

# StoreIt Protocol v0.4

---

## 1. Introduction

Hello, this is the documentation for the StoreIt protocol. It is used to communicate with our server. We will be implementing this protocol on top of WEBSOCKETS to enjoy its messaging model. Everything should be JSON objects.

## 2. The JSON data structures

### 2.1 Command

```
{
  "uid": unique_command_id,
  "command": command_name,
  "parameters": {
    "parameter1Name": parameter1,
    "parameter2Name": parameter2,
  }
}
```

### 2.2 Response

```
{
  "code": code,
  "text": response_message,
  "commandUid": command_id,
  "command": "RESP",
  (optional) "parameters": {
    ...
  }
}
```

```
}
```

TODO: document possible errors.

## 2.3 Commands

### JOIN

From a client to the server. This is the first request to make whenever a client wants to get online.

```
{
  "uid": 263,
  "command": "JOIN",
  "parameters": {
    "authType": "fb", // fb for facebook and gg for goog
    "accessToken": "34j8b4jhb343hbKJH54",
    "hosting": [
      'QmNMNRCgNBvkdXyXuVa2cHwTJJ9wtJQht1Njx1pqNBC9cV'
      'QmYwAPJzv5CZsnA625s3Xf2nemtYgPpHdWEz79ojWnPbdG'
    ]
  }
}
```

The response will contain a FILE object named "home" and user profile info. Example :

```
{
  "code": 0,
  "text": "welcome",
  "commandUid": 42,
  "command": "RESP",
  "parameters": {
    "home": FILEObject,
    "userProfile": {"picture": "pic.jpg"}
  }
}
```

```
}  
}
```

## FDEL

From a client or the server. Delete a file/directory.

```
{  
  "uid": 765,  
  "command": "FDEL",  
  "parameters": {  
    "files": ["/a.txt", "/archive/b.txt", "/dir"]  
  }  
}
```

## FADD

From a client or a server. Add a file to the user three.

```
{  
  "uid": 766,  
  "command": "FADD",  
  "parameters": {  
    "files": [FILEObject, ...]  
  }  
}
```

## FUPT

From a client or a server. Update a file.

```
{  
  "uid": 767,  
  "command": "FUPT",  
  "parameters": {
```

```
    "files": [FILEObject, ...]
  }
}
```

You should do only one FUPT per file/directory and omit the files parameter of your directory. For example, if your home is :

```
| foo
L___ bar.txt
L___ pictures
```

And you want to update foo's timestamp, just send :

```
{
  "uid": 767,
  "command": "FUPT",
  "parameters": {
    "files": [{
      "path": "/foo",
      "metadata": "updated metadata",
      "IPFHash": null,
      "isDir": true,
      "files": null
    }]
  }
}
```

## FMOV

From a client or a server. move or rename a file.

```
{
  "uid": 768,
  "command": "FMOV",
```

```
  "parameters": {
    "src": "/foo/bar.txt"
    "dest": "/foo/toto.txt"
  }
}
```

If you are moving a file, please don't omit the file name in the destination.  
For example :

DON'T DO:

```
{
  "src": "/foo/bar"
  "dest": "/target/"
}
```

expecting to move /foo/bar into /target/bar

DO:

```
{
  "src": "/foo/bar"
  "dest": "/target/bar"
}
```

## FSTR

From the server to a client Store an IPFS object

```
{
  "uid": 7668,
  "command": "FSTR",
  "parameters": {
    "hash": "QmYwAPJzv5CZsnA625s3Xf2nemtYgPpHdWEz79ojWnP"
```

```
    "keep": true|false
  }
}
```

If the "keep" parameter is false, the object should be deleted. Otherwise it should be downloaded from IPFS and stored in the local repository.

## 2.4 FILE object

This object describe a file or a directory.

```
{
  "path": "/foo/bar",
  "metadata": metadata,
  "IPFSHash": "IPFS hash of all the data in the file",
  "isDir": true,
  "files": {
    "foo.txt": FILEObject,
    "someDirectory": FILEObject,
  }
}
```