

# 青阳文件传输系统使用手册

欢迎使用 *kift v1.0.2*

@青阳龙野



## 目录

什么是 kiftd? .....	2
现在, 让我们开始吧 .....	3
在 Windows 系统下 3 分钟快速安装 .....	3
在 Linux/Unix(Mac OS X)系统下 3 分钟快速安装 .....	9
功能说明 .....	13
自定义端口号 .....	13
设置“必须登入”功能 .....	15
各种预览与播放功能 .....	15
批量操作 .....	17
修改文件系统存放位置 (或切换文件系统) .....	18
日志记录功能 .....	19
设置用户与权限 .....	20
修改缓存大小 .....	21
纯命令模式 .....	22
安全性与维护 .....	22
许可声明 .....	23
联系作者/希望帮助 kift 做的更好? .....	23
特别鸣谢 .....	24
开发小记 .....	24

# 什么是 kiftd?

“青阳文件传输系统”（kiftd）是一款小型、多功能、开源的个人网盘。

您还在使用 U 盘分享软件么？很不幸，U 盘易丢，同时又无法兼容各种平台的文件系统，在需要大规模分享时——通过 U 盘拷贝简直就像是接力赛一样原始又低效。

您还在使用免费的公用网盘么？时不时传来的网盘关停热潮令人担忧，而其由外人管理的特性则让您不敢轻易把机密文件存放在上面。

您需要搭建起一款私人的网盘而苦于没有好的选择？破解软件同样无法确保安全，隐藏的病毒令人担忧。同时，多数已有的网盘服务器软件设计老旧、性能笨重、语言的障碍更是使用中不可忽视的问题。

现在，您可以选择 kift 了。

本应用的作者 [青阳龙野@kohgylw](mailto:青阳龙野@kohgylw) 相信：不仅仅是他本人，而是还需要有很多人都需要在局域网内自己搭建起一个属于个人或团体的网盘系统，这样就可以让所有在同一局域网内的朋友、同事或学生去访问、上传或下载自己网盘上的文件——无论是不足 1MB 的 doc 文本，还是超过 4GB 的 1080p 视频。而且这种网盘的安装必须十分简单快速，界面要使用中文，且能够完全兼容 Windows/Linux/Unix(包括 Mac OS X)系统。

此外，这个网盘还应该像主流<sup>的</sup>免费网盘一样——支持 Mp4 视频直接在线播放、pdf 文档直接预览、图片直接查看功能、MP3 音乐播放功能、ZIP 打包下载、批量上传……等等高级功能。

——实际上，作者最初编写 kift 的目的就是为了能实现上述功能，在他成功弄丢了几次 U 盘之后，便萌生了利用闲暇时间设计一款这样的应用的想法。

# 现在，让我们开始吧

接下来本文会用最直白的方式，来生动地演示如何在 3 分钟内愉快地安装和使用这个网盘。当然，您需要准备出 3 分钟 的耐心（如果您实际跟随示例操作一遍，会发现 3 分钟足够了）。

首先，请根据您的操作系统选择一个阅读方向：

## 在 Windows 系统下 3 分钟快速安装

### 1，安装 JAVA 运行环境

Kiftd 是用 java 语言写成的，这也是服务器开发的经典语言，因此想要运行 kiftd 就需要一个叫 java 运行环境的东西支持它运行——这点类似于 Flash、VC++Library 等，但又不一样（我不会解释这句话的）。

总之，我可以保证 java 的安装过程比上述两个环境简单得多。

那么，到底要怎么安装呢？

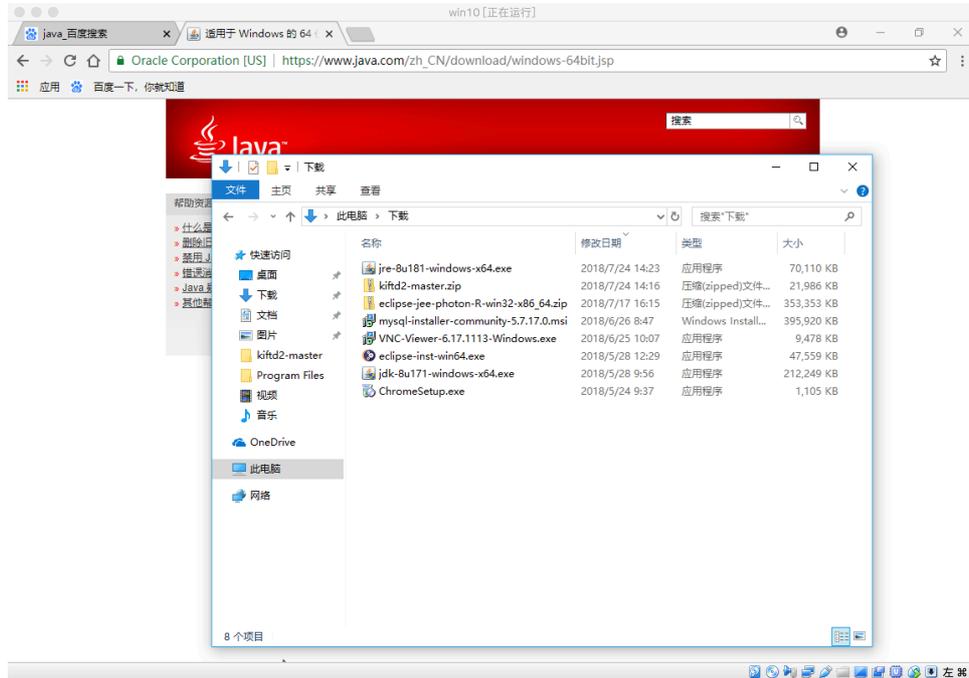
1.1 首先，请访问 java 官方网站：



您可以使用主流搜索引擎搜索“java”关键字并找到其官网（很显眼），或直

请访问：[https://www.java.com/zh\\_CN/](https://www.java.com/zh_CN/)

1.2 之后，根据提示点击“免费 java 下载”并将其下载到本地，下载之后会看到一个.exe 安装程序：



如图，这个名为“jre-8u181-windows-x64.exe”的程序即为 java 当前版本的安装包。其版本号可能因您下载的时间不同而有所差别（反正都是 java 运行环境）。

