# XZB-1248/Spark: ✨Spark is a web-based, cross-platform and full-featured Remote Administration Tool (RAT) written in Go that allows you control all your devices anywhere. Spark是一个Go编写的，网页UI、跨平台以及多功能的远程控制和监控工具，你可以随时随地监控和控制所有设备。

github.com/XZB-1248/Spark

XZB-1248

### [English] [中文] [API Document] [API文档]

## Spark

**Spark** is a free, safe, open-source, web-based, cross-platform and full-featured RAT (Remote Administration Tool) that allow you to control all your devices via browser anywhere.

We **won't** collect any data, thus the server will never self-upgrade. Your clients will only communicate with your server forever.

repo size 29.8 MB   issues 16 open   issues 28 closed

downloads 4.1k   downloads@latest 1.4k

## Disclaimer

**THIS PROJECT, ITS SOURCE CODE, AND ITS RELEASES SHOULD ONLY BE USED FOR EDUCATIONAL PURPOSES.**
**ALL ILLEGAL USAGE IS PROHIBITED!**
**YOU SHALL USE THIS PROJECT AT YOUR OWN RISK.**
**THE AUTHORS AND DEVELOPERS ARE NOT RESPONSIBLE FOR ANY DAMAGE CAUSED BY YOUR MISUSE OF THIS PROJECT.**

**YOUR DATA IS PRICELESS. THINK TWICE BEFORE YOU CLICK ANY BUTTON OR ENTER ANY COMMAND.**

## Quick start

### binary

- Download executable from releases.
- Following this to complete configuration.
- Run executable and browse to `http://IP:Port` to access the web interface.
- Generate a client and run it on your target device.
- Enjoy!

## Configuration

Configuration file `config.json` should be placed in the same directory as the executable file. Example:

```
{
    "listen": ":8000",
    "salt": "123456abcdefg",
    "auth": {
        "username": "password"
    },
    "log": {
        "level": "info",
        "path": "./logs",
        "days": 7
    }
}
```

- `listen` `required`, format: `IP:Port`
- `salt` `required`, length <= 24

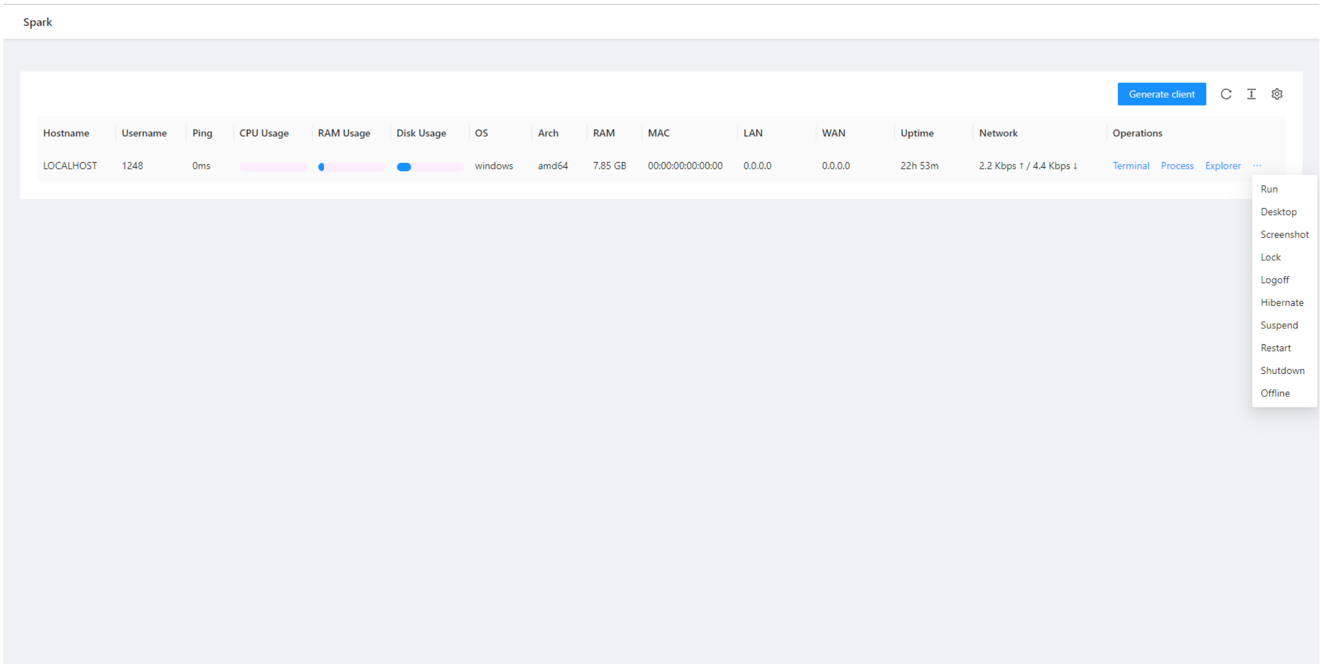
  after modification, you need to re-generate all clients
- `auth` `optional`, format: `username:password`
  - hashed-password is highly recommended
  - format: `$algorithm$hashed-password`, example: `$sha256$123456abcdefg`
  - supported algorithms: `sha256`, `sha512`, `bcrypt`
  - if you don't follow the format, password will be treated as plain-text
- `log` `optional`
  - `level` `optional`, possible value: `disable`, `fatal`, `error`, `warn`, `info`, `debug`
  - `path` `optional`, default: `./logs`
  - `days` `optional`, default: `7`

