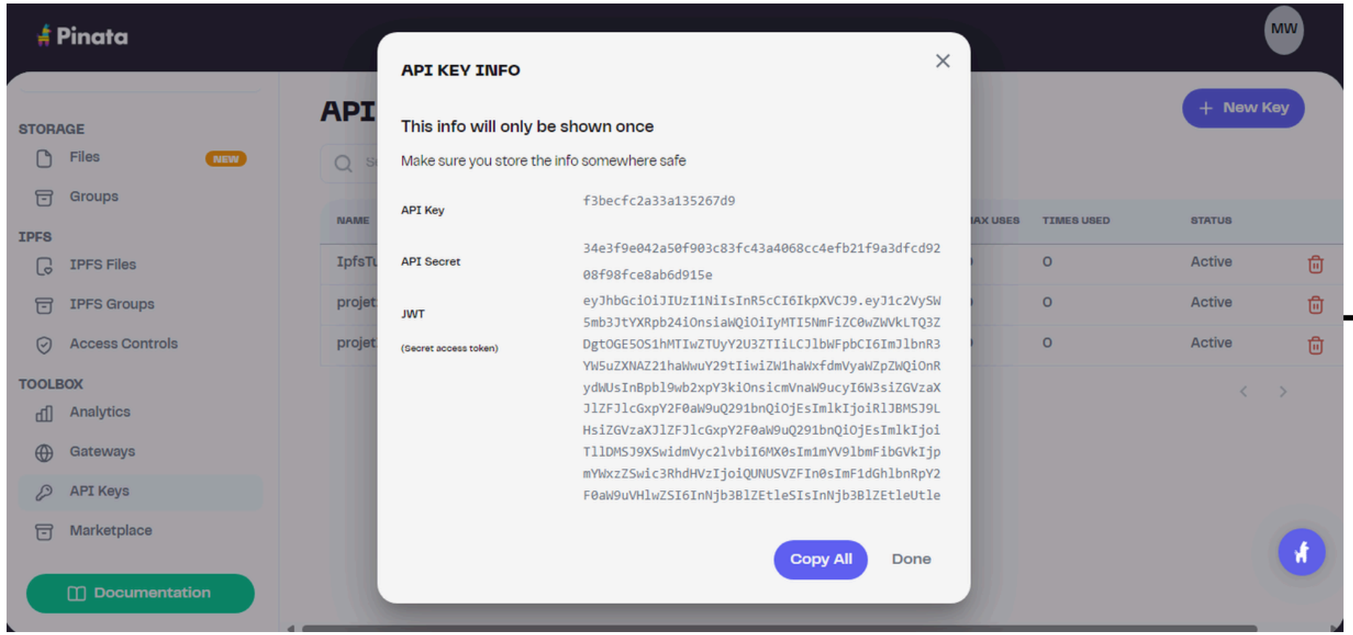
2- Create API Keys: After logging in, navigate to the API Keys section and create a new key with the required permissions



Setting Up Pinata

@maryemhadjwannes

3- Set up your environment: Replace your pinataApiKey and pinataSecretApiKey with the ones provided by Pinata.




```
const pinataApiKey = " f3becfc2a33a135267d9"; // Replace with your API key
const pinataSecretApiKey = "34e3f9e042a50f903c83fc43a4068cc4efb21f9a3dfcd9208f98fce8ab6d915e";
```

Uploading JSON to IPFS

@maryemhadjwannes

- You can send JSON data to IPFS using the pinJSONToIPFS endpoint.
- This function will accept a JSON object and upload it to IPFS.

```
// Function to upload JSON
async function uploadJSONToPinata(jsonData) {
    try {
        const url = `https://api.pinata.cloud/pinning/pinJSONToIPFS`;

        const response = await axios.post(url, jsonData, {
            headers: {
                pinata_api_key: pinataApiKey,
                pinata_secret_api_key: pinataSecretApiKey,
            },
        });

        console.log("JSON Uploaded:", response.data);
        return response.data;
    } catch (error) {
        console.error("Error uploading JSON:", error.response?.data || error.message);
    }
}
```