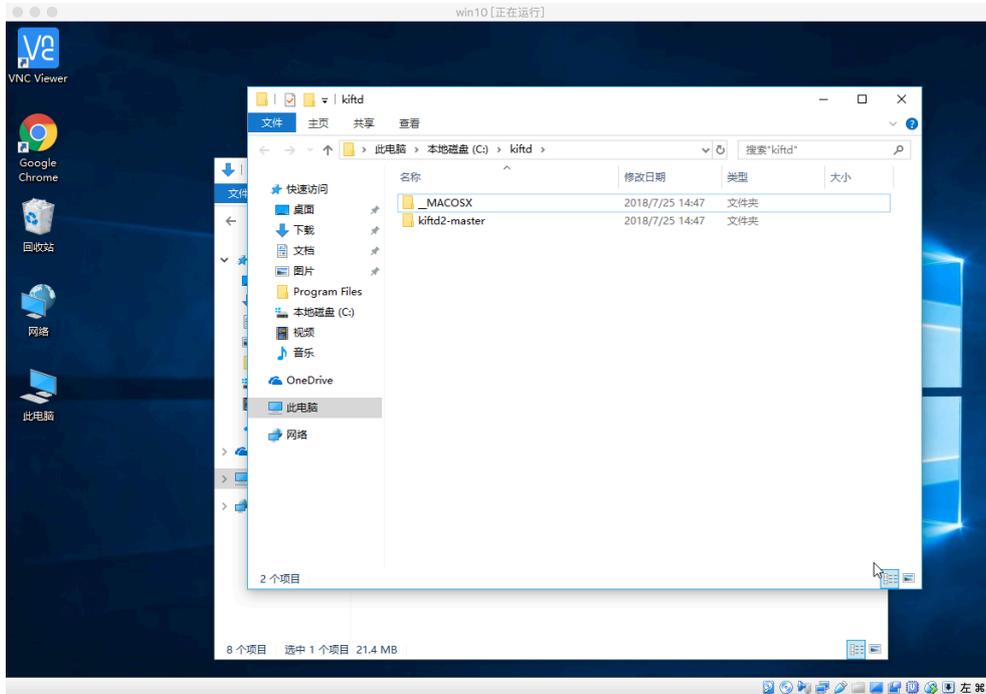
1.3 接下来，我们安装 java 运行环境：



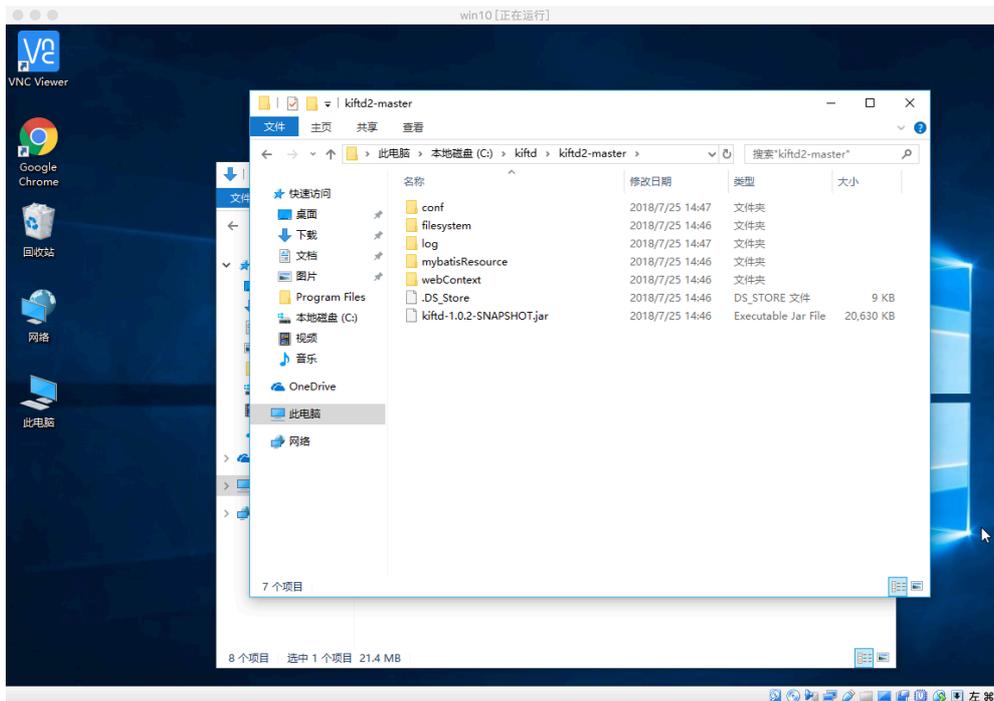
这个过程仅需您一路“下一步”即可。期间请不要中断安装，它很快就能结束。注：java 的默认安装位置为 `C:\Program Files\java`，当然，您也可以根据自己的喜好在安装过程中修改这一位置。