## Features

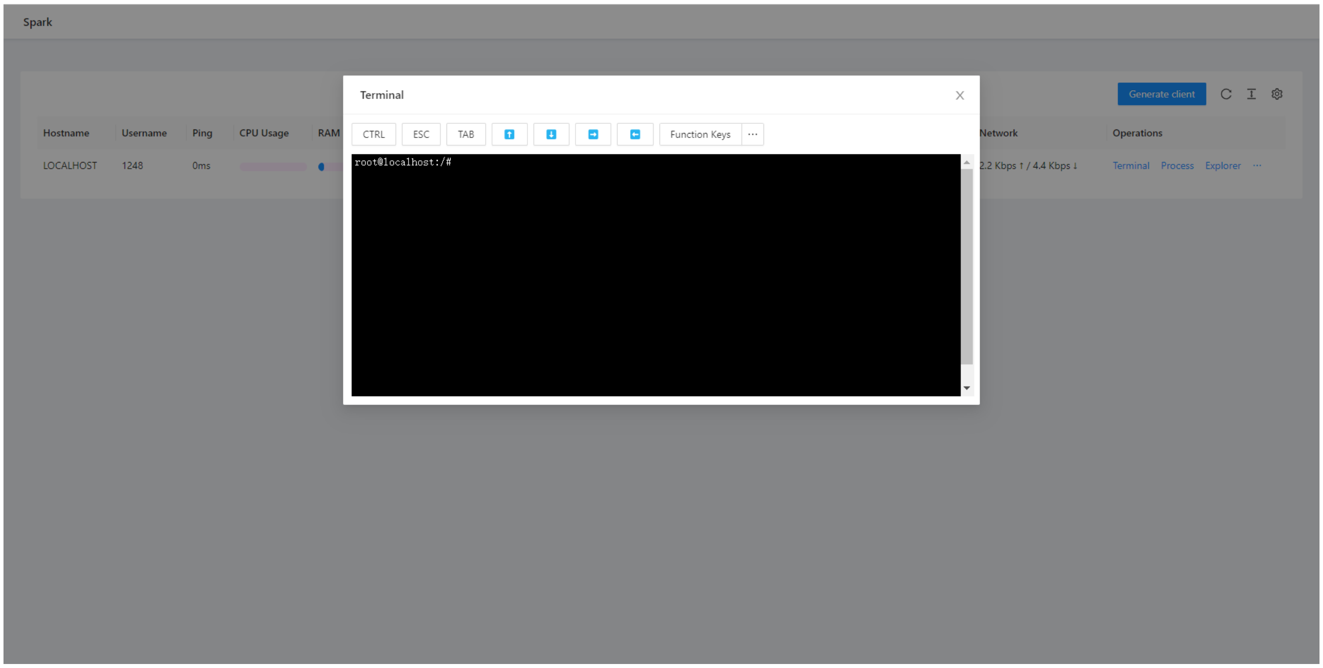
| Feature/OS | Windows | Linux | MacOS |
|---|---|---|---|
| Process manager | ✔ | ✔ | ✔ |
| Kill process | ✔ | ✔ | ✔ |
| Network traffic | ✔ | ✔ | ✔ |
| File explorer | ✔ | ✔ | ✔ |
| File transfer | ✔ | ✔ | ✔ |
| File editor | ✔ | ✔ | ✔ |
| Delete file | ✔ | ✔ | ✔ |
| Code highlight | ✔ | ✔ | ✔ |
| Desktop monitor | ✔ | ✔ | ✔ |
| Screenshot | ✔ | ✔ | ✔ |
| OS info | ✔ | ✔ | ✔ |
| Terminal | ✔ | ✔ | ✔ |
| * Shutdown | ✔ | ✔ | ✔ |
| * Reboot | ✔ | ✔ | ✔ |
| * Log off | ✔ | ❌ | ✔ |
| * Sleep | ✔ | ❌ | ✔ |
| * Hibernate | ✔ | ❌ | ❌ |
| * Lock screen | ✔ | ❌ | ❌ |

