Запуск реального андроид приложения на iOS и в браузере

Константин Цховребов
JetBrains

Запуск реального андроид приложения на iOS и в браузере

Константин Цховребов
JetBrains

Часть #1

Константин Цховребов

https://telegra.ph/Terrakok-05-01

2010 - 2020: Android разработчик

2020 - ... Kotlin Multi Platform

Мотивация

Теория:

- Официальная документация.
- Видео с конференций и митапов про KMP.

Практика:

- Видео с конференций и митапов.
- Статьи на Habr, Medium и так далее.

1. Пишем Hello World
2. …
3. PROFIT (теперь вы можете написать любое приложение и запустить его везде, где захотите)

1. Выделяем из проекта кусочек логики в KMP модуль
2. …
3. PROFIT (теперь вы можете перенести любое приложение и запустить его везде, где захотите)

Большинство туториалов в интернете

GitFox

- Не хочется выдумывать проект из воздуха

https://gitlab.com/terrakok/gitlab-client

GitFox

- Не хочется выдумывать проект из воздуха
- Проверить гипотезу, что **очень типичный андроид проект можно завести на иОС**

https://gitlab.com/terrakok/gitlab-client

GitFox

- Не хочется выдумывать проект из воздуха
- Проверить гипотезу, что **очень типичный андроид проект можно завести на иОС**
- реально проверить все старания над архитектурой

https://gitlab.com/terrakok/gitlab-client

GitFox

- Не хочется выдумывать проект из воздуха
- Проверить гипотезу, что **очень типичный андроид проект можно завести на иОС**
- реально проверить все старания над архитектурой
- Собрать проблемы и опыт и поделиться им внутри JetBrains (и исправить)

https://gitlab.com/terrakok/gitlab-client

GitFox

- Не хочется выдумывать проект из воздуха
- Проверить гипотезу, что **очень типичный андроид проект можно завести на иОС**
- реально проверить все старания над архитектурой
- Собрать проблемы и опыт и поделиться им внутри JetBrains (и исправить)
- Дать обществу разработчиков недостающий элемент

https://gitlab.com/terrakok/gitlab-client

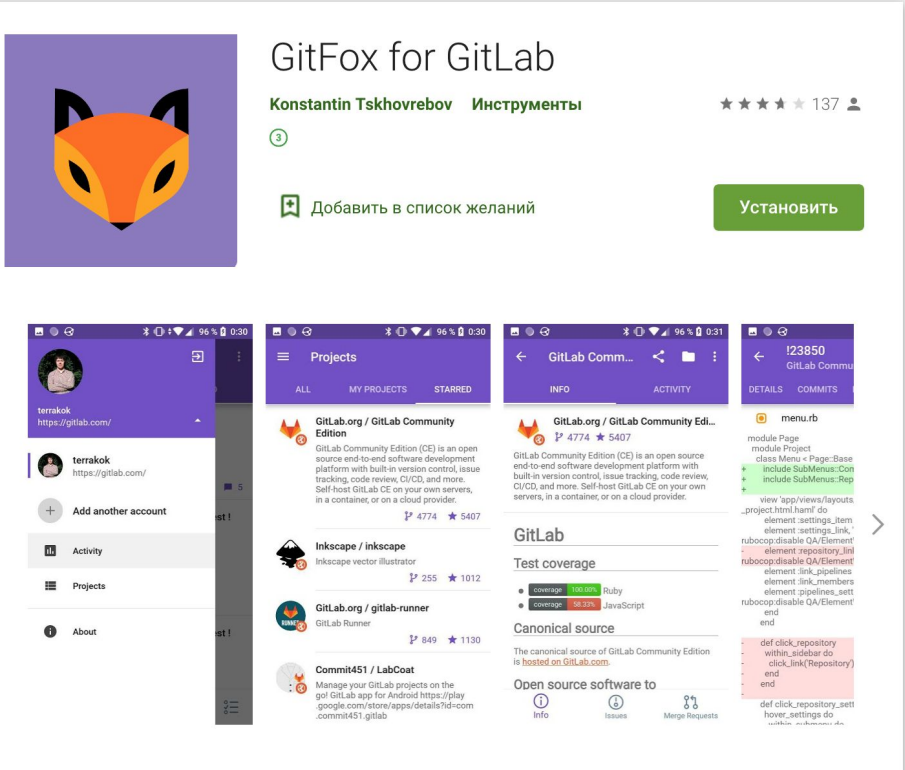
GitFox

- Не хочется выдумывать проект из воздуха
- Проверить гипотезу, что **очень типичный андроид проект можно завести на иОС**
- реально проверить все старания над архитектурой
- Собрать проблемы и опыт и поделиться им внутри JetBrains (и исправить)
- Дать обществу разработчиков недостающий элемент
- Github представил официальный клиент

https://gitlab.com/terrakok/gitlab-client

Планирование работ

Что имеем?



Clean Architecture

Multi Account

Android 10

Single Activity

MVP + Redux

Технологический стек

Язык:
Логгер:
Json:
Network:
Async and other:
DateTime:
DI framework:

Организация UI:
Image loading:
Navigation:
UI библиотеки:

Технологический стек

Язык:	Kotlin
Логгер:	
Json:	
Network:	
Async and other:	
DateTime:	
DI framework:	
Организация UI:	
Image loading:	
Navigation:	
UI библиотеки:	

Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	
Network:	
Async and other:	
DateTime:	
DI framework:	
Организация UI:	
Image loading:	
Navigation:	
UI библиотеки:	

Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	Gson
Network:	
Async and other:	
DateTime:	
DI framework:	
Организация UI:	
Image loading:	
Navigation:	
UI библиотеки:	

Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	Gson
Network:	OkHttp + Retrofit
Async and other:	
DateTime:	
DI framework:	
Организация UI:	
Image loading:	
Navigation:	
UI библиотеки:	

Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	Gson
Network:	OkHttp + Retrofit
Async and other:	RxJava
DateTime:	
DI framework:	
Организация UI:	
Image loading:	
Navigation:	
UI библиотеки:	

Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	Gson
Network:	OkHttp + Retrofit
Async and other:	RxJava
DateTime:	com.jakewharton.threetenabp
DI framework:	
Организация UI:	
Image loading:	
Navigation:	
UI библиотеки:	

Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	Gson
Network:	OkHttp + Retrofit
Async and other:	RxJava
DateTime:	com.jakewharton.threetenabp
DI framework:	Toothpick

Организация UI:	
Image loading:	
Navigation:	
UI библиотеки:	

Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	Gson
Network:	OkHttp + Retrofit
Async and other:	RxJava
DateTime:	com.jakewharton.threetenabp
DI framework:	Toothpick
Организация UI:	MVP (Moxy)
Image loading:	Glide
Navigation:	Cicerone
UI библиотеки:	androidx, material, etc

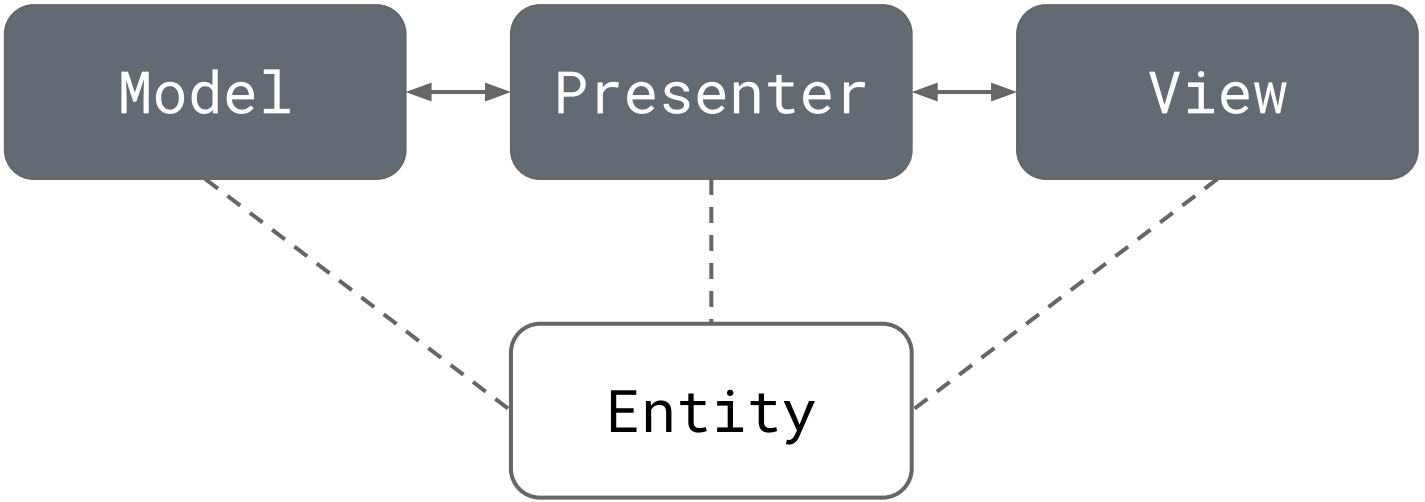
Технологический стек

Язык:	Kotlin
Логгер:	Timber
Json:	Gson
Network:	OkHttp + Retrofit
Async and other:	RxJava
DateTime:	com.jakewharton.threetenabp
DI framework:	Toothpick
Организация UI:	MVP (Moxy)
Image loading:	Glide
Navigation:	Cicerone
UI библиотеки:	androidx, material, etc

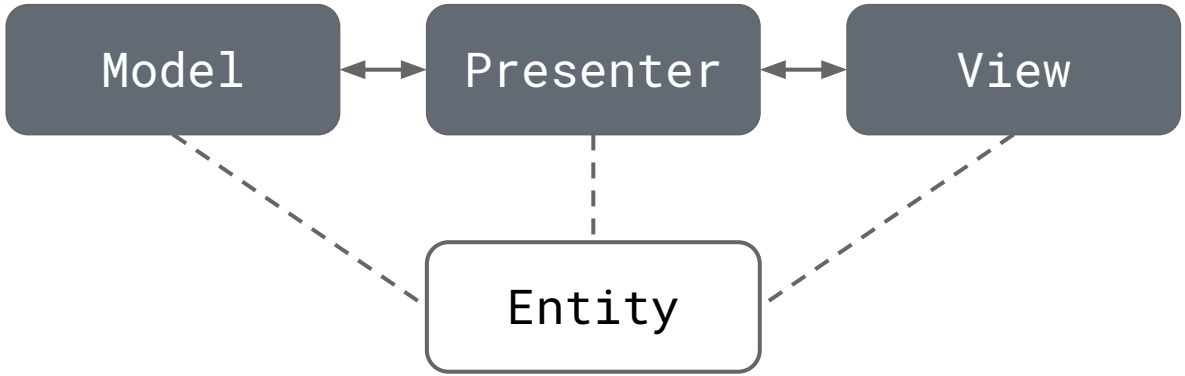
2caad75c

Архитектура (MVP)