至此，java 运行环境已经准备完毕。

## 2，解压 kiftd 并运行

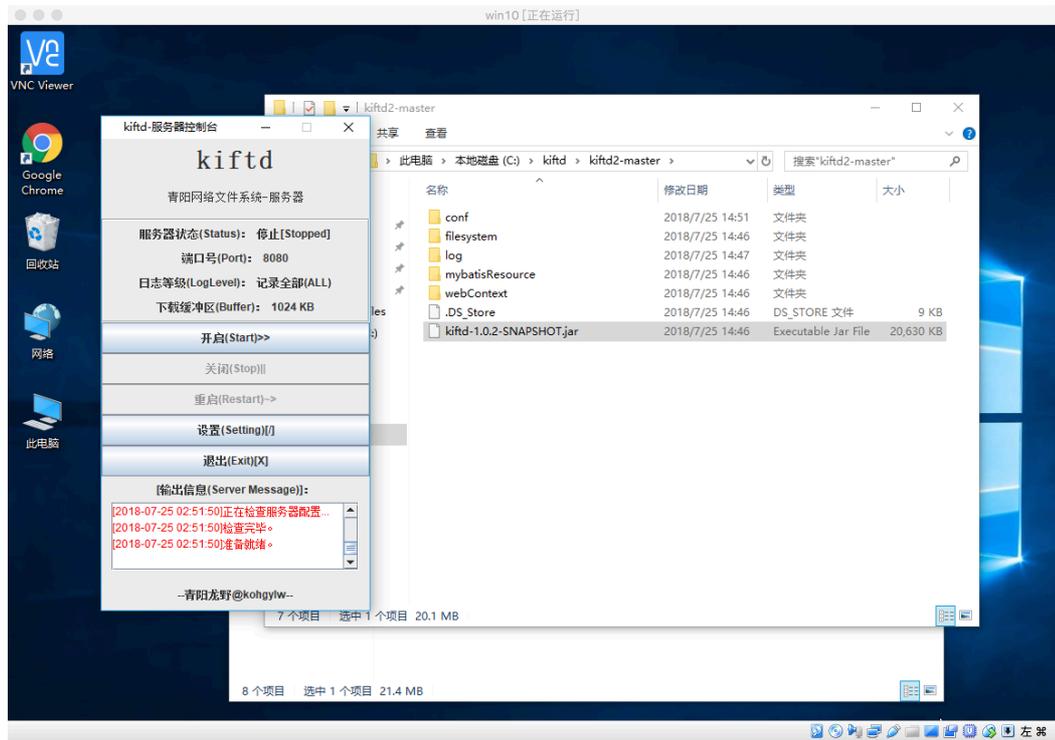


如图，本示例将其解压在 `C:\kiftd\kiftd2-master` 内，其中应包含以下文件：

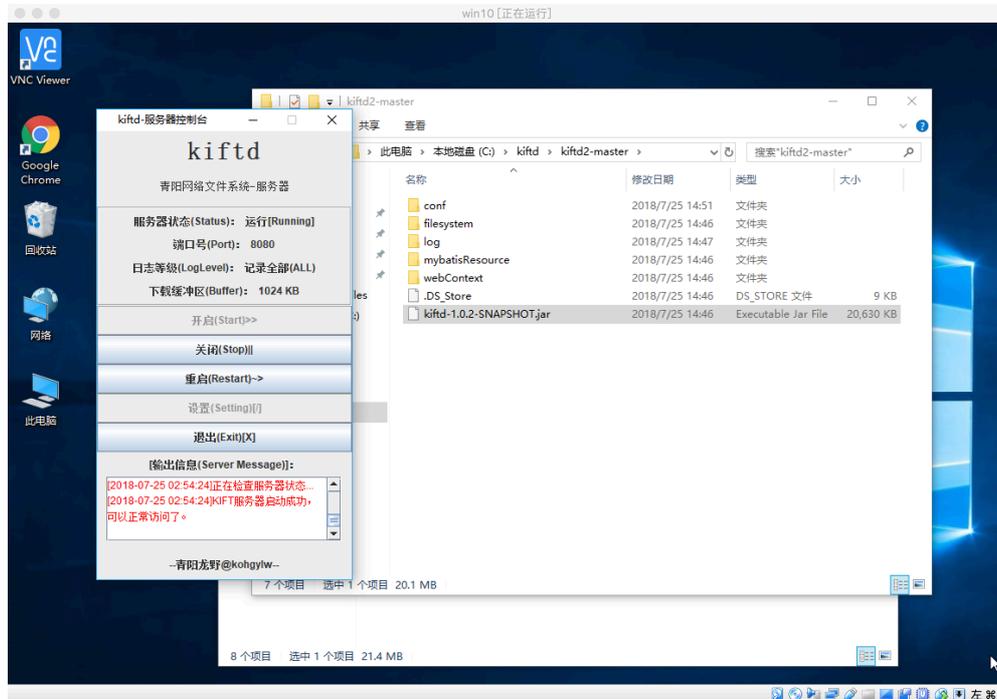


我们接下来会将该文件夹称作“kiftd 应用主目录”。在这个目录中，名为

“kiftd-1.0.2-SNAPSHOT.jar” 的文件即为 kiftd 应用程序。下面，我们双击运行它。注：jar 文件为 java 可执行文件类型。



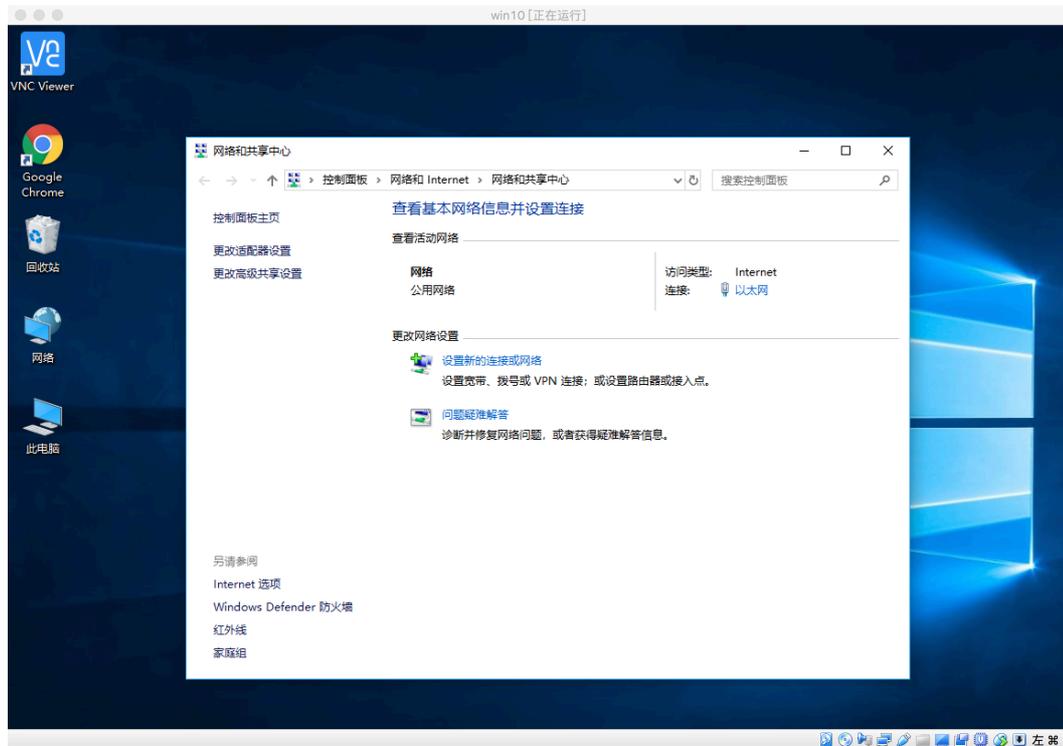
开启后，您能看到 kiftd 服务器的主界面，点击“开启”按钮即可启动服务器。如图所示：



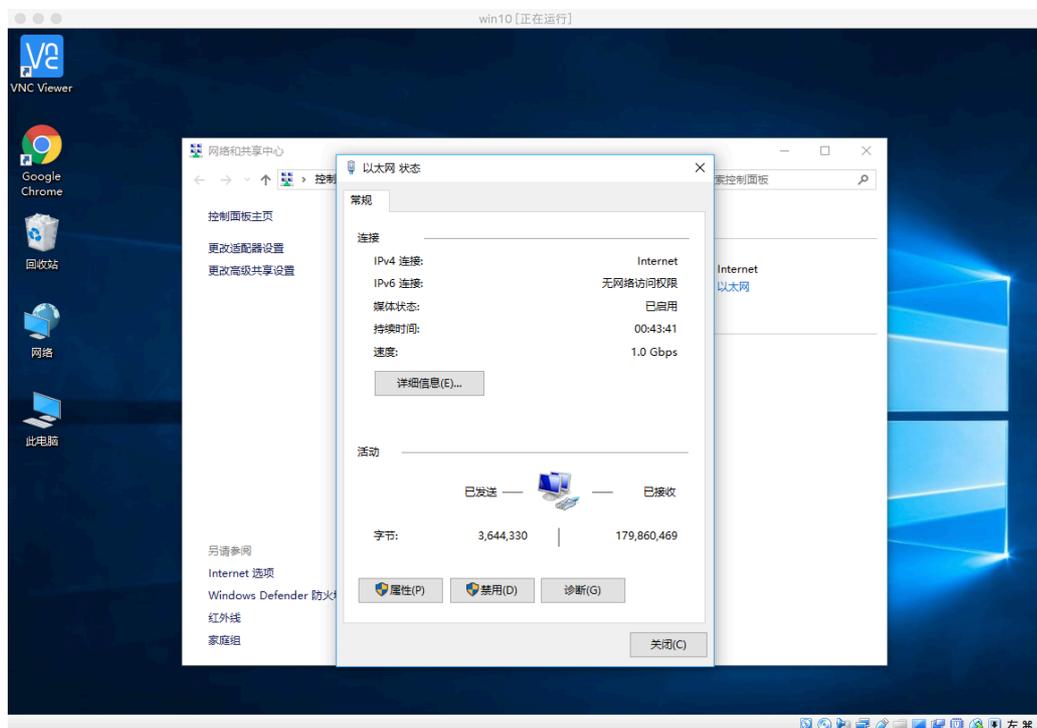
如果您看到的效果与上图一致，就说明服务器启动完成，接下来您就可以访问 kiftd 的主页面了。