- Blank cell means the situation is not tested yet.
- The Star symbol means the function may need administration or root privilege.

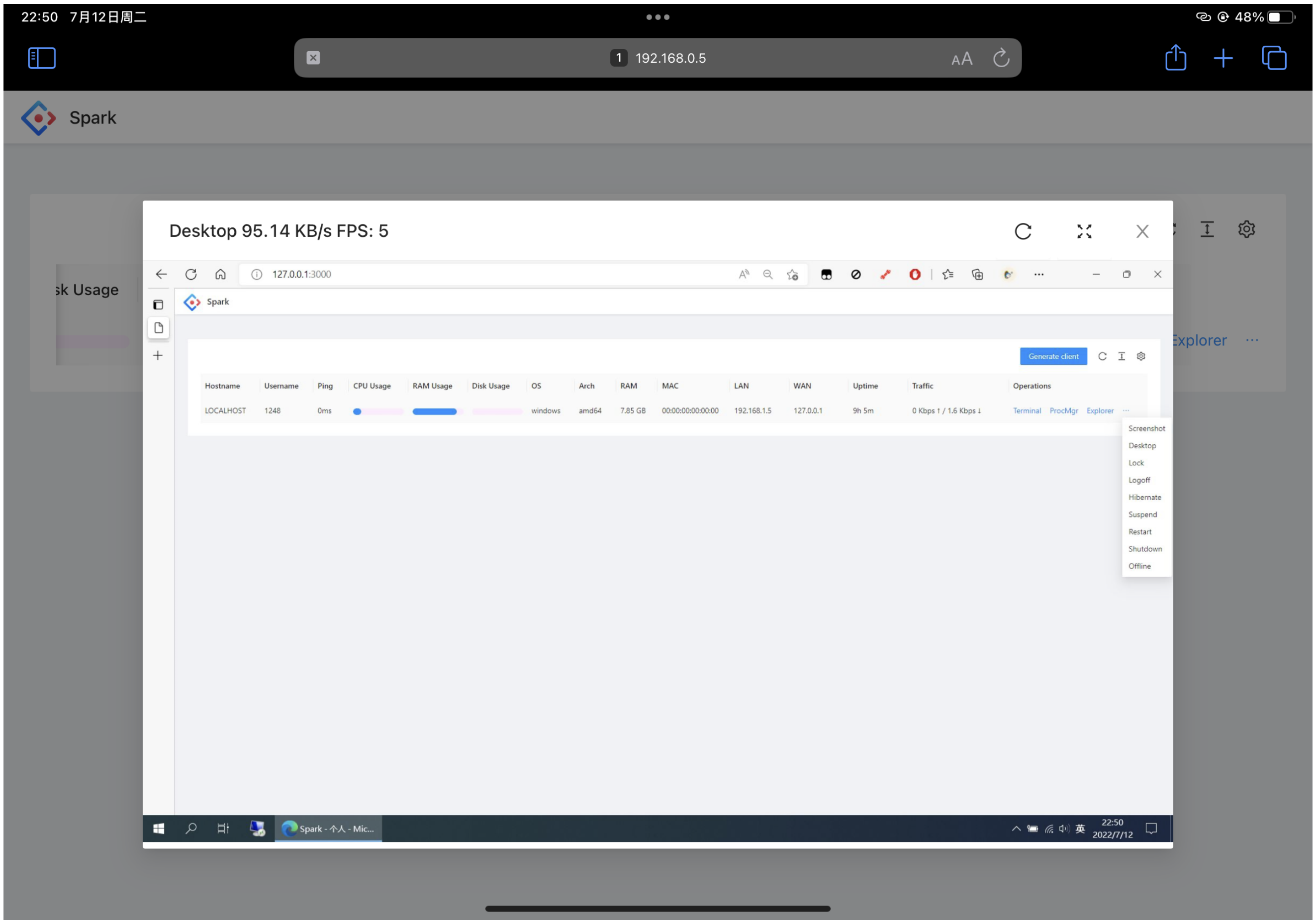
## Screenshots



| Hostname | Username | Ping | CPU Usage | RAM Usage | Disk Usage | OS | Arch | RAM | MAC | LAN | WAN | Uptime | Network | Operations |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOCALHOST | 1248 | 0ms | | | | windows | amd64 | 7.85 GB | 00:00:00:00:00:00 | 0.0.0.0 | 0.0.0.0 | 22h 53m | 2.2 Kbps ↑ / 4.4 Kbps ↓ | Terminal Process Explorer ··· |