Архитектура (MVP)

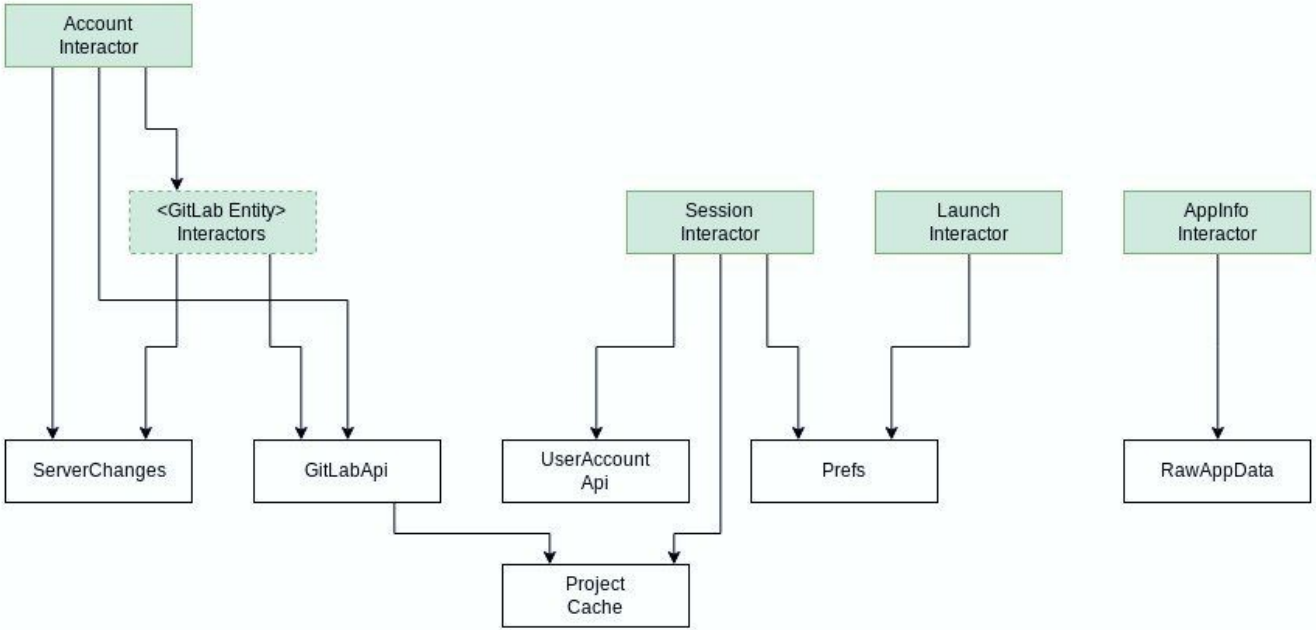


Архитектура (MVP)

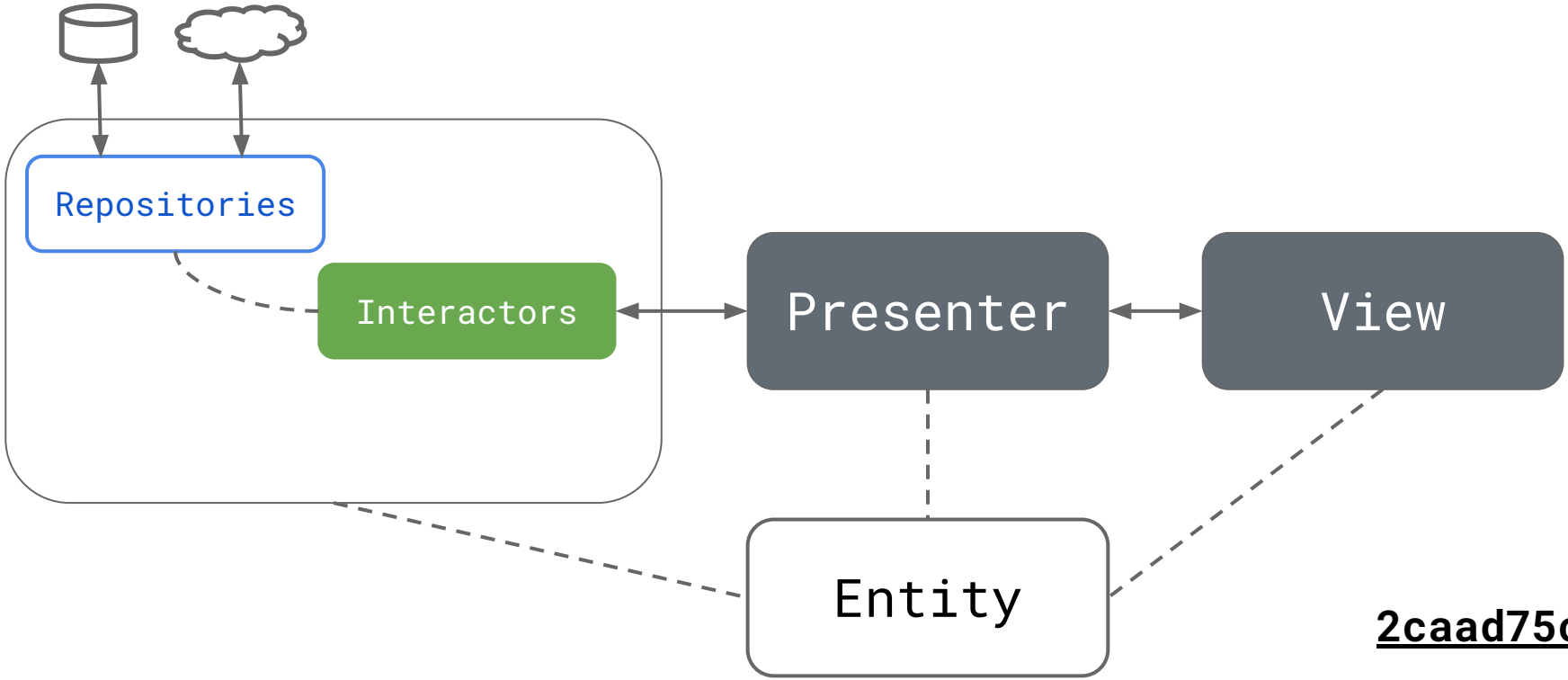


- entity
- model
- presentation
- ui

Архитектура (Model)