### 3, 访问应用主页

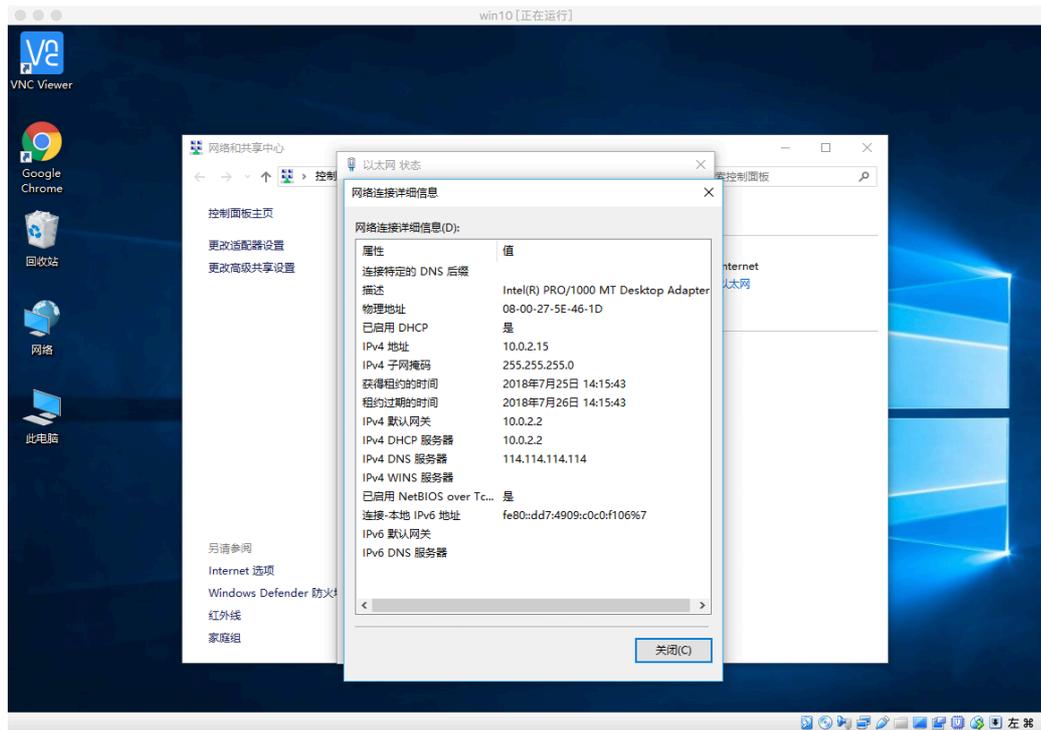
请打开“网络和共享中心”（对于大多数 windows 版本而言，您可以右键“网络”并点击“属性”打开这一界面。或者直接在控制面板找到“网络和共享中心”）：



选中您的网卡（例如图示中的“以太网”）：



在上图对话框中，点击“详细信息...”按钮：



在这里，您能看到您计算机的 IP 地址，也就是“IPv4 地址”后面的那一串数字。请记住这个数字，因为我们需要通过这个 IP 地址来访问本机上运行的 kiftd。

接下来，您需要打开浏览器（这里推荐使用 Chrome 或其内核的浏览器，对于某些浏览器而言，请找到右上方的 IE 图标，并切换为“极速模式”浏览），在地址栏输入：`http://{您的 IP 地址}:{端口号}/`，回车。端口号可在服务器主界面上查看。例如示例中需要输入 `http://10.0.2.15:8080/`。



现在，您已经进入和您的 kiftd 主页。同理，您也可以让与您处于同一网段

(链接在同一个 WIFI 下或者使用同一个路由器上网) 的人用这个地址访问您的 kiftD 了。

#### 4, 体验全部功能

现在, 您可以开始体验 kiftD 的全部功能了。在您第一次启动 kiftD 时, 它会自动生成一个默认的管理员账户, 账户名为 “admin”, 密码为 “000000” (当然, 您以后也可以修改它或创建其他账户), 请点页面右上方的 “登录” 按钮进行登录, 之后您就可以自由操作这个 “多功能网盘” 了。

至此, 您已经完全可以脱离《说明》来自己尝试如何使用 kiftD, 您可能希望先自己体验一会儿。不过, 如果您需要更加详细地了解所有功能, 那么您需要挑个时间继续阅读 “功能说明” 一章。

## 在 Linux/Unix(Mac OS X)系统下 3 分钟快速安装

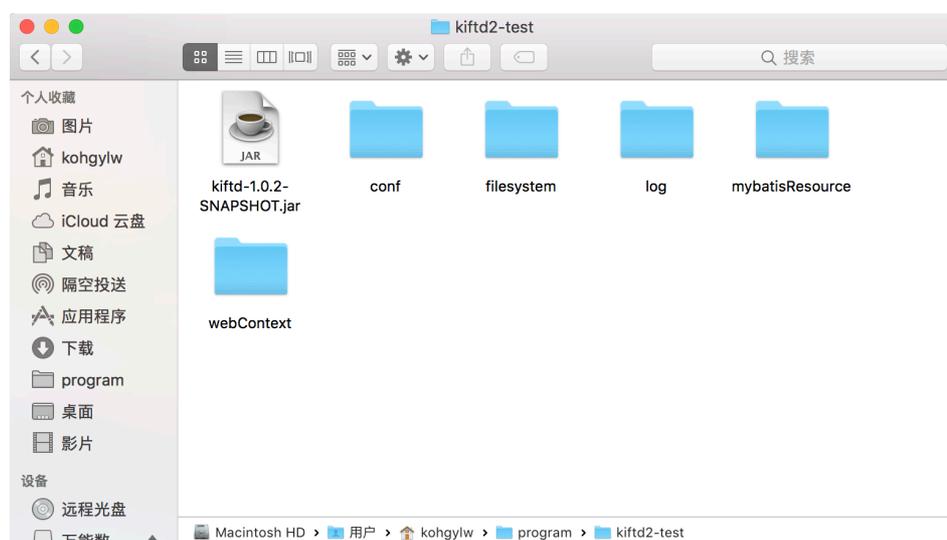
我一直觉得, 能使用 Unix/Linux 系统的人都不会是小白。所以这里就简单说一下过程:

### 1, 安装 java 运行环境

请到 java 官网下载一份并进行安装。对于 Mac OS X, 这个过程就是一路下一步。对于 Linux, 不同的发行版体验不太一样, 请亲自感受~ (总之就是安装个 java 运行环境)

### 2, 解压 kiftD 并运行

将下载好的 zip 压缩包解压在一个具备读写与运行权限的文件夹内。例如您的用户目录 (一般来说, 你的用户名是什么, 这个文件夹就叫什么) 下。



如上图，示例中将其解压在了~/program/kiftd2-test/下，接下来我们称这一文件夹为“kiftd 应用主目录”。

下面，对于 Mac OS X 用户来说，直接双击“kiftd-1.0.2-SNAPSHOT.jar”程序即可开启 kiftd 服务器了。而一种对于 Mac OS X 和 Linux 用户通用的做法是使用终端运行它：