Uploading Files to IPFS

@maryemhadjwannes

- **For file uploads, you need to use the pinFileToIPFS endpoint and send a file as part of the form data.**

```
// Function to upload file
async function uploadFileToPinata(filePath) {
    try {
        const url = `https://api.pinata.cloud/pinning/pinFileToIPFS`;

        const formData = new FormData();
        formData.append("file", fs.createReadStream(filePath));

        const response = await axios.post(url, formData, {
            headers: {
                "Content-Type": `multipart/form-data; boundary=${formData._boundary}`,
                pinata_api_key: pinataApiKey,
                pinata_secret_api_key: pinataSecretApiKey,
            },
        });

        console.log("File Uploaded:", response.data);
        return response.data;
    } catch (error) {
        console.error("Error uploading file:", error.response?.data || error.message);
    }
}
```

Example Usage

- **This part demonstrates uploading both a JSON object and a file to IPFS.**

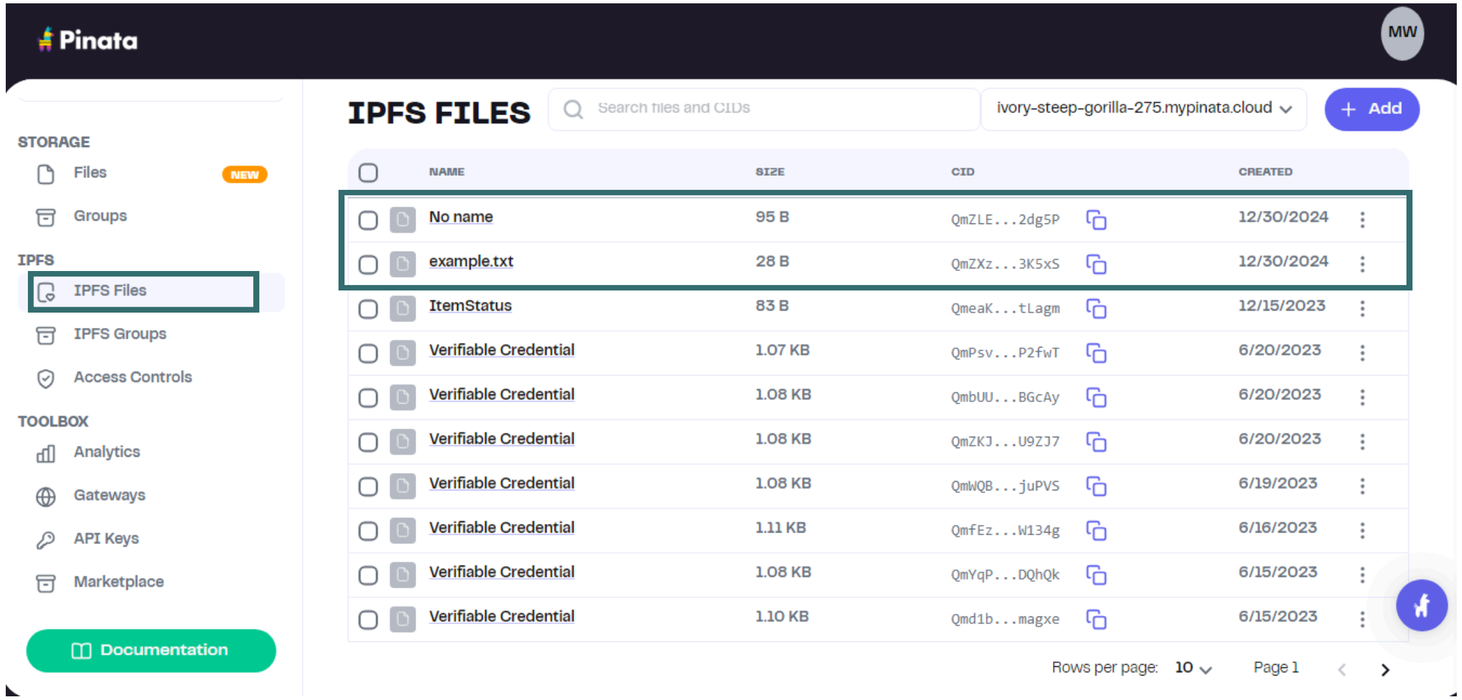
```
// Example usage
(async () => {
    // Upload JSON
    const jsonData = {
        name: "Pinata Example",
        description: "Uploading JSON and files to IPFS using Pinata",
    };
    await uploadJSONToPinata(jsonData);

    // Upload a file
    const filePath = "./example.txt"; // Replace with the path to your file
    await uploadFileToPinata(filePath);
})();
```