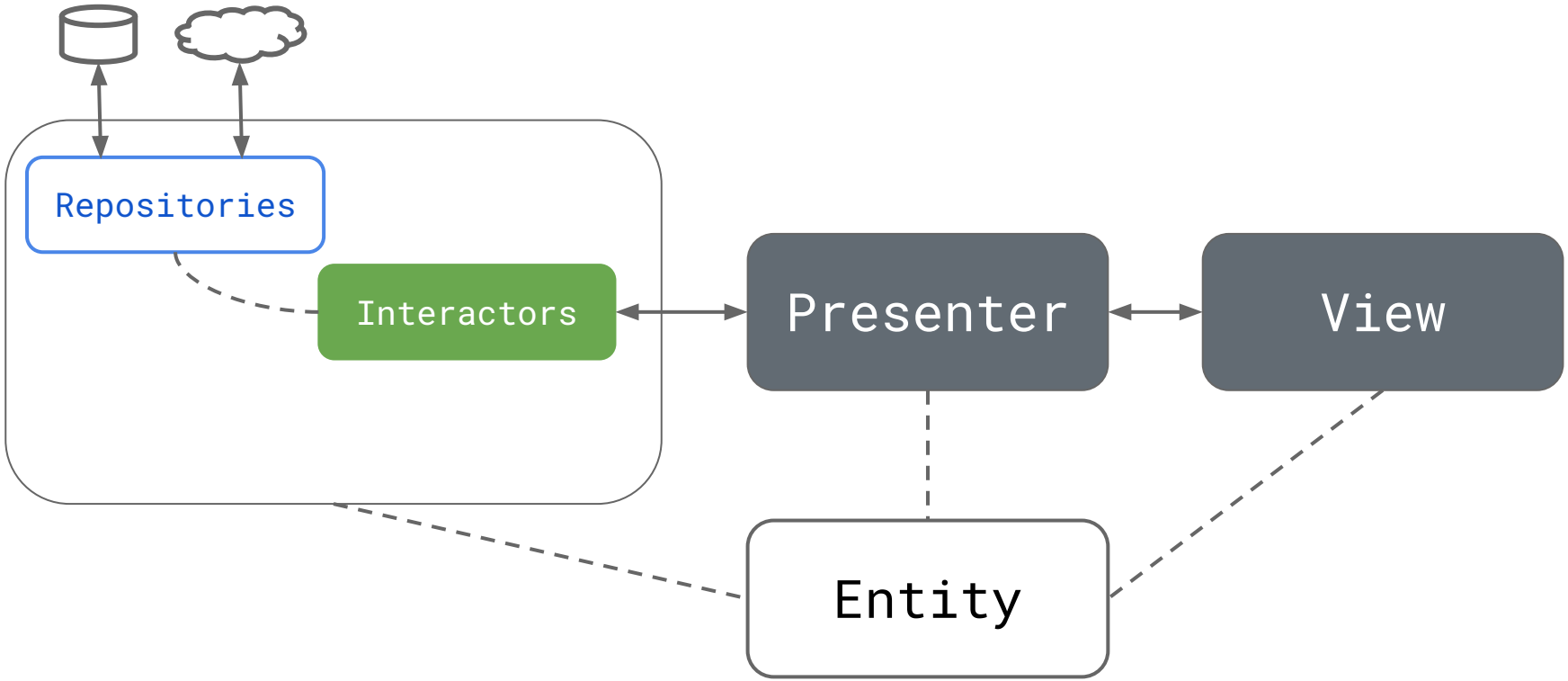


Архитектура

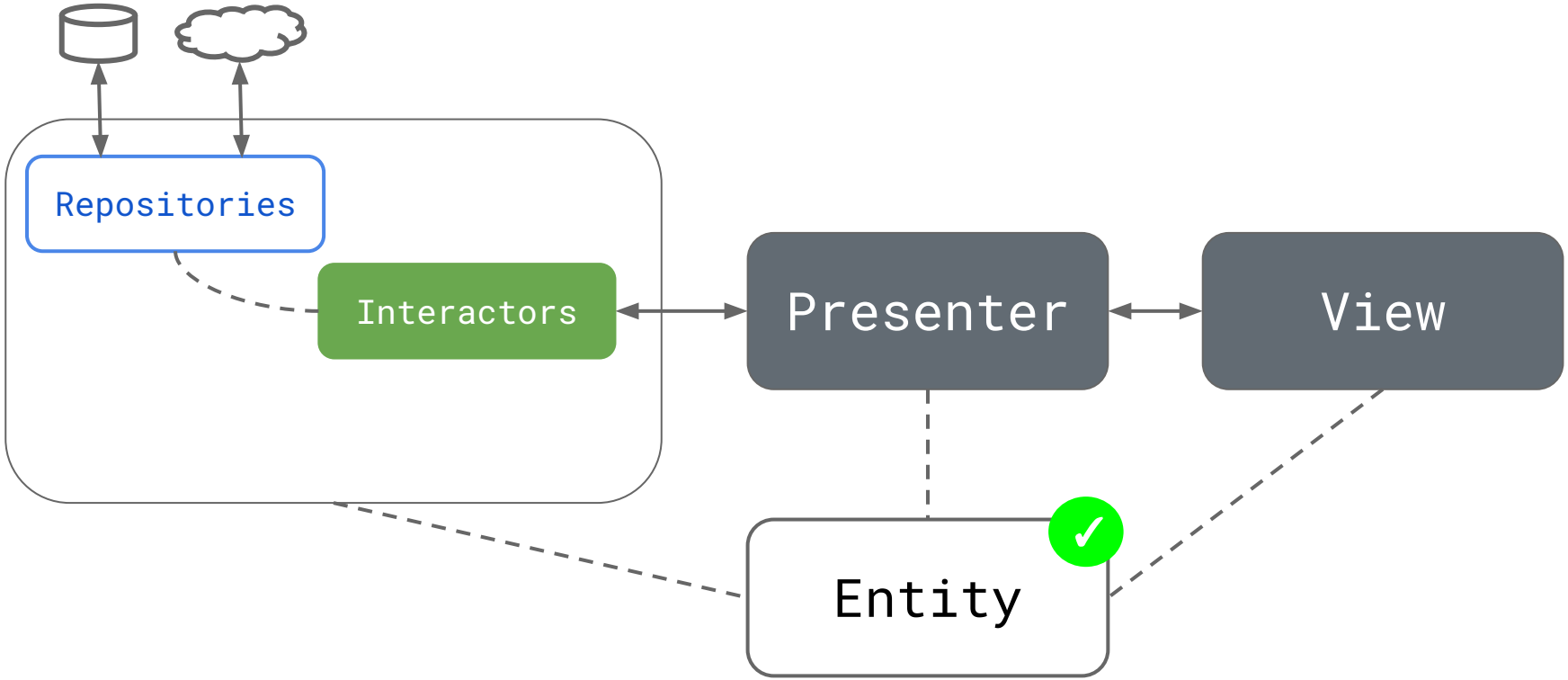


2caad75c

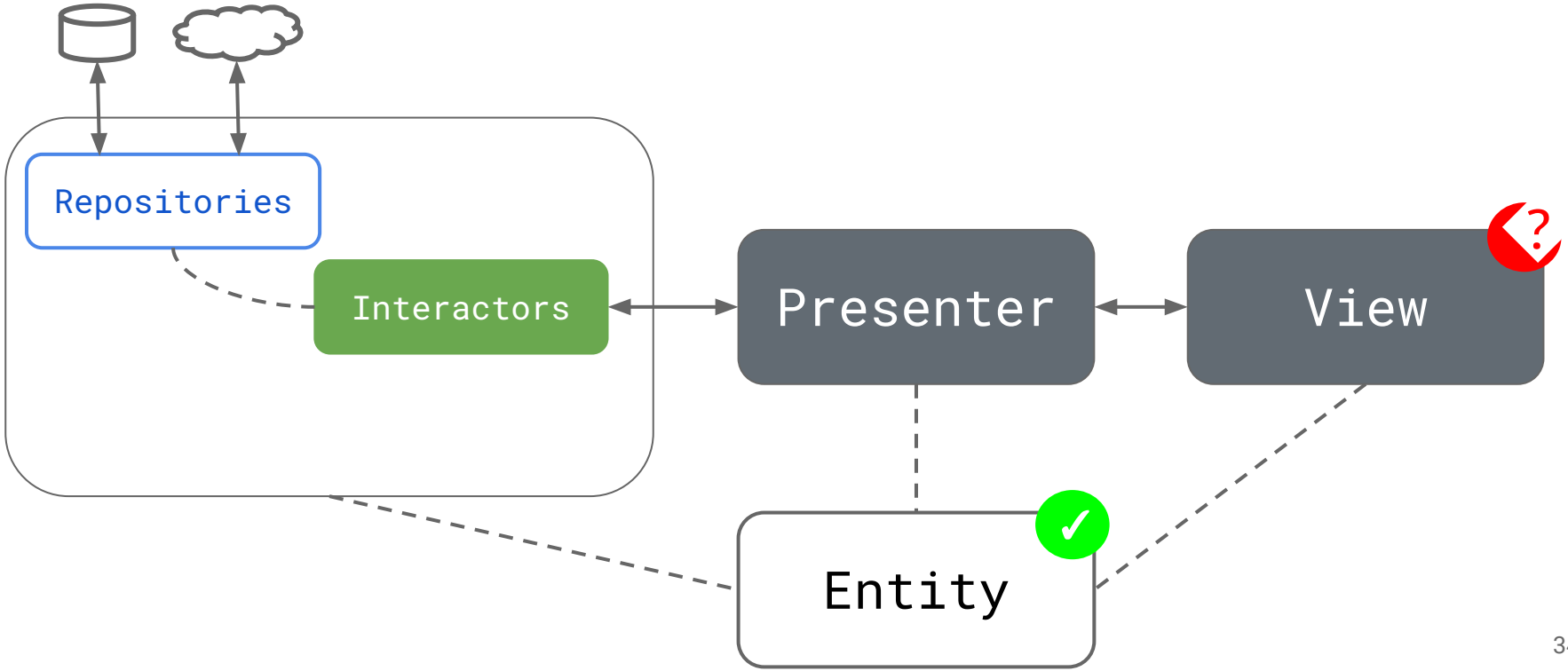
Что перенести на KMP?



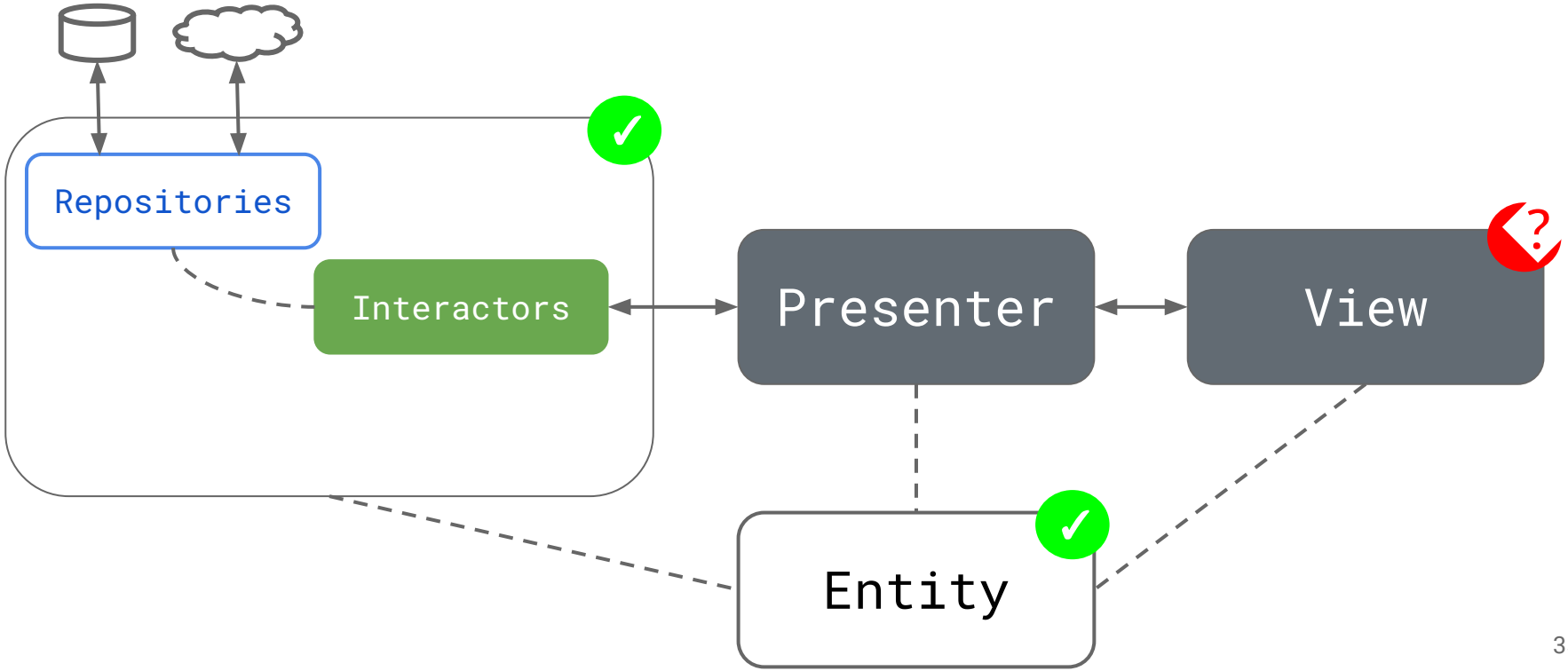
Что перенести на KMP?