(高能预警！示例的终端颜色可能比较刺眼)



```
kiftd2-test — java -jar kiftd-1.0.2-SNAPSHOT.jar — 80x24
kohgyiw:kiftd2-test kohgyiw$ java -jar kiftd-1.0.2-SNAPSHOT.jar
```

使用 cd 命令进入到 kiftd 应用主目录，使用如下命令以图形界面方式运行服务器：

```
java -jar kiftd-1.0.2-SNAPSHOT.jar
```

**注意**，对于 **Linux 用户**而言，您可能需要自己在操作系统中配置 java 的**环境变量**才能直接使用上述命令。一般来说，我们习惯将 java 安装目录下的 bin 文件夹加入到 path 变量中，具体方法请参见所用操作系统推荐的配置方式。

Mac OS X 用户请忽略上面这一条。

当然，有的时候您可能不希望使用图形界面控制 kiftd 服务器，尤其是对于使用云服务器或远程管理服务器的用户而言。所以 kiftd 也提供了命令操作的方式，具体信息请见“功能说明”一章的“使用命令方式控制 kiftd”一节。

无论如何，下面我们开启 kiftd 服务器：



如果您看到上图所示的效果，则代表 kiftd 服务器启动成功了。当然，这是图形界面方式的示例。

### 3，访问应用主页

下面我们需要使用本机的 IP 地址和端口号来访问应用主页。

3.1 对于 Mac OS X 用户而言，请打开  系统偏好设置，找到  网络：



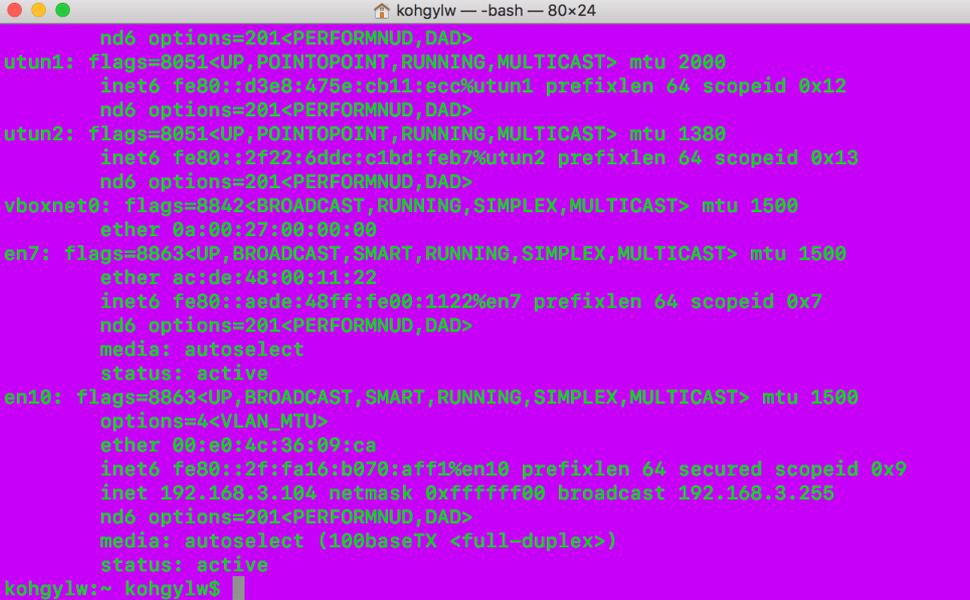
请记住“IP 地址”后面的那串数字，这边是本机的 IP 地址。稍后我们会使

用它访问本机 kiftd 的主页。

3.2 当然，还有一种 Mac OS X 和 Linux 用户可以通用的做法——在终端中输入以下命令：

```
ifconfig
```

之后您便可以看到您本机的 IP 地址：



```
kohgylw ~ -bash — 80x24
nd6 options=201<PERFORMNUD,DAD>
utun1: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 2000
inet6 fe80::d3e8:475e:cb11:ecc%utun1 prefixlen 64 scopeid 0x12
nd6 options=201<PERFORMNUD,DAD>
utun2: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1380
inet6 fe80::2f22:6ddc:c1bd:feb7%utun2 prefixlen 64 scopeid 0x13
nd6 options=201<PERFORMNUD,DAD>
vboxnet0: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 0a:08:27:08:00:00
en7: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether ac:de:48:00:11:22
inet6 fe80::aede:48ff:fe00:1122%en7 prefixlen 64 scopeid 0x7
nd6 options=201<PERFORMNUD,DAD>
media: autoselect
status: active
en10: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
options=4<VLAN_MTU>
ether 00:e0:4c:36:09:ca
inet6 fe80::2f:fa16:b070:aff1%en10 prefixlen 64 secured scopeid 0x9
inet 192.168.3.104 netmask 0xfffff00 broadcast 192.168.3.255
nd6 options=201<PERFORMNUD,DAD>
media: autoselect (100baseTX <full-duplex>)
status: active
kohgylw:~ kohgylw$
```

下面，请打开浏览器（推荐使用 Chrome 或其内核的浏览器。对于某些浏览器而言，请点击右上角的 IE 图标，之后选择“极速模式”浏览），在地址栏中输入：`http://{您的 IP 地址}:{端口号}/`，回车。端口号可在服务器主界面查看，例如示例需要输入：`http://192.168.3.104:9000/`。



现在，您已经进入 kiftd 的主页了（这是我自用的，所以会有一些文件，初始情况下这个列表是空的）。同理，您也可以让其他与您处在同一网段的人使用这一地址来访问您的 kiftd 主页。

#### 4，体验全部功能

在 kiftd 第一次启动时，会自动生成一个管理员账户，账户名为“admin”，密码为“000000”（当然，您之后可以修改它或者创建其他账户），请点击右上角的“登录”按钮进行登录，之后您就可以自由操作这个“多功能网盘”了。

至此，您已经完全可以脱离《说明》来自己尝试如何使用 kiftd，您可能希望先自己体验一会儿。不过，如果您需要更加详细地了解所有功能，那么您需要挑个时间继续阅读“功能说明”一章。

## 功能说明

这里是功能说明——虽然你能够通过摸索学会绝大多数 kiftd 的操作方法，但是还有一些功能不是那么容易理解。所以，闲话少叙，这一章主要介绍哪些不太容易理解的功能和设置。

### 自定义端口号

当您第一次使用 kiftd 时，它会以默认端口——8080 来启动服务器。所谓端口号，就是他人访问您的地址时用来指定和谁“通话”的，操作系统中每个端口只能被一个程序占用，先到先得。这就导致了有些时候您可能无法通过默认端口启动，或是希望更换为一个自己喜欢的端口号（例如 6666）。下面将介绍如何设置自定义端口号：