Example Usage

@maryemhadjwannes

- The script successfully uploads JSON data and files to IPFS using Pinata. After uploading, the IPFS Hash is returned for both JSON and file uploads, which can be used to access the data stored on IPFS.
- You can easily access your uploaded data through the Pinata interface by navigating to the IPFS Files section, where you can view and manage your uploaded files.

```
PS C:\Users\user\Documents\ipfs-tutorial> node index.js
JSON Uploaded: {
  IpfsHash: 'QmZLE6BjMa38mP6uy7bD8iZgQA1ttwwGCma9B9pum2dg5P',
  PinSize: 95,
  Timestamp: '2024-12-30T00:39:08.490Z'
}
File Uploaded: {
  IpfsHash: 'QmZXzqsteHuVxqR3xC3QYCwj1jprfYbPC5q57mURk3K5xS',
  PinSize: 28,
JSON Uploaded: {
  IpfsHash: 'QmZLE6BjMa38mP6uy7bD8iZgQA1ttwwGCma9B9pum2dg5P',
  PinSize: 95,
  Timestamp: '2024-12-30T00:39:08.490Z'
}
```

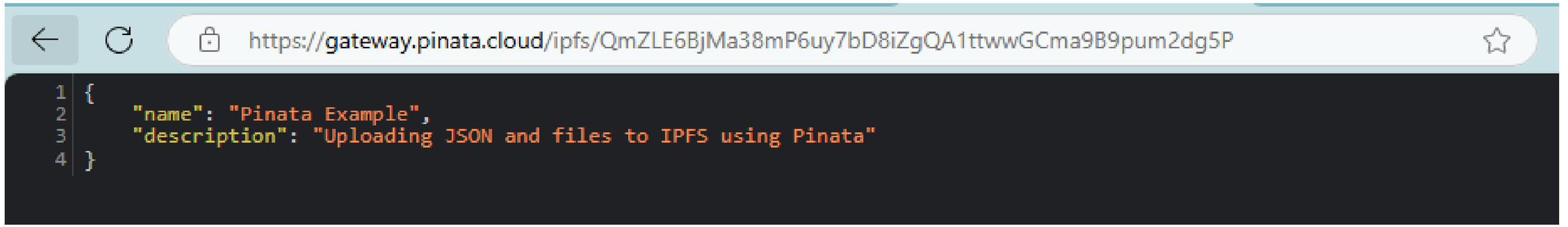


Example Usage

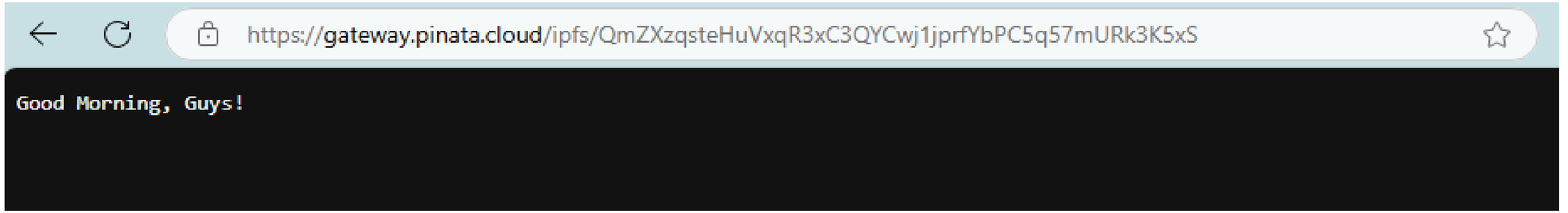
@maryemhadjwannes

We can Access Uploaded Files via the IPFS Hash

- **JSON**



- **Example.txt**



Handling Errors and Debugging

- **Error Handling:** If the API key doesn't have the right permissions or if there is an issue with the file format, you'll receive an error message. The code uses try...catch blocks to handle and display errors.

- **Common Errors:**
  - Missing API key or wrong API key: 401 Unauthorized
  - Invalid file format or large file size: 413 Payload Too Large

I hope you found this tutorial helpful!