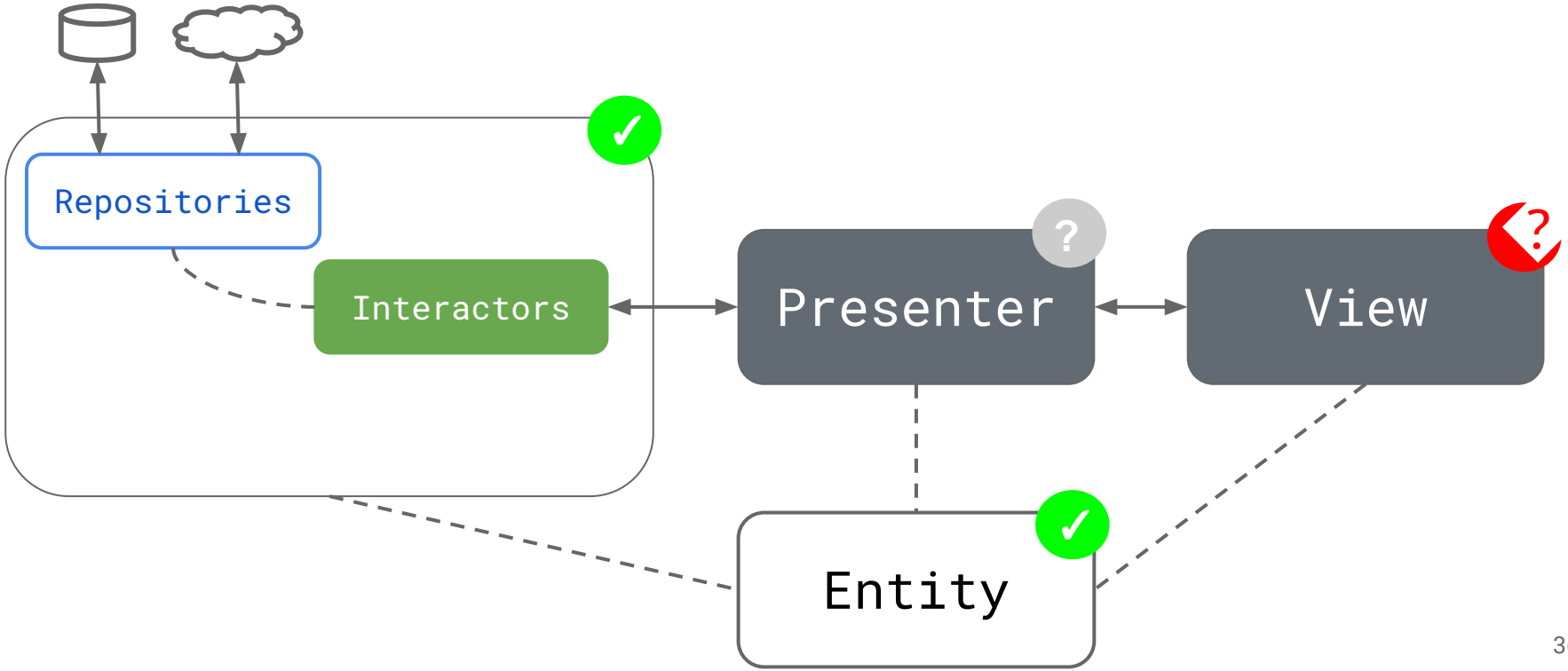
Что перенести на KMP?



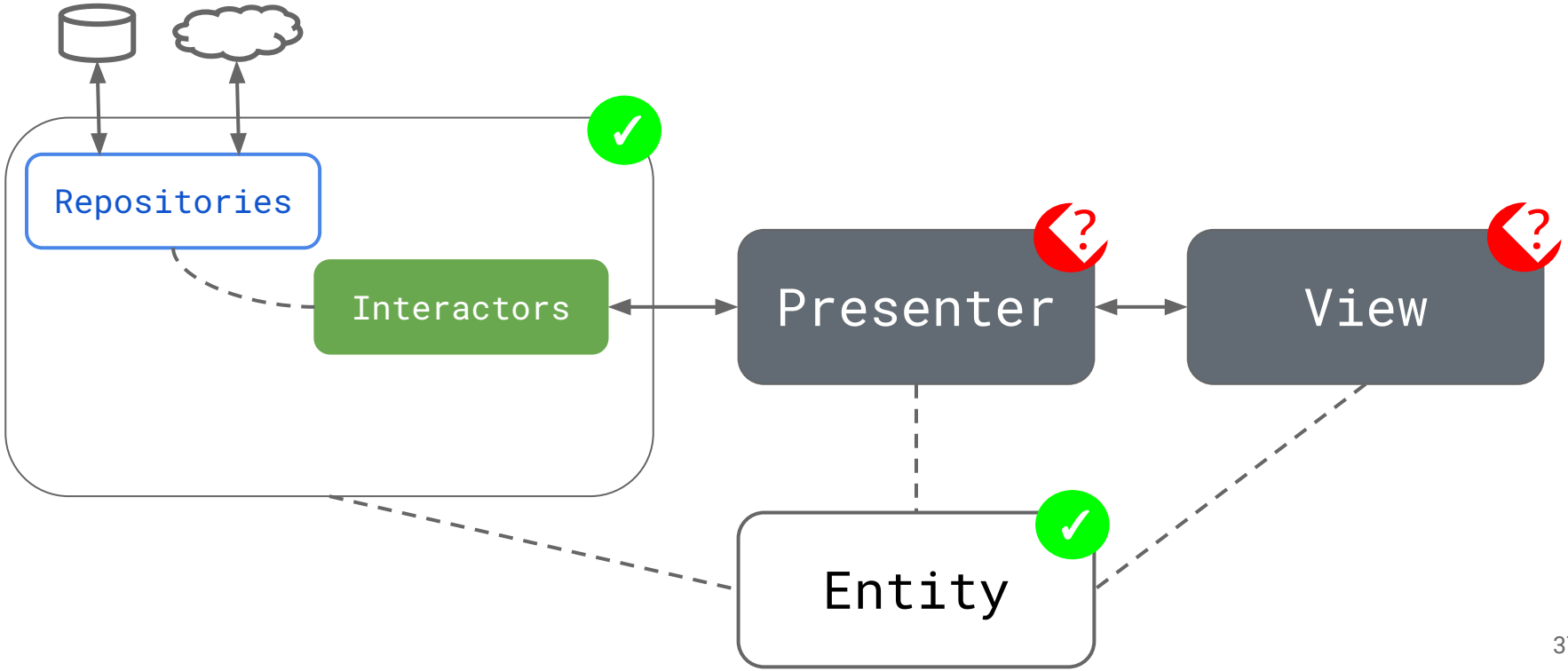
Что перенести на KMP?



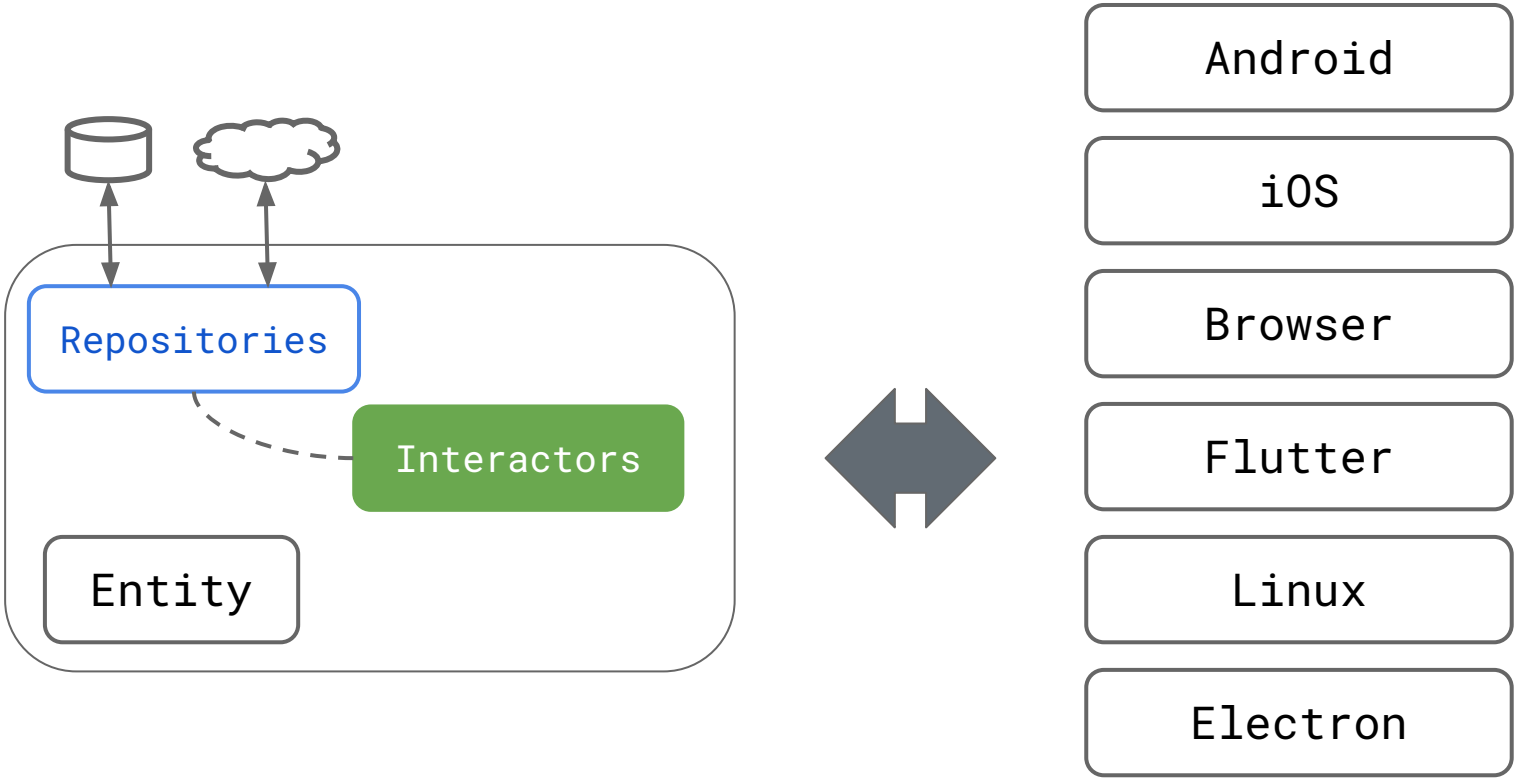
Что перенести на KMP?



Что перенести на KMP?



Multiplatform SDK



План

- Миграция библиотек на мультиплатформенные аналоги
- Замена Android специфичных частей в модели (SharedPreferences)

- Вынос SDK в отдельный Gradle модуль
- Реализация платформенных вызовов через expect-actual декларации

- Добавление новых таргетов :)

Важно!

1. Не буду много говорить о теории, больше практики и реальных проблем
2. Все можно отследить в истории Git
3. Все впечатления будут от лица андроид разработчика, который мигрирует проект на KMP

Поехали!

