

---

# **P2P File Sharing Software**

**Cedric Bone**

**April 13, 2025**



**CONTENTS:**

<b>1</b>	<b>cedric_bone_project</b>	<b>3</b>
1.1	cli module . . . . .	3
1.2	main module . . . . .	3
1.3	node module . . . . .	3
1.4	utils module . . . . .	5
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>
	<b>Index</b>	<b>11</b>



Add your content using `reStructuredText` syntax. See the [reStructuredText](#) documentation for details.



## CEDRIC\_BONE\_PROJECT

### 1.1 cli module

Command-line interface for P2P file sharing app

**class** `cli.CommandLine(node)`

Bases: `object`

Command-line interface

**connect**(*args*)

Connect to peer

**download**(*args*)

Download a file

**download\_thread**(*peer, filename*)

Thread function for downloading

**list**(*args*)

List local files

**peers**(*args*)

List peers

**search**(*args*)

Search for files

**start**()

Start the CLI

### 1.2 main module

Main for P2P file sharing app

`main.main()`

### 1.3 node module

P2P node for file sharing

```
class node.P2PNode(shared_dir)
    Bases: object
    Modded simple version P2P file sharing from BitTorrent

    add_peer(ip, port)
        Manually add a peer

    download_chunk(peer, filename, chunk_index)
        Download a file chunk

    download_file(peer, filename)
        Download file

    get_file_info(file_path)
        File metadata

    get_files()
        list of local files

    get_peers()
        Get list of peers

    handle_chunk_request(client, request)
        Handle file chunk request

    handle_client(client, addr)
        Handle client request

    handle_info_request(client, request)
        Handle file info request

    handle_list_request(client)
        Request for file list

    index_files()
        Index files indirectory

    maintenance()
        Periodic maintenance

    remove_peer(ip, port)
        Remove a peer

    request_file_info(peer, filename)
        Request file

    request_file_list(peer)
        Request file list from peer

    run_server()
        Run server to handle incoming requests

    search_files(query)
        Search for files matching query

    start()
        Start P2P node
```



**stop()**

Stop P2P node

## 1.4 utils module

Utility function for P2P file sharing application

**utils.find\_free\_port()**

Find available port



## INDICES AND TABLES



## PYTHON MODULE INDEX

### c

`cli`, 3

### m

`main`, 3

### n

`node`, 3

### u

`utils`, 5



## INDEX

### A

`add_peer()` (*node.P2PNode* method), 4

### C

`cli`

    module, 3

`CommandLine` (*class in cli*), 3

`connect()` (*cli.CommandLine* method), 3

### D

`download()` (*cli.CommandLine* method), 3

`download_chunk()` (*node.P2PNode* method), 4

`download_file()` (*node.P2PNode* method), 4

`download_thread()` (*cli.CommandLine* method), 3

### F

`find_free_port()` (*in module utils*), 5

### G

`get_file_info()` (*node.P2PNode* method), 4

`get_files()` (*node.P2PNode* method), 4

`get_peers()` (*node.P2PNode* method), 4

### H

`handle_chunk_request()` (*node.P2PNode* method), 4

`handle_client()` (*node.P2PNode* method), 4

`handle_info_request()` (*node.P2PNode* method), 4

`handle_list_request()` (*node.P2PNode* method), 4

### I

`index_files()` (*node.P2PNode* method), 4

### L

`list()` (*cli.CommandLine* method), 3

### M

`main`

    module, 3

`main()` (*in module main*), 3

`maintenance()` (*node.P2PNode* method), 4

`module`

*cli*, 3

*main*, 3

*node*, 3

*utils*, 5

### N

`node`

    module, 3

### P

`P2PNode` (*class in node*), 3

`peers()` (*cli.CommandLine* method), 3

### R

`remove_peer()` (*node.P2PNode* method), 4

`request_file_info()` (*node.P2PNode* method), 4

`request_file_list()` (*node.P2PNode* method), 4

`run_server()` (*node.P2PNode* method), 4

### S

`search()` (*cli.CommandLine* method), 3

`search_files()` (*node.P2PNode* method), 4

`start()` (*cli.CommandLine* method), 3

`start()` (*node.P2PNode* method), 4

`stop()` (*node.P2PNode* method), 4

### U

`utils`

    module, 5