#### **1，如果您使用图形界面模式启动：**

这个设置十分简单，请在**服务器关闭状态**下点击主界面上的“设置”按钮，之后在端口一栏输入您想要的端口号：



之后点击“应用”按钮。

## 2, 如果您不使用图形界面模式启动：

这种情况下，您也可以通过直接修改配置文件来设置端口号。请进入 kiftfd 应用主目录，打开 conf 文件夹，找到 **server.properties** 文件（下同）。使用任意的文本编辑器打开这一文件（注意，该文件的换行符使用 Unix 规范，**这会导致在 Windows 下格式不正确**），在其中找到 port 一行，将等号后面的数值改为您想要设定的端口号，保存即可。上述操作须在服务器关闭的状态下进行。

### 端口号修改完成后：

您就可以启动服务器并使用新的端口号进行访问了。例如当您的 IP 为 192.168.0.2 且端口号改为 9000 时，则您的 kiftfd 主页访问地址为 <http://192.168.0.2:9000/>。

注意：对于 **Linux/Mac OS X** 用户而言，如果需要使用 1024 以下的端口号启动服务器时，必须获得管理员权限。也就是说，需要在终端使用 **sudo** 命令运行 kiftfd 才能够以 80 端口启动 kiftfd 服务器。

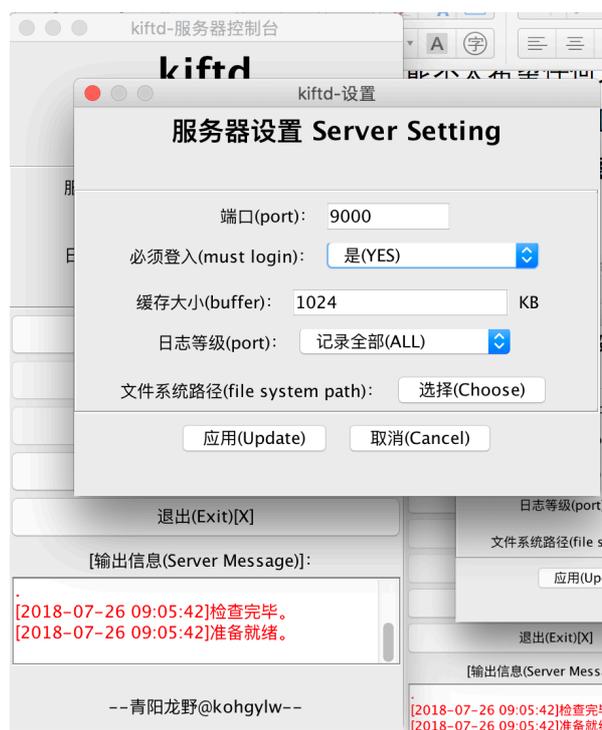
提示：如果将端口号设置为 **80**，则在访问主页时无需再声明端口号，直接输入 IP 地址即可访问。例如，当本机 IP 为 192.168.0.2、且 kiftfd 端口设置为 80 时，其他人进行访问只需要输入 <http://192.168.0.2/> 即可进入主页了。

## 设置“必须登入”功能

有些时候,您可能需要将 kiftd 作为一个私人云存储平台使用。这种情况下,您可能不太希望任何人都能直接查看您的文件列表。此时,您可以选择开启“必须登入”功能来限制匿名访问:

### 1, 如果您使用图形界面启动:

同上,仅需要将“必须登入”设置改为“是”,之后点击应用并启动服务器即可。



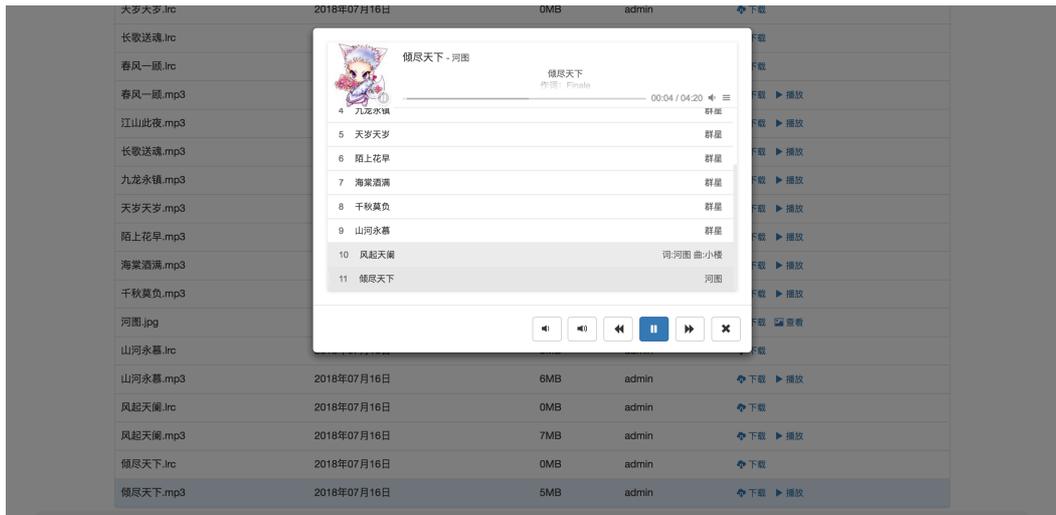
### 2, 如果您未使用图形界面启动:

请将配置文件中的“mustLogin”项改为“N”即可(N=NO,O=OFF)。

## 各种预览与播放功能

目前, kiftd 支持以下几种多媒体在线体验功能:

### 1, 在线音乐播放功能



可以进行下载的账户均可使用本功能。目前支持的音乐格式为 MP3 (推荐) 和 OGG，其中仅有 MP3 格式能够自动解析歌手和歌曲信息。当您把多个音频文件放在同一文件夹时，点击其中任意一个都会打开音乐播放组件并以列表形式开始播放。如果需要显示歌词，请在同级文件夹内上传 **LRC 歌词**，**LRC 文件名与音频文件名完全一致**即可被自动解析。同理，如果需要显示歌曲封面，仅需再在同级文件夹内上传图片 (png/jpg/jpeg/img)，图片名与其**音频文件名或解析出来的歌手名完全一致**即可被自动显示。

(注：展示音乐来源于网络，版权归其作者所有)

## 2，图片在线预览功能



可以进行下载的账户均可使用本功能。目前支持的图片格式为：img/png/jpg/jpeg，同级文件夹内的所有图片可以进行依次预览。同时，图片查看器还支持翻转、旋转、缩放、全屏……

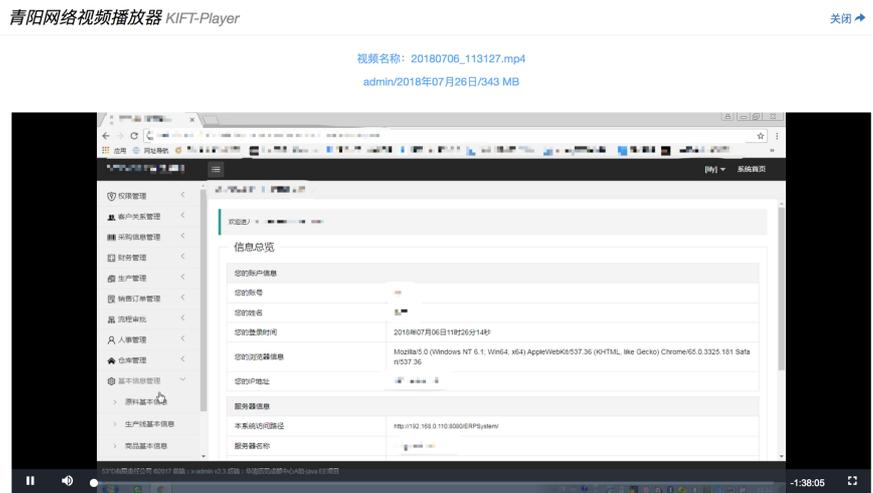
(注：展示图片来自网络，版权归其作者所有。)