Поиск KMP библиотек

Поиск KMP библиотек

https://github.com/AAkira/Kotlin-Multiplatform-Libraries

Kotlin Multiplatform Libraries

Libraries

Network

Bluetooth

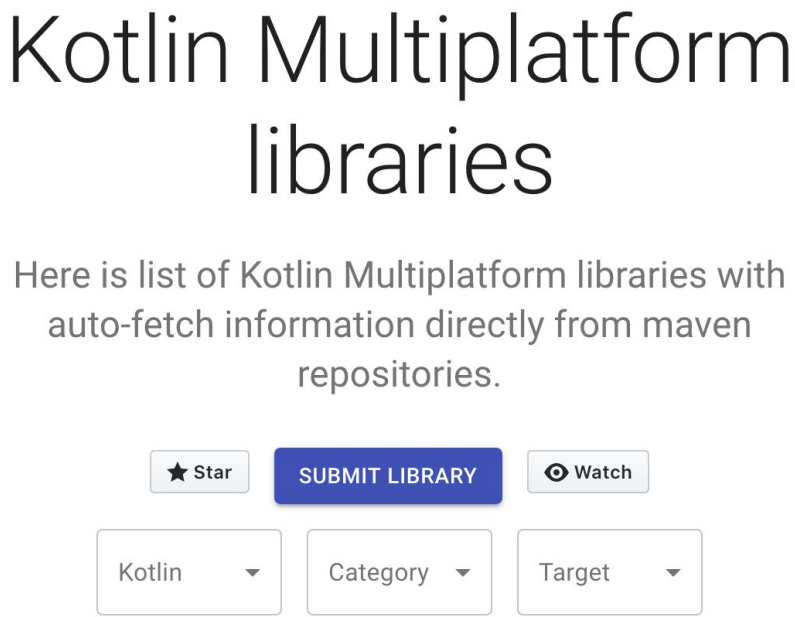
- Blue-Falcon - A unified Bluetooth library for Android and iOS.
  platform android platform native

Http

- Ktor - Most popular Http client in kotlin multiplatform.
  platform android platform native platform js platform jvm
- Ktor Client - OAuth Feature - Ktor Client Feature for handling OAuth token refreshes
  platform native platform jvm

Поиск KMP библиотек

https://libs.kmp.icerock.dev/



Миграция библиотек

Миграция библиотек

```
Gson          ->
RxJava        ->
Timber        ->
Retrofit      ->
OkHttp        ->
DateTime      ->
SharedPref    ->
Toothpick     ->
```

Миграция библиотек

```
Gson          -> Kotlinx-Serialization  hash: 3c6071a9
RxJava        ->
Timber        ->
Retrofit      ->
OkHttp        ->
DateTime      ->
SharedPref    ->
Toothpick     ->
```

Миграция библиотек

```
Gson        -> Kotlinx-Serialization  hash: 3c6071a9
RxJava      -> Kotlinx-Coroutines     hash: ab32da84
Timber      ->
Retrofit    ->
OkHttp      ->
DateTime    ->
SharedPref  ->
Toothpick   ->
```

Миграция библиотек

```
Gson          -> Kotlinx-Serialization  hash: 3c6071a9
RxJava        -> Kotlinx-Coroutines     hash: ab32da84
Timber        -> Napier                 hash: 6ccbf764
Retrofit      ->
OkHttp        ->
DateTime      ->
SharedPref    ->
Toothpick     ->
```

Миграция библиотек

```
Gson        -> Kotlinx-Serialization  hash: 3c6071a9
RxJava      -> Kotlinx-Coroutines     hash: ab32da84
Timber      -> Napier                 hash: 6ccbf764
Retrofit    -> Ktor + написал билдеры hash: 5b1345c4
OkHttp      ->
DateTime    ->
SharedPref  ->
Toothpick   ->
```

Миграция библиотек

```
Gson        -> Kotlinx-Serialization  hash: 3c6071a9
RxJava      -> Kotlinx-Coroutines     hash: ab32da84
Timber      -> Napier                 hash: 6ccbf764
Retrofit    -> Ktor + написал билдеры hash: 5b1345c4
OkHttp      -> Ktor + OkHttp engine   hash: 110fe585
DateTime    ->
SharedPref  ->
Toothpick   ->
```

Миграция библиотек

```
Gson        -> Kotlinx-Serialization  hash: 3c6071a9
RxJava      -> Kotlinx-Coroutines     hash: ab32da84
Timber      -> Napier                 hash: 6ccbf764
Retrofit    -> Ktor + написал билдеры hash: 5b1345c4
OkHttp      -> Ktor + OkHttp engine   hash: 110fe585
DateTime    -> ISO String             hash: 1be748db
SharedPref  ->
Toothpick   ->
```

Миграция библиотек

```
Gson        -> Kotlinx-Serialization  hash: 3c6071a9
RxJava      -> Kotlinx-Coroutines     hash: ab32da84
Timber      -> Napier                 hash: 6ccbf764
Retrofit    -> Ktor + написал билдеры hash: 5b1345c4
OkHttp      -> Ktor + OkHttp engine   hash: 110fe585
DateTime    -> ISO String             hash: 1be748db
SharedPref  -> Multiplatform-Settings hash: a11e894d
Toothpick   ->
```

Миграция библиотек

```
Gson        -> Kotlinx-Serialization  hash: 3c6071a9
RxJava      -> Kotlinx-Coroutines     hash: ab32da84
Timber      -> Napier                 hash: 6ccbf764
Retrofit    -> Ktor + написал билдеры hash: 5b1345c4
OkHttp      -> Ktor + OkHttp engine   hash: 110fe585
DateTime    -> ISO String             hash: 1be748db
SharedPref  -> Multiplatform-Settings hash: a11e894d
Toothpick   -> Написал фабрику        hash: 451ed5d3
```

андроид приложение
пока работает

SDK Gradle модуль

KMP module: build.gradle.kts

```
plugins { this: PluginDependenciesSpecScope
    kotlin( module: "multiplatform")
    kotlin( module: "plugin.serialization")
    id( id: "com.android.library")
}

kotlin { this: KotlinMultiplatformExtension
    android()...

    sourceSets { this: NamedDomainObjectContainer<KotlinSourceSet>
        ...
        commonMain { this: KotlinSourceSet
            dependencies {...}
        }
        val androidMain by getting { this: KotlinSourceSet!
            dependencies {... }
        }...
    }
}
```

mpp-dsl-reference.html

Java dependencies

Замена Java импортов

```
System.currentTimeMillis()
UUID.randomUUID().toString()
import android.util.Base64
import java.net.URI
```

expect/actual

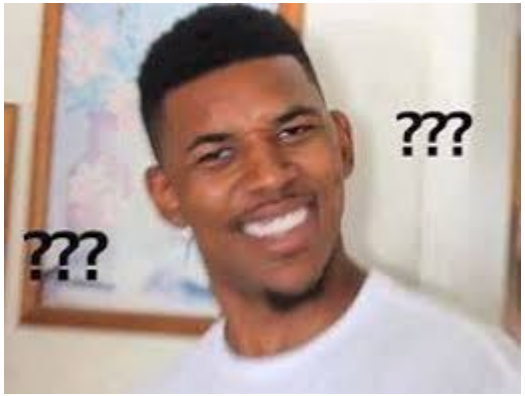
Замена Java импортов

```
System.currentTimeMillis()
UUID.randomUUID().toString()
import android.util.Base64
```

expect/actual

```
import java.net.URI          ->   import io.ktor.http.Url
```

UUID JavaScript



https://stackoverflow.com/questions/105034/how-to-create-guid-uuid

4047

For an RFC4122 version 4 compliant solution, this one-liner(ish) solution is the most compact I could come up with:

```
function uuidv4() {
  return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, function(c) {
    var r = Math.random() * 16 | 0, v = c == 'x' ? r : (r & 0x3 | 0x8);
    return v.toString(16);
  });
}

console.log(uuidv4());
```

Update, 2015-06-02: Be aware that UUID uniqueness relies heavily on the underlying random number generator (RNG). The solution above uses `Math.random()` for brevity, however `Math.random()` is *not* guaranteed to be a high-quality RNG. See Adam Hyland's excellent writeup on Math.random() for details. For a more robust solution, consider something like the uuid module (disclaimer: I, @broofa, am the author), which uses higher quality RNG APIs where available.

Update, 2015-08-26: As a side-note, this gist describes how to determine how many IDs can be generated before reaching a certain probability of collision. For example, with 3.26×10^{15} version 4 RFC4122 UUIDs you have a 1-in-a-million chance of collision.

Update, 2017-06-28: A good article from Chrome developers discussing the state of Math.random PRNG quality in Chrome, Firefox, and Safari. tl;dr - As of late-2015 it's "pretty good", but not cryptographic quality. To address that issue, here's an updated version of the above solution that uses ES6, the `crypto` API, and a bit of JS wizardry I can't take credit for:

```
function uuidv4() {
  return ([1e7]+-1e3+-4e3+-8e3+-1e11).replace(/[018]/g, c =>
    (c ^ crypto.getRandomValues(new Uint8Array(1))[0] & 15 >> c / 4).toString(
  );
}

console.log(uuidv4());
```

Update, 2020-01-06: There is a proposal in the works for a standard `uuid` module as part of the JS language

share improve this answer follow edited 2 days ago community wiki 19 revs, 10 users 51%

Замена Java импортов

```
System.currentTimeMillis()

import android.util.Base64
```

expect/actual

```
import java.net.URI                 ->   import io.ktor.http.Url
UUID.randomUUID().toString()        ->   benasher44/uuid
```

Android

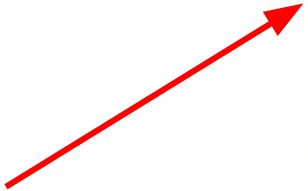
Android library: build.gradle.kts

```
plugins { this: PluginDependenciesSpecScope
    kotlin( module: "multiplatform")
    kotlin( module: "plugin.serialization")
    id( id: "com.android.library")
}

kotlin { this: KotlinMultiplatformExtension
    android()...

    sourceSets { this: NamedDomainObjectContainer<KotlinSourceSet>
        ...
        commonMain { this: KotlinSourceSet
            dependencies {...}
        }
        val androidMain by getting { this: KotlinSourceSet!
            dependencies {... }
        }...
    }
}
```

Android library: build.gradle.kts

```
plugins { this: PluginDependenciesSpecScope
    kotlin( module: "multiplatform")
    kotlin( module: "plugin.serialization")
    id( id: "com.android.library")
}

kotlin { this: KotlinMultiplatformExtension
    android()...

    sourceSets { this: NamedDomainObjectContainer<KotlinSourceSet>
        ...
        commonMain { this: KotlinSourceSet
            dependencies {...}
        }
        val androidMain by getting { this: KotlinSourceSet!
            dependencies {... }
        }...
    }
}

android { this: LibraryExtension
    compileSdkVersion( apiLevel: 29)
    defaultConfig { this: DefaultConfig
        minSdkVersion( minSdkVersion: 19)
        targetSdkVersion( targetSdkVersion: 29)
    }
    sourceSets["main"].manifest.srcFile( srcPath: "src/androidMain/AndroidManifest.xml")
}
```