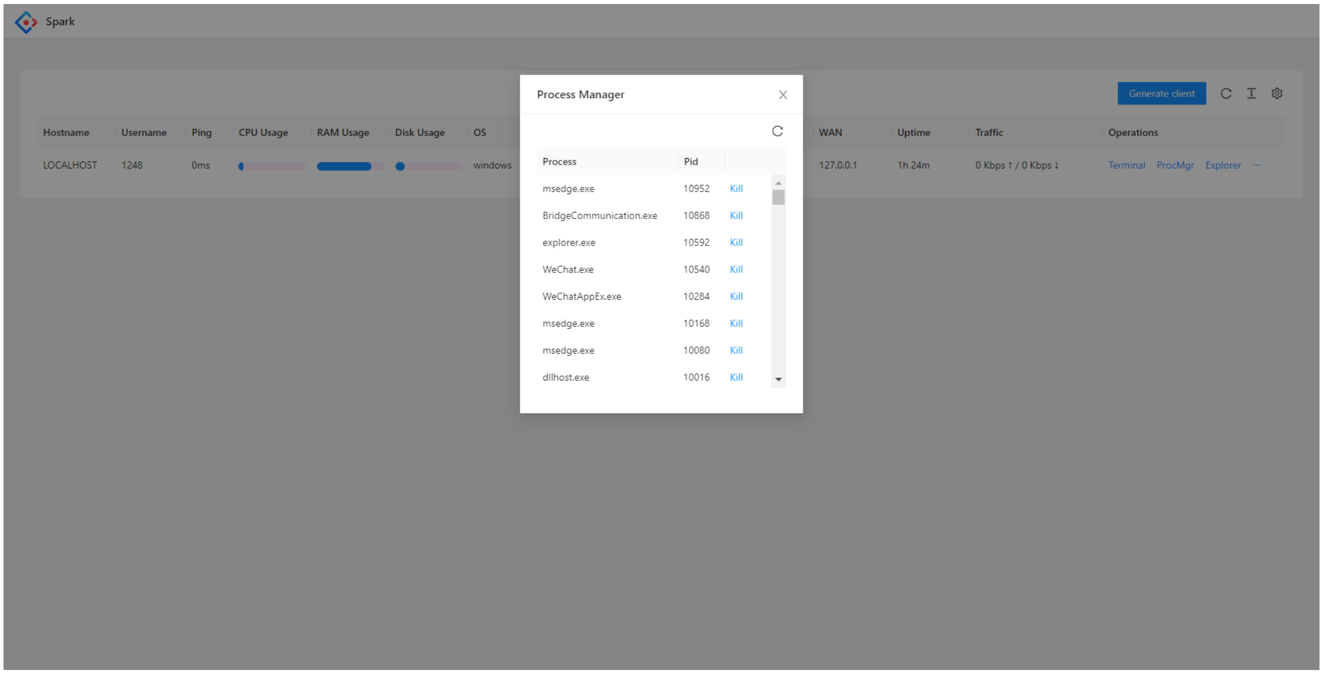


```
root@localhost:/#
```



Desktop 95.14 KB/s FPS: 5

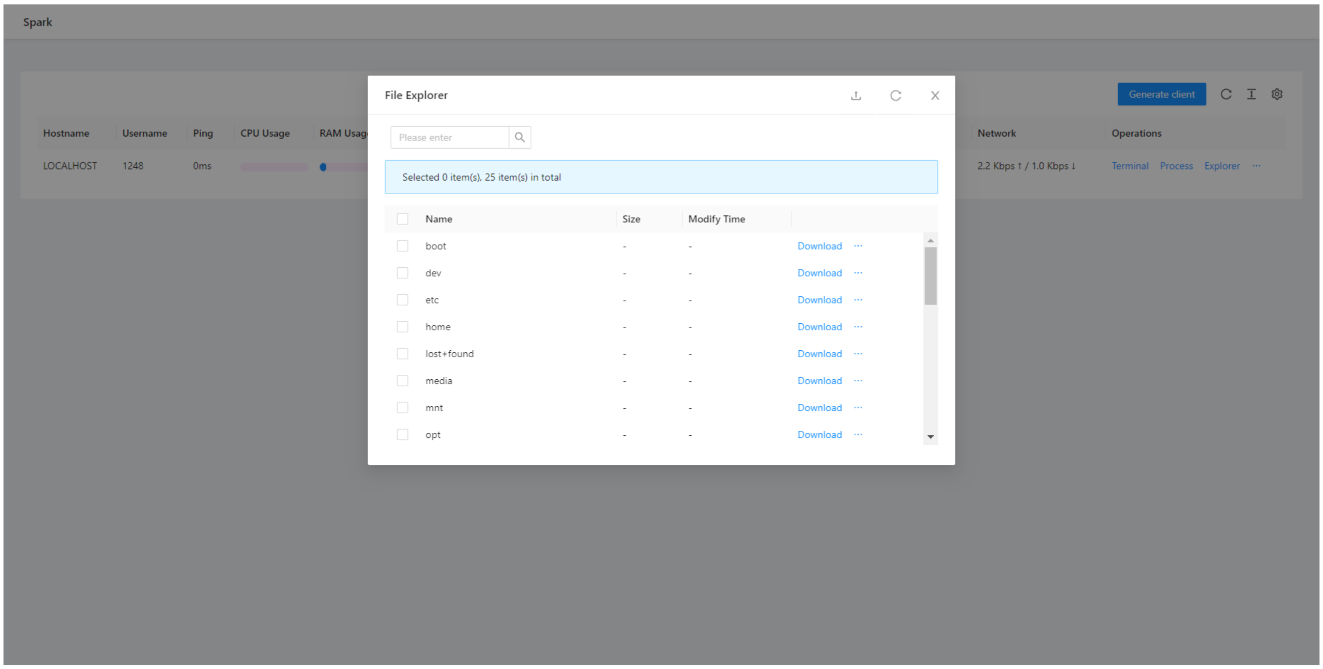
| Hostname | Username | Ping | CPU Usage | RAM Usage | Disk Usage | OS | Arch | RAM | MAC | LAN | WAN | Uptime | Traffic | Operations |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOCALHOST | 1248 | 0ms | | | | windows | amd64 | 7.85 GB | 00:00:00:00:00:00 | 192.168.1.5 | 127.0.0.1 | 9h 5m | 0 Kbps ↑ / 1.6 Kbps ↓ | Terminal ProcMgr Explorer ··· |