### 3, PDF 在线预览功能



可以进行下载的账户均可使用本功能。PDF 文档可以直接在线预览，功能全面。

### 4, 视频在线播放功能



可以进行下载的账户均可使用本功能。支持的格式为：MP4（推荐）、OGG。功能同样全面。

## 批量操作

您可以通过点击文件行来选中某一文件，也可以按住 **Shift** 键选取多个文件，甚至是通过点击文件列表的表头选中文件夹内的全部文件。

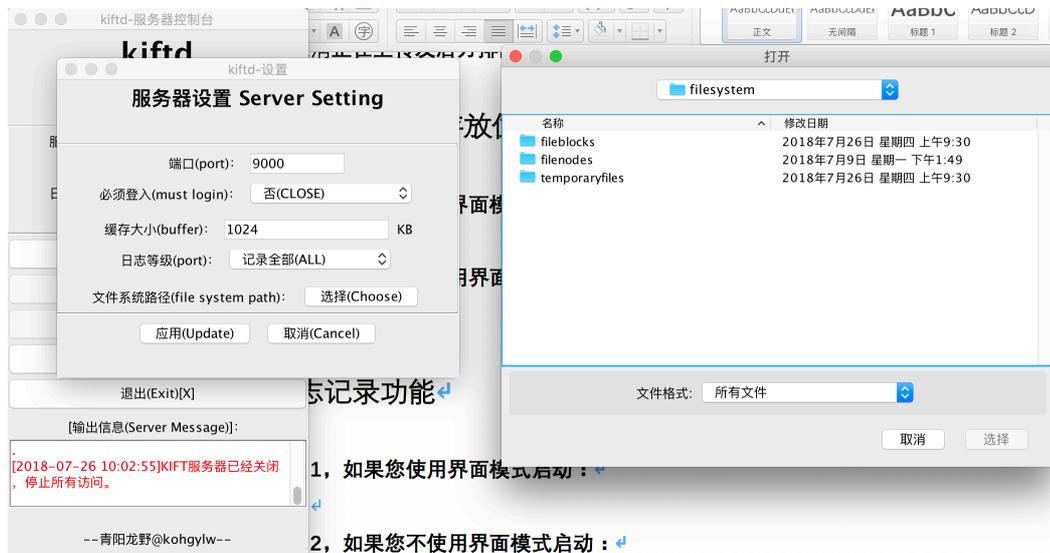
选取若干文件后，您可以使用“打包下载”（需要下载权限）和“批量删除”

(需要删除权限) 功能。

当然, 您还可以一次上传多个文件, 在上传文件对话框内选中多个文件即可一次性上传它们。当然, 如果您中途取消, 则已经完成上传的文件不会受到影响, 仅取消正在上传及后方排队的文件。

## 修改文件系统存放位置 (或切换文件系统)

### 1, 如果您使用界面模式启动 :



您可以点击文件系统路径后面的“选择”按钮来更改文件系统的存放位置。它默认存放在 kift 应用主目录下的 filesystem 文件夹内。如果您希望换一个位置, 可将其拷贝到您想要的任意位置 (运行 kift 的系统账户必须对其具备读写权限), 之后再用这一功能选中它。例如: 放置在 D:\后, 点击“选择”按钮, 选中 D:\filesystem 文件夹, 点击选择按钮并应用。

### 2, 如果您不使用界面模式启动 :

将 kift 应用主目录下的 filesystem 文件夹拷贝到任意位置后, 再次找到配置文件, 将 FS.path 一项后面的参数改为您想要的路径。例如 /User/foo/bar/filesystem。注意, 启动 kift 的账户必须对该路径具备读写权限, 对于 Mac OS X 系统或 Linux 系统, 建议将其放置在用户目录内的位置以避免权限问题。如果您希望使用默认位置 (kift 应用主目录内的 filesystem 文件夹), 可将该值设置为“DFEAULT”一词。

### 提示 :

当然, 您也可以使用该方法将另一个 kift 的文件系统文件夹设置为自己的

kiftd 的文件系统，这样您就可以直接拥有其内部的所有文件资源，该文件系统的文件夹仅在 v1.0.2 及以上版本适用。这一操作让 kiftd 需要迁移或升级时变得很简单。

## 日志记录功能

日志功能能够方便您了解 kiftd 的操作历史——包括**错误信息**和**文件操作记录**（例如谁在什么时候删除了哪个文件）。该日志根据不同需要分为三个等级：**事件级**（记录错误信息及文件操作记录）、**异常级**（仅记录错误信息）、**无**（不生成日志），您可以通过设置自由选择一个等级来进行日志管理。

### 1, 如果您使用界面模式启动：



直接将日志等级调整为您需要的并应用即可。

### 2, 如果您不使用界面模式启动：

请找到配置文件，并将 log 一项的设置改为“E”（事件级）、“R”（异常级）、“N”（无）三种之一，之后保存并启动服务器。

### 3, 如何读取日志：

日志文件会被生成在 kiftd 应用主目录下的 log 文件夹内。它们按照时间命名，且以 klog 为后缀。您可以使用任意文本编辑器打开并读取它们（由于日志文件使用的编码格式为 UTF-8，且换行符为 Unix 风格，因此在 Windows 下打开可能会导致格式混乱，这时您可以选用一些**高级文本编辑器**打开，例如 World，

并将编码格式设置为**其他-“UTF-8”**，行终止符号选为**“仅回车符”**）。

每一个日志段落均代表着一件事，分为 TIME(何时)TYPE(何事)和 CONTEXT(具体内容)三部分。

例如下列日志段落文本就表示在 2018 年 07 月 26 日的 9 点 28 分 44 秒，一个叫做 admin 的账户上传了一个叫做“kiftd 说明文档.pdf”的文件：

```
TIME:
2018年07月26日 09:28:44
TYPE:
Event
CONTENT:
>ACCOUNT [admin]
>OPERATE [Upload file]
>PATH [ROOT]
>NAME [kiftd说明文档.pdf]
```

相信您能很容易读懂他们。

**提示：**

仅在日志等级为“事件级”的时候，kiftd 才会记录文件操作信息，异常级和无均不会记录它们。另外，日志文件不会自动清理(毕竟日志是一个重要的资料)，因此您需要去判定并手动删除 log 文件夹内那些不再需要的日志文件。