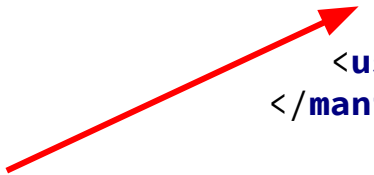
Android manifest



```
package gitfox

import ...

open class SDK internal constructor(
    private val defaultPageSize: Int = SDK.defaultPageSize,
    private val cacheLifetime: Long = SDK.cacheLifetime,
    private val oAuthParams: OAuthParams,
    private val isDebug: Boolean,
    private val httpClientFactory: HttpClientFactory,
    private val settings: Settings
) {...
}
```

Android manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="terrakok.gitfox">

    <uses-permission android:name="android.permission.INTERNET" />
</manifest>
```



Matching Fallbacks

```
buildTypes {
    create("debugPG") {
        initWith(getByName("debug"))
        isMinifyEnabled = true
        versionNameSuffix = " debugPG"
        matchingFallbacks = mutableListOf("debug")

        proguardFiles(
            getDefaultProguardFile("proguard-android-optimize.txt"),
            file("proguard-rules.pro")
        )
    }
}
```

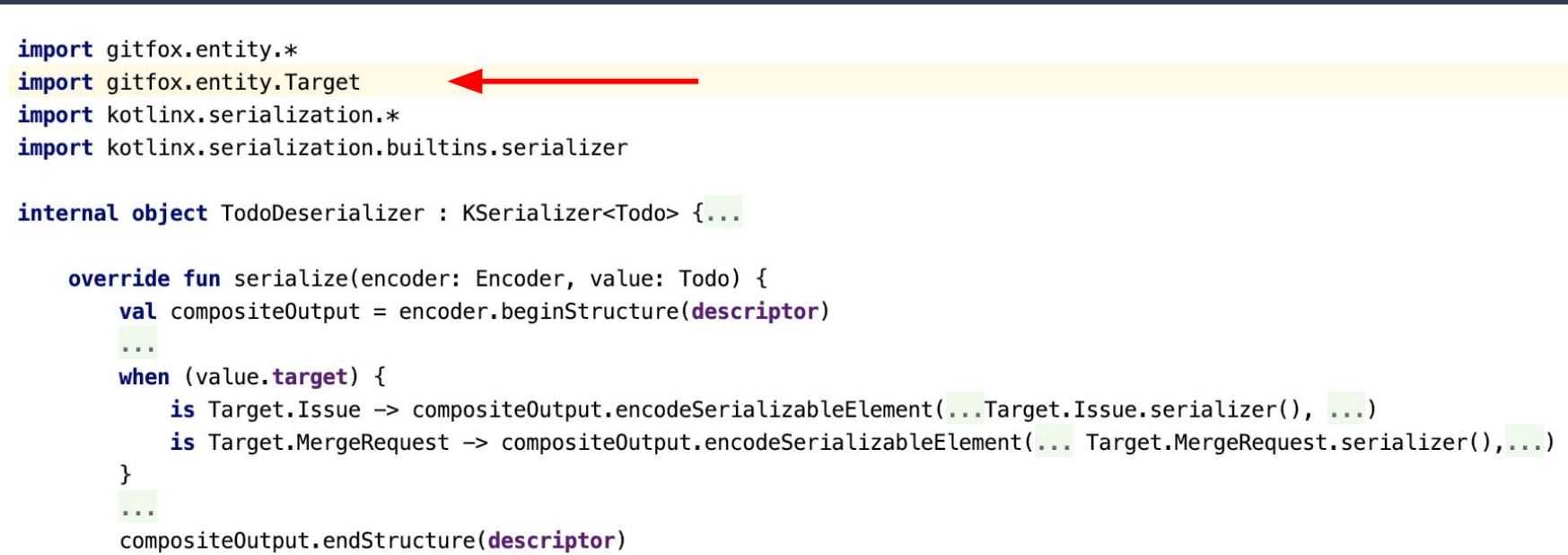
Magic Error!

```
Type mismatch: inferred type is kotlin.annotation.Target?
but gitfox.entity.Target? was expected
```

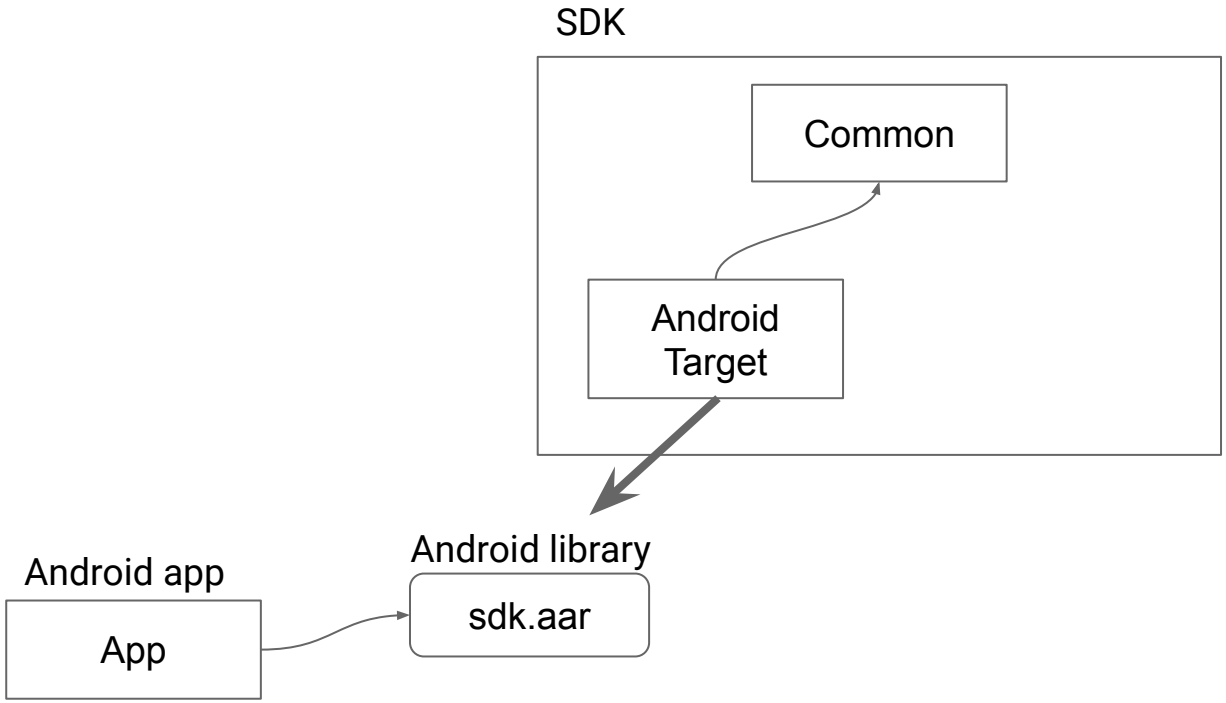
Magic Error!

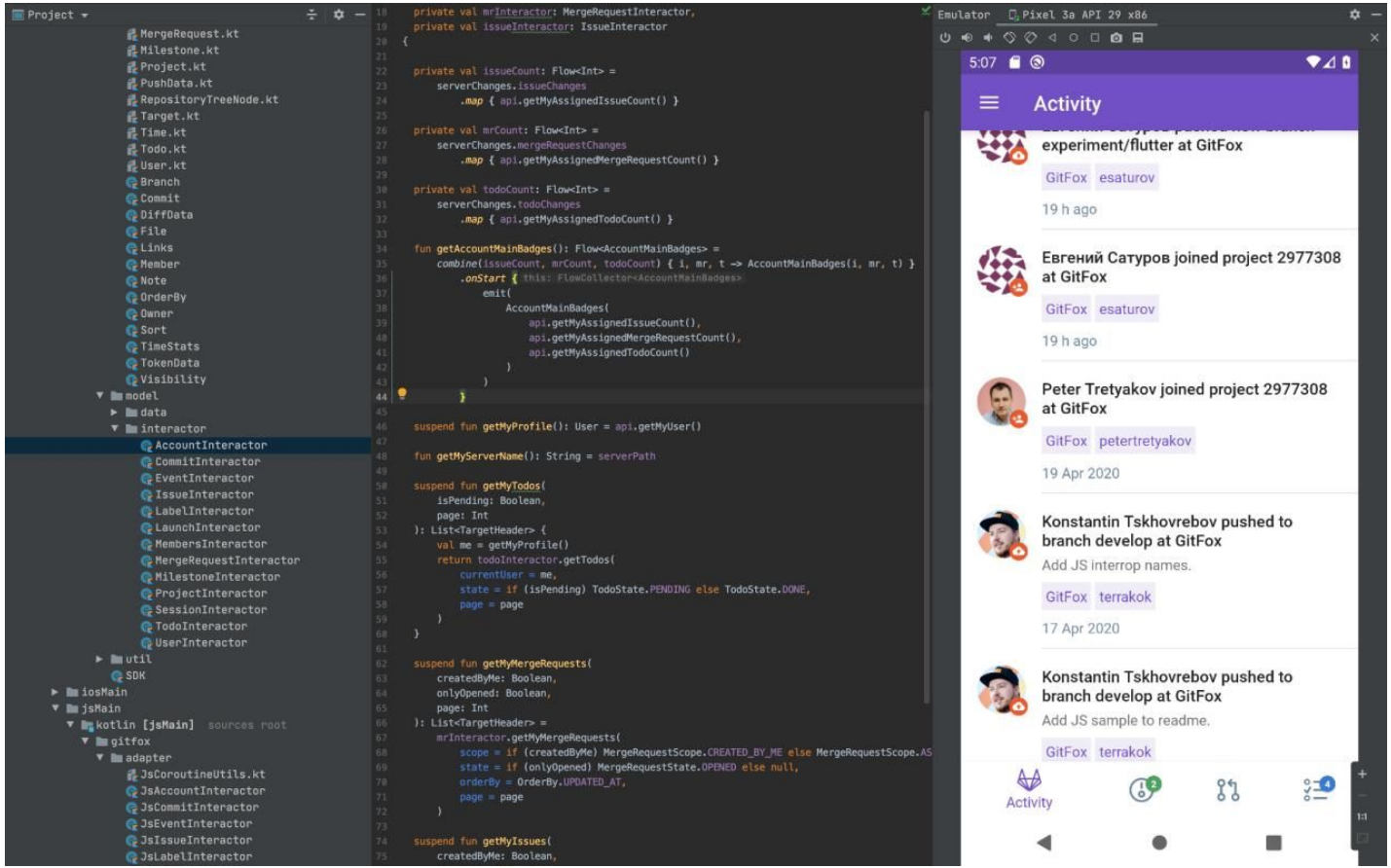
```
import gitfox.entity.*
import kotlinx.serialization.*
import kotlinx.serialization.builtins.serializer

internal object TodoDeserializer : KSerializer<Todo> {...

