

云存储与云备份中间件 MiddlewareSDK 使用说明

一. 运行所需环境

环境	版本号
Ubuntu	14.04
Eclipse	4.6.0 (Neon)

二. 部署使用步骤

1. 安装配置 Eclipse (如果已安装可跳过此步骤)

1.1 安装配置 JDK

- (1) 从官网下载 jdk: `jdk-8u101-linux-x64.tar.gz` (其他版本 jdk 也可以)。
- (2) 解压: `$ tar jdk-8u101-linux-x64.tar.gz`, 解压出来是一个 `jdk1.8.0_101` 文件夹。
- (3) 将文件夹移动到安装目录: `$ sudo mv jdk1.8.0_101 /usr/local/`
- (4) 配置环境变量, 更改 `/etc/profile` 文件。
`sudo gedit /etc/profile`
在文件最后加上如下几行:
`export JAVA_HOME=/usr/local/jdk1.8.0_05`
`export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$PATH`
`export CLASSPATH=$CLASSPATH:.$JAVA_HOME/lib:$JAVA_HOME/jre/lib`
- (5) 保存退出, 在命令行输入 `java -version` 查看 java 版本。

1.2 下载安装 eclipse

- (1) 从官网(<http://www.eclipse.org/downloads/>)下载 Eclipse IDE for Java EE Developers 的 Linux 版本。
- (2) 解压: `$ tar -zxvf eclipse-SDK-3.7.2-linux-gtk.tar.gz`
- (3) 将文件夹移动到安装目录: `$ sudo mv eclipse /usr/local/`
- (4) 启动 eclipse: `$ /usr/local/eclipse/eclipse`

2. MiddlewareSDK 使用

2.1 创建 java 工程

File->New->Java Project->Project Name->Finish

2.2 部署 MiddlewareSDK.jar

- (1) 创建 lib 文件夹: 工程右键->New->Folder->Folder Name(lib)->Finish
- (2) 添加 MiddlewareSDK.jar: 复制 MiddlewareSDK.jar->右键 lib->Paste
- (3) Build Path: 工程右键->Build Path->Configure Build Path->Libraries->Add Jars->Project Name->lib->MiddlewareSDK.jar->Apply->OK
- (4) 实例化 MiddlewareAPI 对象:
`MiddlewareAPI mwAPI = new MiddlewareAPI(kaeyika@163.com, "123456");`
- (5) 调用 MiddlewareAPI 接口: `mwAPI`
`String result = middlewareAPI.getContainer();`

三. 接口说明

MiddlewareAPI		
返回类型	接口及参数	接口描述
String	getAccessToken(String userID, String pwd) <i>userID:ID; pwd: 密码</i>	获取 access token
String	String getContainer()	获取容器列表
String	createContainer(String containerName) <i>containerName: 容器名</i>	创建容器
String	createDirectory(String directoryPath) <i>directoryPath: 目录路径及名称</i>	创建目录
String	createSymlink(String srcPath, String dstPath) <i>srcPath: 源路径; dstPath: 目标路径</i>	创建符号链接
String	getRecycler(long start, long limit) <i>start: 开始位置; limit: 数量</i>	获取回收站文件
String	getFileList(String directoryPath, boolean recursive) <i>directoryPath: 目录路径; recursive: 是否递归查询</i>	获取文件列表
String	getFileHistory(String filePath) <i>filePath: 文件路径</i>	获取历史文件
String	getFileAttribute(String filePath, String version) <i>filePath: 文件路径; version: 版本</i>	获取文件属性
String	getOpHistory(int recent) <i>recent: 数量</i>	获取历史操作记录
String	getHistoryContainer(String containerName) <i>containerName: 容器名</i>	获取历史容器
String	getOpTask(String taskId) <i>taskId: 任务 ID</i>	获取操作任务
String	getQuota()	获取配额信息
String	clearRecycler()	清空回收站
String	setQuota(long quotaValue) <i>quotaValue: 配额量</i>	设置配额
String	setPermission(String path, long permission) <i>path: 文件路径; permission: 权限</i>	设置权限
String	setHistoryContainer(String containerName) <i>containerName: 容器名</i>	设置历史容器
String	String deleteContainer(String containerName) <i>containerName: 容器名</i>	删除容器
String	deleteDirectory(String directoryPath, boolean cover) <i>directoryPath: 目录名; cover: 是否覆盖</i>	删除目录
String	deleteFile(String filePath, boolean cover)、 <i>filePath: 目录名; cover: 是否覆盖</i>	删除文件

String	deleteOpHistory(int recent) <i>recent: 次数</i>	删除历史记录
String	uploadFile(String srcPath, String dstPath, boolean overwrite, String metadata, MODE mode) <i>srcPath: 源路径; dstPath: 目标路径; overwrite: 是否覆盖; metadata: 元数据; mode: 模式</i>	上传文件
String	downloadFile(String srcPath, long offset, long length, String version, MODE mode, String dstPath) <i>srcPath: 源路径; dstPath: 目标路径; offset: 偏移量; length: 长度; version: 版本; mode: 模式</i>	文件下载
String	moveDirectory(String srcPath, String dstPath) <i>srcPath: 源路径; dstPath: 目标路径</i>	转移目录
String	moveFile(String srcPath, String dstPath) <i>srcPath: 源路径; dstPath: 目标路径</i>	转移文件
String	copyDirectory(String srcPath, String dstPath, boolean async) <i>srcPath: 源路径; dstPath: 目标路径; async: 是否同步</i>	复制目录
String	copyFile(String srcPath, String dstPath, boolean async) <i>srcPath: 源路径; dstPath: 目标路径; async: 是否同步</i>	复制文件
String	renameDirectory(String srcPath, String dstPath) <i>srcPath: 源路径; dstPath: 目标路径</i>	重命名目录
String	renameFile(String srcPath, String dstPath) <i>srcPath: 源路径; dstPath: 目标路径</i>	重命名文件
String	recoverDirectory(String uuid, String path) <i>uuid: 目录标识; path 目录路径</i>	恢复目录
String	recoverFile(String uuid, String path) <i>uuid: 文件标识; path 文件路径</i>	恢复文件