| Hostname | Username | Ping | CPU Usage | RAM Usage | Disk Usage | OS | WAN | Uptime | Traffic | Operations |
|---|---|---|---|---|---|---|---|---|---|---|
| LOCALHOST | 1248 | 0ms | | | | windows | 127.0.0.1 | 1h 24m | 0 Kbps ↑ / 0 Kbps ↓ | Terminal ProcMgr Explorer ··· |

Process Manager

| Process | Pid | |
|---|---|---|
| msedge.exe | 10952 | Kill |
| BridgeCommunication.exe | 10868 | Kill |
| explorer.exe | 10592 | Kill |
| WeChat.exe | 10540 | Kill |
| WeChatAppEx.exe | 10284 | Kill |
| msedge.exe | 10168 | Kill |
| msedge.exe | 10080 | Kill |
| dllhost.exe | 10016 | Kill |



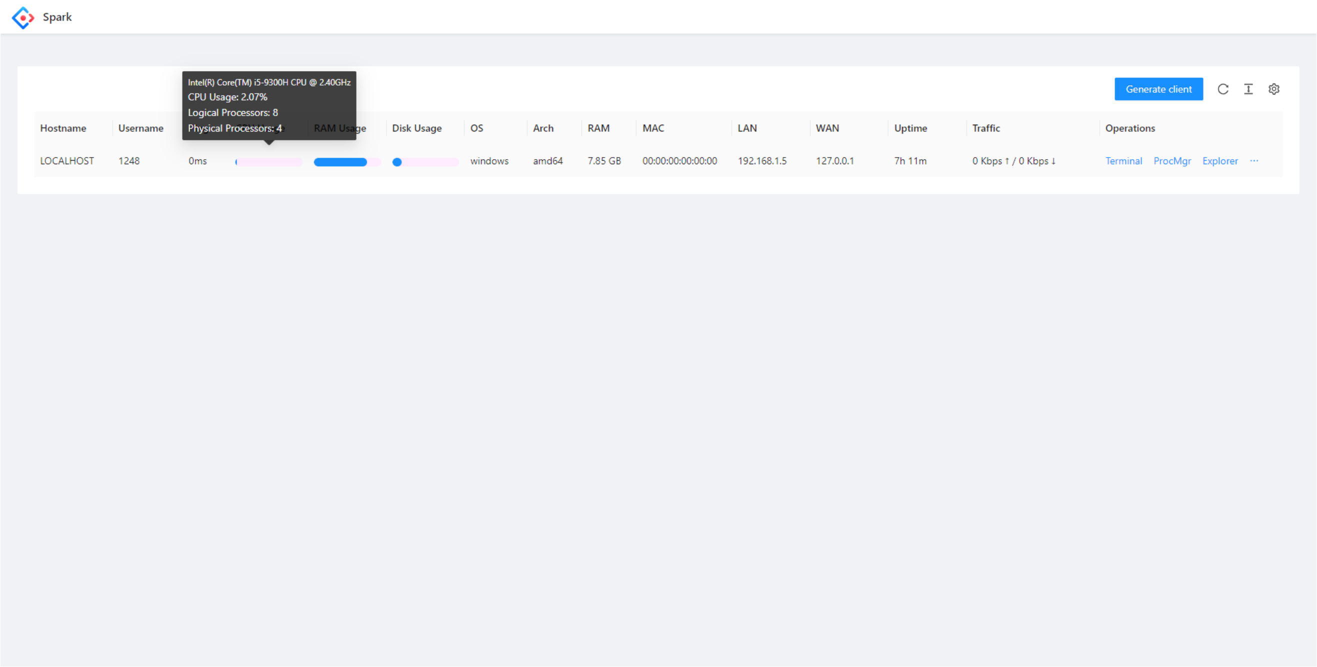
Spark