    override fun serialize(encoder: Encoder, value: Todo) {
        val compositeOutput = encoder.beginStructure(descriptor)
        ...
        when (value.target) {
            is Target.Issue -> compositeOutput.encodeSerializableElement(...Target.Issue.serializer(), ...)
            is Target.MergeRequest -> compositeOutput.encodeSerializableElement(... Target.MergeRequest.serializer(),...)
        }
        ...
        compositeOutput.endStructure(descriptor)
    }
```

Magic Error!



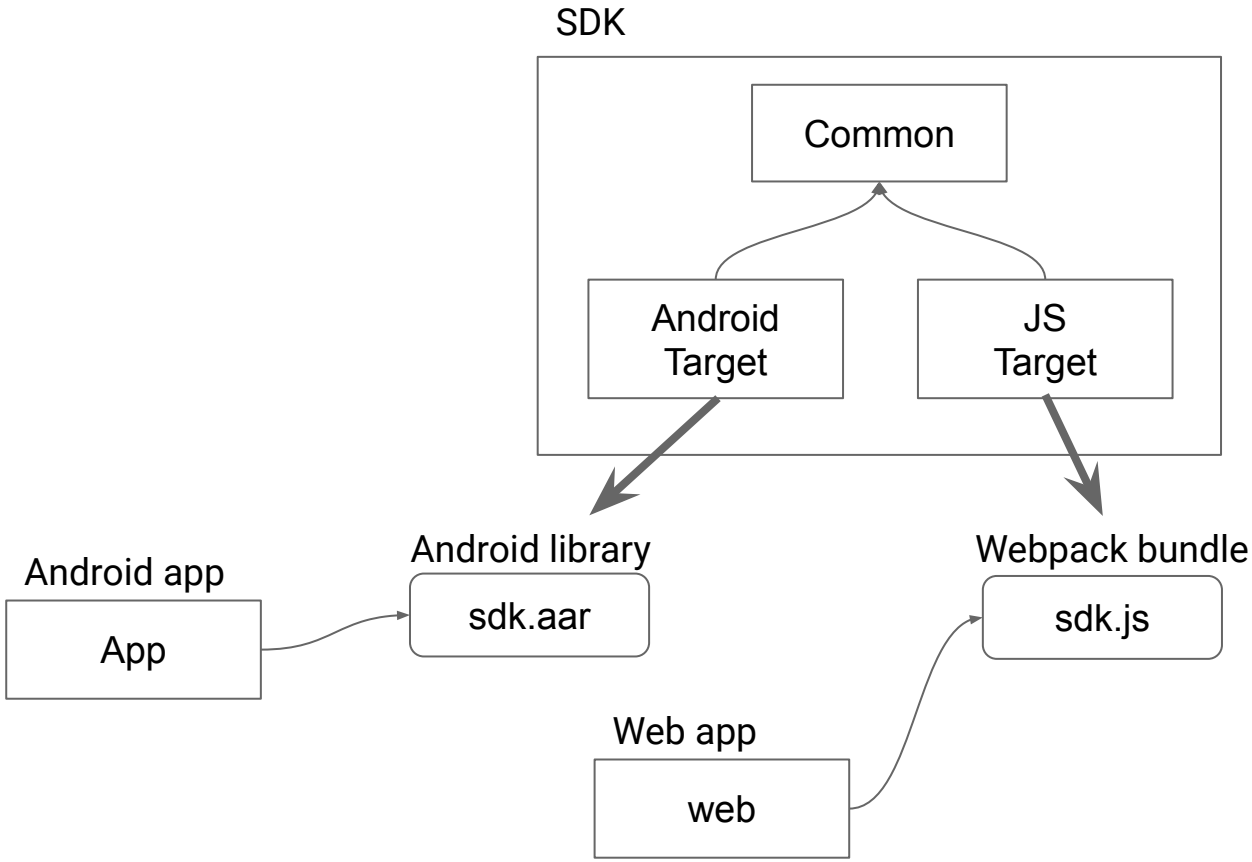
```
import gitfox.entity.*
import gitfox.entity.Target
import kotlinx.serialization.*
import kotlinx.serialization.builtins.serializer

internal object TodoDeserializer : KSerializer<Todo> {...

    override fun serialize(encoder: Encoder, value: Todo) {
        val compositeOutput = encoder.beginStructure(descriptor)
        ...
        when (value.target) {
            is Target.Issue -> compositeOutput.encodeSerializableElement(...Target.Issue.serializer(), ...)
            is Target.MergeRequest -> compositeOutput.encodeSerializableElement(... Target.MergeRequest.serializer(),...)
        }
        ...
        compositeOutput.endStructure(descriptor)
    }
```


SDK
Common
Android
Target
Android library
sdk.aar
Android app
App



Android работает!

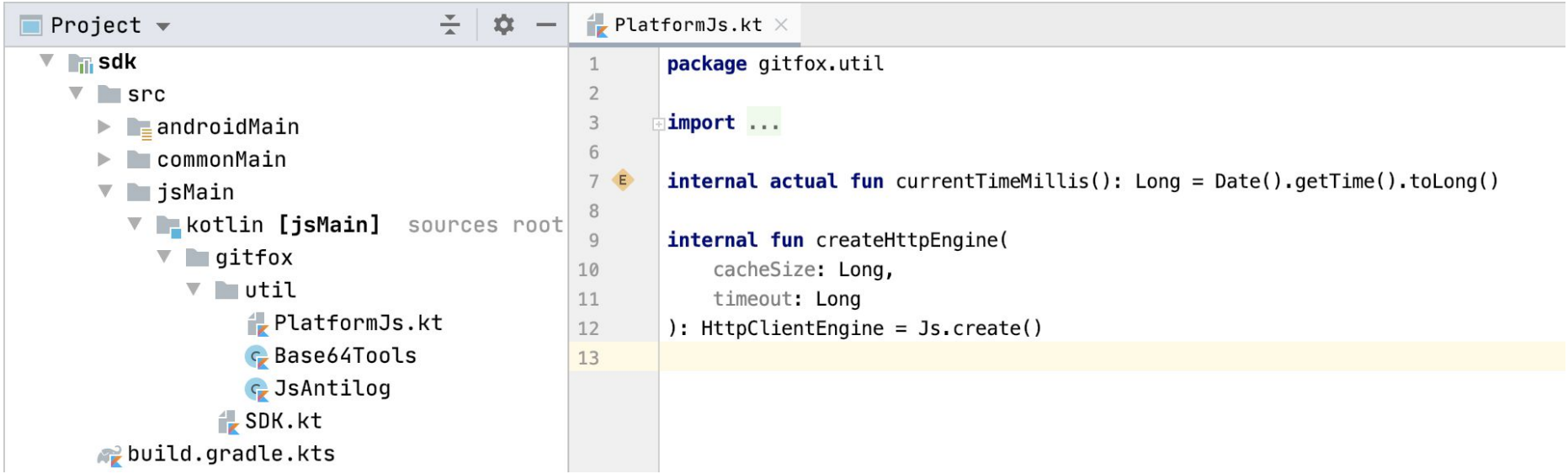
Browser (JavaScript)


SDK
Common
Android
Target
JS
Target
Android library
sdk.aar
Webpack bundle
sdk.js
Android app
App
Web app
web

KMP plugin

```
js { browser { } }

sourceSets {
    val jsMain by getting {
        dependencies {
            //Kotlin
            implementation("org.jetbrains.kotlin:kotlin-stdlib-js")
            //Log
            implementation("com.github.aakira:napier-js:$napierVersion")
            //Network
            implementation("io.ktor:ktor-client-core-js:$ktorVersion")
            implementation("io.ktor:ktor-client-js:$ktorVersion")
            implementation("io.ktor:ktor-client-serialization-js:$ktorVersion")
            implementation("io.ktor:ktor-client-auth-js:$ktorVersion")
            implementation("io.ktor:ktor-client-logging-js:$ktorVersion")
            //Coroutines
            implementation("org.jetbrains.kotlinx:kotlinx-coroutines-core-js:$coroutinesVersion")
            //JSON
           implementation("org.jetbrains.kotlinx:kotlinx-serialization-runtime-js:$serializationVersion")
        }
    }
}
```

Платформенная часть



./gradlew jsBrowserWebpack

```
Unresolved reference: java
```

```
1  package gitfox.entity.app.target
2
3  import java.io.Serializable
4
5  /**
6   * @author Maxim Myalkin (MaxMyalkin) on 19.05.2019.
7   */
8  sealed class TargetAction : Serializable {
9
10     data class CommentedOn(val noteId: Long) : TargetAction()
11
```

```
1  package gitfox.entity.app.target
2
3  /**
4   * @author Maxim Myalkin (MaxMyalkin) on 19.05.2019.
5   */
6  sealed class TargetAction {
7
8      data class CommentedOn(val noteId: Long) : TargetAction()
9
```

fda0b108

Serializable интерфейс

Один таргет в KMP проекте - случай уникальный

./gradlew jsBrowserWebpack

```
ERROR in /Users/.../js/packages_imported/ktor-ktor-io/1.3.2/ktor-ktor-io.js
  Module not found: Error: Can't resolve 'text-encoding' in
'/Users/.../js/packages_imported/ktor-ktor-io/1.3.2'
   @ /Users/konstantin.tskhovrebov/Documents/git/gitlab-client/...
   @ ./kotlin/gitlab-client-sdk.js
   @ multi ./kotlin/gitlab-client-sdk.js
```

Добавляем потерянные npm зависимости