## 设置用户与权限

用户及权限的设置是使用文件方式进行设置的。您会发现这个方法如此简单以至于很快就能学会它：

首先，请在 kiftd 应用主目录下找到 conf 文件夹，之后使用文本编辑器打开名为“account.properties”的文件(该文件以 UTF-8 编码格式生成，且换行符为 Unix 风格，因此在 Windows 系统下，请使用 World 进行打开，其中编码格式要选为“其他”：UTF-8，行终止符号要选为“仅回车符”)。

该文件会在**第一次启动 kiftd 时**自动生成，其中包含了默认管路员账户 admin 的设置以及默认全局权限 authOverall 的设置。

一个用户的设置格式为(每个设置独占一行，以全英文输入法进行输入)：

{用户名}.pwd={该用户的密码}

{用户名}.auth={该用户的权限}

权限可以使用这些标识符进行设定：

- “c”：创建文件夹
- “u”：上传文件
- “d”：删除文件或文件夹
- “r”：重命名文件或文件夹
- “l”：下载文件

例如，我们新建如下设定：

```
Kohgylw.pwd=123456
```

```
Kohgylw.auth=u
```

那么我们就声明了一个名为“Kohgylw”的账户，登录密码为“123456”，其权限除了全局权限定义的部分外，还额外拥有“上传”权限。

**全局权限“authOverall”**定义了所有用户（包括未登录的用户）都具备的权限，默认为“l”（下载文件）。

您可以根据上述示例自有创建新的账户或修改已有的账户了。

## 修改缓存大小

### 1，如果您使用界面模式启动：



如图所示，请将缓存大小这一设置改为一个您希望的数值。当您的内存较小时，请适当调低该值。该值必须大于 0 且为整数。

### 2，如果您不使用界面模式启动：

请将配置文件中的 `buff.size` 一项设置为一个大于 0 的整数。该数值以 byte 为单位 (1KB=1024byte)，当您的内存较小时，请适当调低该值。

## 纯命令模式

对于一些需要远程管理 Linux 服务器的用户而言，图形界面模式显然并不是个好选择，这时候 `kiftd` 的纯命令模式将会派上用场。

使用以下命令启动 `kiftd`，从而开启命令模式：

```
java -jar kiftd-1.0.2-SNAPSHOT.jar -console
```

在这个模式下，您可以直接使用命令来控制服务器，具体操作请根据提示进行。例如开启服务器需要输入 “-start” 并回车。

这一过程中请勿关闭终端，否则将导致程序结束。

提示：实际上，纯命令模式可以获得更高的效能，毕竟这样就无需让 `kiftd` 加载图形资源。但是切记命令模式和图形界面模式同时使用，也不要同时开启两个相同的 `kiftd` 服务器。

## 安全性与维护

`Kiftd` 在设置之初就是认为其运行环境是绝对安全的，因此其所有安全措施均针对外来破坏，而不考虑内部破坏。这意味着，您必须确保运行 `kiftd` 的平台足够安全可靠。

虽然 `kiftd` 对于内部存储的文件进行了初步的信息混淆，但那仍非加密手段，也意味着如果您将 `kiftd` 运行在一个不可靠的服务器运营商提供的资源上，那么就可能出现诸如其工作人员擅自查看您网盘上的文件并进行盗用的风险（只要他的时间充足，混淆这种迷惑手段是很容易被攻破的）。因此，最为安全的做法是将其运行在一个由自己管理的平台上，并做好杀毒工作。倘若您有必要将其托管至第三方，则应尽量确保其管理的规范性。

`Kiftd` 的所有文件资源均存放于“文件系统”中，它在操作系统里表现为一个文件夹（该路径可自定义）。请勿随意修改该文件夹内的任何内容，否则可能造成文件丢失。

请勿修改应用主目录下“`mybatisResource`”和“`webContext`”文件夹内的内

容，它们是 kiftD 运行的必要资源，其修改可能造成 kiftD 无法启动。

请务必确保 “conf”、“log”、“mybatisResource”、“webContext” 4 个文件夹与 kiftD 应用主程序位于同一目录下，且该目录具备读写和运行权限。

**如需卸载 kiftD**，仅需将 kiftD 应用主目录完全删除即可（如将文件系统放置在其他位置，则也需要一并删除），这样即可将其从操作系统中完全清除。

提示：**卸载前要确保已将 kiftD 内存储的所有重要文件全部下载至本地**，否则您丢失它们。

## 许可声明

使用 kiftD 或阅读 kiftD 的源代码则视为接受以下许可：

- 1, 任何个人或组织均有权利无偿获取并使用 kiftD 的原版拷贝或其源代码，以及将其运用于任何用途，可以是商业性质的。
- 2, kiftD 的作者用于其著作权，任何使用者均需对其进行承认。
- 3, 任何组织或个人均有权阅读并修改 kiftD 的源代码、将其重新发布、运用于任何用途。
- 4, 作者无需承担他人获取并使用 kiftD 及其源代码所造成的任何直接及间接后果。

## 联系作者/希望帮助 kift 做的更好？

- 如果您在使用中发现的任何 Bug，请马上提交到：

<https://github.com/KOHGYLW/kiftD/issues>

您可以点击该页面上的  按钮编写并写您遇到 Bug 的经历。作者会定期浏览这里并尽快解决该问题。

- 需要直接联系作者？请发送电子邮件至：

[Kohgylw@163.com](mailto:Kohgylw@163.com)

# 特别鸣谢



以及 : APlayer、ZT-zip、viewer.js

## 开发小记

**关于 v1.0.2 源代码的声明**：在 kifted v1.0.2 开发的最后阶段，作者决定将其发布到 github 并覆盖旧版本。由于 kifted v1.0.2 是使用新的项目进行构建的，因此作者也需要重建本地版本仓库进行提交：这一过程中，作者需将新的版本仓库建立好并把 Eclipse 中的项目提交到其中，之后进行 push 操作。然而由于网络原因，这一过程未能成功，于是作者决定先放一放。之后，作者又觉得既然不再 push 了，就把**刚刚建立的版本仓库删掉了，还清空了垃圾桶**（处女座，习惯）。这个时候，作者再回到 Eclipse 工作空间找原项目，发现它竟然神奇地不见了。后来作者得知一个不幸的消息：原来当选定本地仓库位置时，Eclipse 会**自动**将工作空间中的项目**移动到那个本地仓库中，且不在工作空间中备份**。

就这样，作者写了将近 3 个月的东西全部丢失了。且作者未能在自己的电脑上找到给源代码的第二份备份。

至此，kifted v1.0.2 在这个世界上就只剩下了**编译好的版本**，再无源代码。

目前，作者在痛定思痛后，决定重头开始构建 kifted 并将其作为 kifted v1.1.0 发布。

当然，塞翁失马——这也给了作者重新对之前的一堆烂代码进行重构的机会。而且这一次，作者绝不会再信步由缰地写了，争取完成一个结构更加严谨、规范和易用的 kifted。

（注：作者十分欢迎想要加入这一项目的爱好者们！他们可与 kohgylw@163.com 直接联系，并直接对其称呼青阳龙野这个听起来有点中二其实一点也不中二的名字）。

@青阳龙野

文档编写 2018-7-25