| Hostname | Username | Ping | CPU Usage | RAM Usag | Network | Operations |
|---|---|---|---|---|---|---|
| LOCALHOST | 1248 | 0ms | | | 2.2 Kbps ↑ / 1.0 Kbps ↓ | Terminal Process Explorer ··· |

File Explorer

Selected 0 item(s), 25 item(s) in total

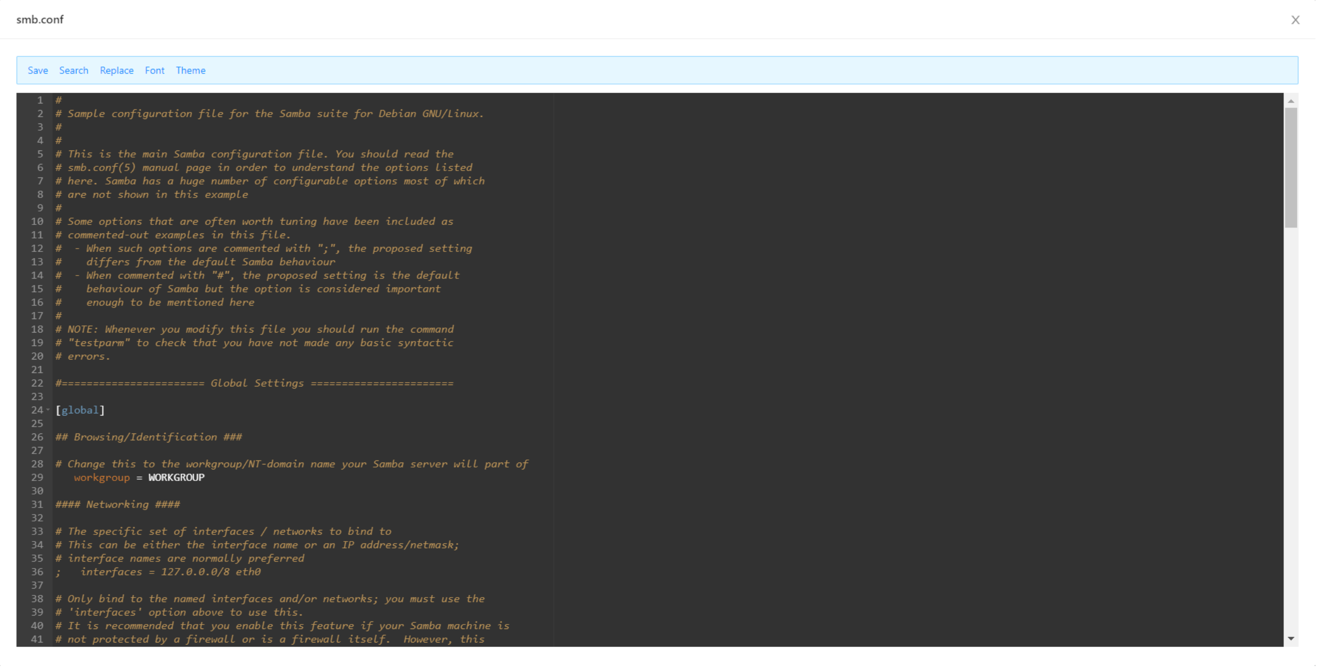
| ☐ | Name | Size | Modify Time | |
|---|---|---|---|---|
| ☐ | boot | - | - | Download ··· |
| ☐ | dev | - | - | Download ··· |
| ☐ | etc | - | - | Download ··· |
| ☐ | home | - | - | Download ··· |
| ☐ | lost+found | - | - | Download ··· |
| ☐ | media | - | - | Download ··· |
| ☐ | mnt | - | - | Download ··· |
| ☐ | opt | - | - | Download ··· |



