

Вадим Салаватов

Реализация мультиплатформенного доступа к файловым хранилищам на языке Kotlin

Выпускная квалификационная работа

Научный руководитель: профессор, д.ф.-м.н. А. С. Куликов

ДАТА ЗАЩИТЫ



Факультет математики и компьютерных наук СПбГУ
Программа «Современное программирование»

Введение в предметную область

- Kotlin Multiplatform
- Файловые системы — платформоспецифичны
- Хотим писать код в общем модуле
- Как работать с файловыми хранилищами в веб-браузере?

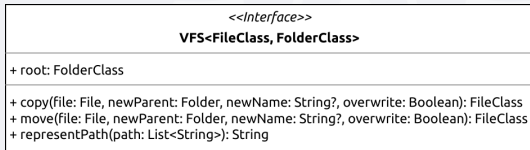
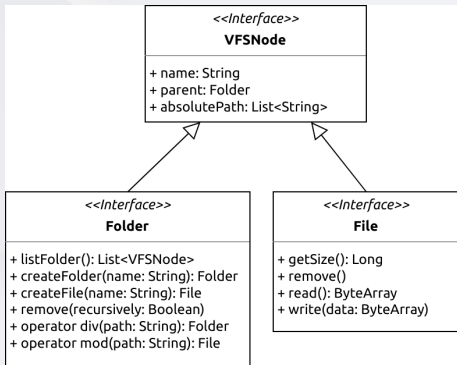


Постановка задачи

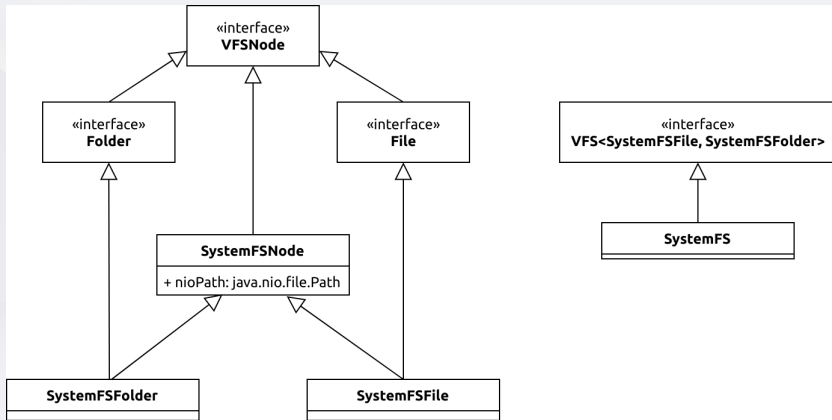
1. Разработать мультиплатформенную библиотеку для работы с файловыми хранилищами
2. Поддерживать платформы JVM, Android, JS (browser)
3. Поддерживать как минимум одно облачное хранилище
4. Обеспечить простую расширяемость как в плане поддержки новых хранилищ, так и в плане предоставляемой функциональности



Архитектура библиотеки



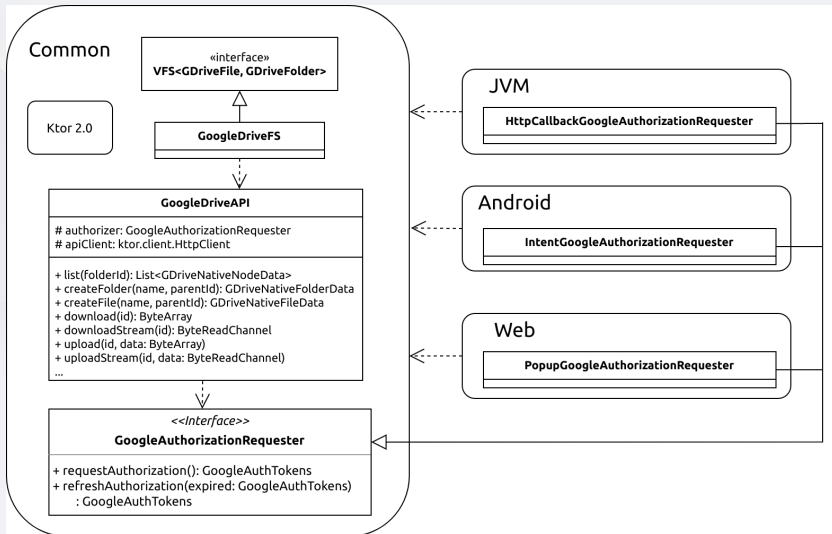
SystemFS



доступно на JVM и Android



GoogleDriveFS



SqliteFS

- SQL-эмуляция файловой системы
- Две таблицы: папки и файлы, у файлов есть бинарное содержимое (blob)
- Доступно только на Android



Инварианты и гарантии

- Родительская папка корня — сам корень
- Все методы бросают исключения только типов-наследников `VFSException` (и, возможно, помеченными некоторыми другими метками)
- Удаление непустой папки только с установленным флагом `recursively`
- Глобальных ограничений на имена файлов и папок не накладывается — это может контролировать пользовательский код



Как этим пользоваться. Общий код

```
typealias Storage = VFS<out File, out Folder>
suspend fun printFiles(folder: Folder) {
    for (node in folder.listFiles()) {
        when (node) {
            is File -> println("${node.name} at ${node.absolutePath}")
            is Folder -> printFiles(node)
        }
    }
}

suspend fun printStorage(fs: Storage) {
    printFiles(fs.root)
}
```



Как этим пользоваться. Клиентский код

```
val creds: GoogleAppCredentials = ...
suspend fun main() {
    val systemfs = SystemFS(
        Paths.get(".") // корень - рабочая директория
    )
    val googleAuth = HttpCallbackGoogleAuthorizationRequester(
        creds, GoogleDriveAPI.Companion.DriveScope.General
    )
    val gdrivefs = GoogleDriveFS(GoogleDriveAPI(googleAuth))

    printStorage(systemfs)
    printStorage(gdrivefs)
}
```



Расширения

<<Interface>>

StreamingIO

+ readStream(): ByteReadChannel
+ writeStream(data: ByteReadChannel)

SystemFS и GoogleDriveFS реализуют потоковое чтение и запись



Как этим пользоваться. Расширения

```
// typealias Storage = VFS<out {File & StreamingIO}, out Folder>
suspend fun <T> upload(fs: VFS<out T, out Folder>)
    where T : File, T : StreamingIO {
    val file = (fs.root / "folder" / "subfolder" % "file.txt") as T
    val data = ByteReadChannel(
        ByteArray(10_000_000) { it.toByteArray() }
    )
    file.writeStream(data)
}
...
upload(systemfs)
upload(gdrivefs)
// upload(sqlitefs)
```



Сводная таблица поддерживаемых возможностей

Хранилище/Платформа	JVM	Android	Web
SystemFS ¹	+	+	
GoogleDriveFS ¹	+	+	+
SqliteFS		+	

¹есть поддержка StreamingIO



Приложения на основе библиотеки

- Multieditor — мультиплатформенный текстовый редактор, работающий на платформах JVM, Android, Web
- gdrive-cli — интерфейс командной строки для работы с содержимым Google Drive, позволяет скачивать и загружать файлы в потоковом режиме



Контакты

Вадим Салаватов, vmsalavatov@gmail.com

<https://github.com/vsalavatov/multifs>