Spark

Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
CPU Usage: 2.07%
Logical Processors: 8
Physical Processors: 4

| Hostname | Username | | | RAM Usage | Disk Usage | OS | Arch | RAM | MAC | LAN | WAN | Uptime | Traffic | Operations |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOCALHOST | 1248 | 0ms | | | | windows | amd64 | 7.85 GB | 00:00:00:00:00:00 | 192.168.1.5 | 127.0.0.1 | 7h 11m | 0 Kbps ↑ / 0 Kbps ↓ | Terminal ProcMgr Explorer ··· |



```
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
#  - When such options are commented with ";", the proposed setting
#    differs from the default Samba behaviour
#  - When commented with "#", the proposed setting is the default
#    behaviour of Samba but the option is considered important
#    enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.

#======================= Global Settings =======================

[global]

## Browsing/Identification ###

# Change this to the workgroup/NT-domain name your Samba server will part of
   workgroup = WORKGROUP

#### Networking ####

# The specific set of interfaces / networks to bind to
# This can be either the interface name or an IP address/netmask;
# interface names are normally preferred
;   interfaces = 127.0.0.0/8 eth0

# Only bind to the named interfaces and/or networks; you must use the
# 'interfaces' option above to use this.
# It is recommended that you enable this feature if your Samba machine is
# not protected by a firewall or is a firewall itself.  However, this
```

# Development

## note

There are three components in this project, so you have to build them all.

Go to Quick start if you don't want to make yourself boring.

- Client
- Server
- Front-end

If you want to make client support OS except linux and windows, you should install some additional C compiler.

For example, to support android, you have to install Android NDK.

## tutorial

```
# Clone this repository.
$ git clone https://github.com/XZB-1248/Spark
$ cd ./Spark


# Here we're going to build front-end pages.
$ cd ./web
# Install all dependencies and build.
$ npm install
$ npm run build-prod


# Embed all static resources into one single file by using statik.
$ cd ..
$ go install github.com/rakyll/statik
$ statik -m -src="./web/dist" -f -dest="./server/embed" -p web -ns web


# Now we should build client.
# When you're using unix-like OS, you can use this.
$ mkdir ./built
$ go mod tidy
$ go mod download
$ ./scripts/build.client.sh


# Finally we're compiling the server side.
$ mkdir ./releases
$ ./scripts/build.server.sh
```

Then create a new directory with a name you like.
Copy executable file inside `releases` to that directory.
Copy the whole `built` directory to that new directory.
Copy configuration file mentioned above to that new directory.
Finally, run the executable file in that directory.

---

## Dependencies

Spark contains many third-party open-source projects.

Lists of dependencies can be found at `go.mod` and `package.json`.

Some major dependencies are listed below.

### Back-end

- Go (License)
- gin-gonic/gin (MIT License)

- imroc/req (MIT License)
- kbinani/screenshot (MIT License)
- shirou/gopsutil (License)
- gorilla/websocket (BSD-2-Clause License)
- orcaman/concurrent-map (MIT License)

## Front-end

- React (MIT License)
- Ant-Design (MIT License)
- axios (MIT License)
- xterm.js (MIT License)
- crypto-js (MIT License)

## Acknowledgements

- natpass (MIT License)
- Image difference algorithm inspired by natpass.

## Stargazers over time

rate limited, please try again later

# License

BSD-2 License