```
//required NPM dependencies
implementation(npm("text-encoding", "*"))
implementation(npm("abort-controller", "*"))
```

получили заветную
sdk.js

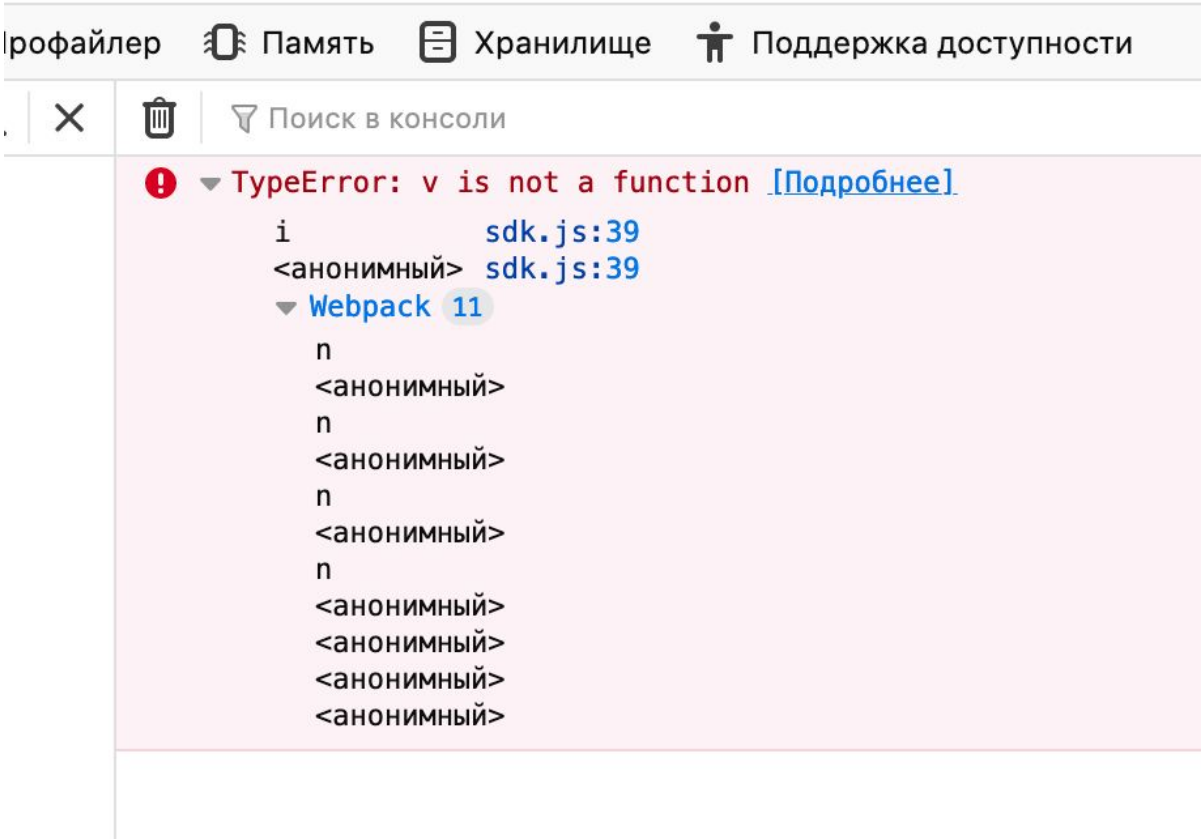
Для тестов добавил `fun main()`

```
+ package gitfox
+
+ import gitfox.entity.app.develop.AppInfo
+ import gitfox.entity.app.session.OAuthParams
+ import kotlinx.coroutines.GlobalScope
+ import kotlinx.coroutines.launch
+
+ fun main() {
+     println("Hello terrakok!")
+     val sdk = SDK(
+         "https://gitlab.com/",
+         oAuthParams = OAuthParams("", "", "", ""),
+         appInfo = AppInfo("jsVersion", 42, "js test", "local", "", ""),
+         getLibraries = { emptyList() },
+         isDebug = true
+     )
+     println(sdk.getAppInfoInteractor().getAppInfo().versionName)
+
+     GlobalScope.launch {
+         sdk.getSessionInteractor().loginOnCustomServer(
+             "https://gitlab.com/",
+             "test-token" //todo add real token
+         )
+         val events = sdk.getEventInteractor().getEvents(page = 0)
+         println("!!!!!!!!!!!!!EVENTS!!!!!!!!!!!!!")
+         println(events)
+     }
+ }
```

И простую **index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Console Output</title>
</head>
<body>
<script type="text/javascript" src="./sdk/build/distributions/sdk.js"></script>
</body>
</html>
```

открываем браузер...


рофайлер
Память
Хранилище
Поддержка доступности
Поиск в консоли
TypeError: v is not a function [Подробнее]
i sdk.js:39
<анонимный> sdk.js:39
Webpack 11
n
<анонимный>
n
<анонимный>
n
<анонимный>
n
<анонимный>
<анонимный>
<анонимный>
<анонимный>

DCE tool

```
js {
  browser {
    dceTask {
       keep("ktor-ktor-io.\$\$importsForInline\$\$.ktor-ktor-io.io.ktor.utils.io")
    }
  }
}
```

https://youtrack.jetbrains.com/issue/KT-37894

открываем браузер...

```
TypeError: this.userId is undefined
```

```kotlin
@Serializable
data class UserAccount(
    @SerialName("userId") val userId: Long,
    @SerialName("token") val token: String,
    @SerialName("serverPath") val serverPath: String,
    @SerialName("avatarUrl") val avatarUrl: String,
    @SerialName("userName") val userName: String,
    @SerialName("isOAuth") val isOAuth: Boolean
) {
    val id: String = "$userId : $serverPath"
}
```

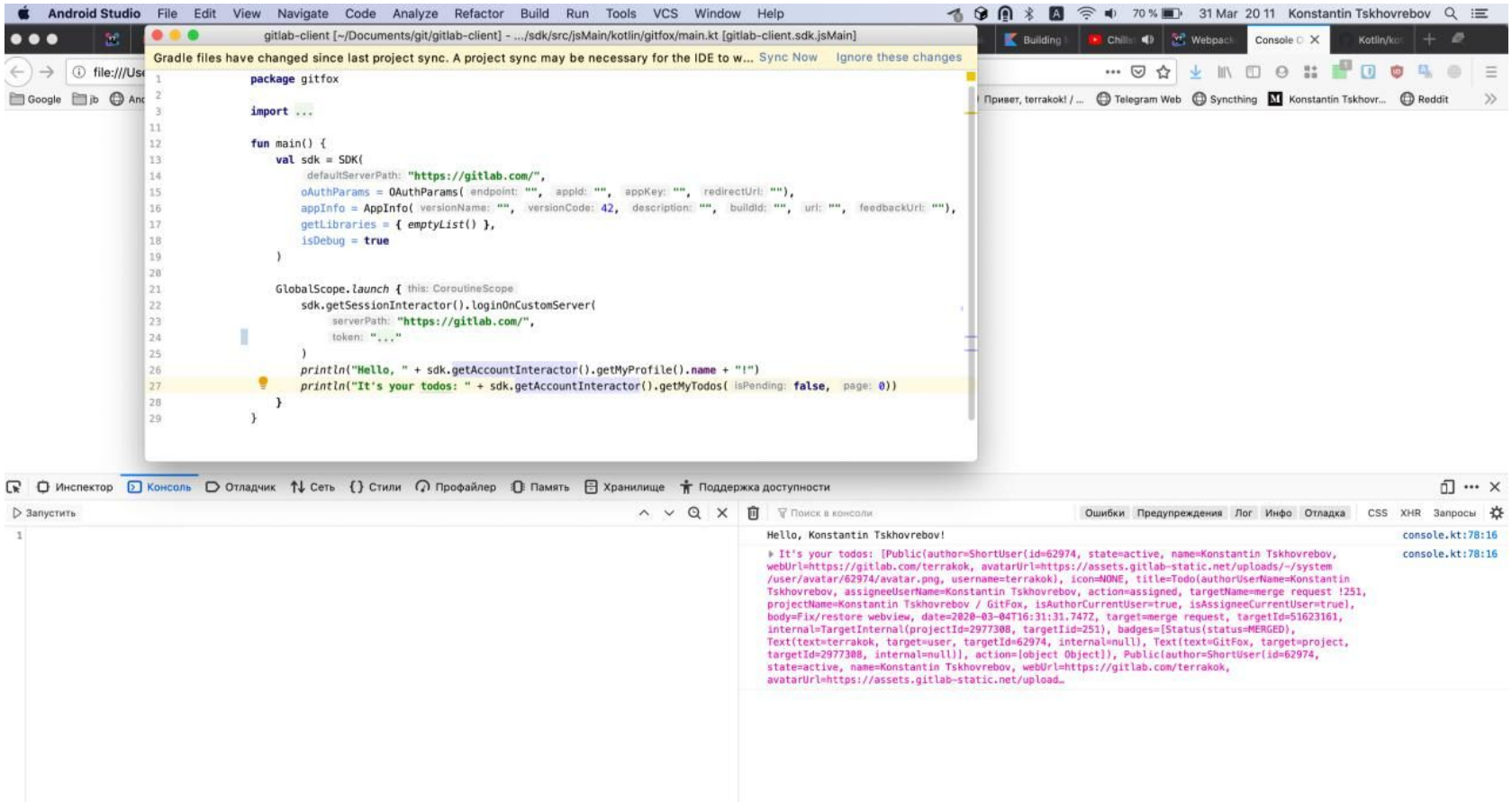
TypeError: this.userId is undefined

3f435691

```kotlin
@Serializable
data class UserAccount(
    @SerialName("userId") val userId: Long,
    @SerialName("token") val token: String,
    @SerialName("serverPath") val serverPath: String,
    @SerialName("avatarUrl") val avatarUrl: String,
    @SerialName("userName") val userName: String,
    @SerialName("isOAuth") val isOAuth: Boolean
) {
    val id: String get() = "$userId : $serverPath"
}
```

3f435691

открываем браузер...


Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
gitlab-client [~/Documents/git/gitlab-client] - .../sdk/src/jsMain/kotlin/gitfox/main.kt [gitlab-client.sdk.jsMain]
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to w... Sync Now Ignore these changes
package gitfox
import ...
fun main() {
val sdk = SDK(
defaultServerPath: "https://gitlab.com/",
oAuthParams = OAuthParams(endpoint: "", appId: "", appKey: "", redirectUrl: ""),
appInfo = AppInfo(versionName: "", versionCode: 42, description: "", buildId: "", url: "", feedbackUrl: ""),
getLibraries = { emptyList() },
isDebug = true
)
GlobalScope.launch { this: CoroutineScope
sdk.getSessionInteractor().loginOnCustomServer(
serverPath: "https://gitlab.com/",
token: "..."
)
println("Hello, " + sdk.getAccountInteractor().getMyProfile().name + "!")
println("It's your todos: " + sdk.getAccountInteractor().getMyTodos(isPending: false, page: 0))
}
}
Инспектор Консоль Отладчик Сеть Стили Профайлер Память Хранилище Поддержка доступности
Запустить
Поиск в консоли
Ошибки Предупреждения Лог Инфо Отладка CSS XHR Запросы
Hello, Konstantin Tskhovrebov!
console.kt:78:16
It's your todos: [Public(author=ShortUser(id=62974, state=active, name=Konstantin Tskhovrebov, webUrl=https://gitlab.com/terrakok, avatarUrl=https://assets.gitlab-static.net/uploads/-/system/user/avatar/62974/avatar.png, username=terrakok), icon=NONE, title=Todo(authorUserName=Konstantin Tskhovrebov, assigneeUserName=Konstantin Tskhovrebov, action=assigned, targetName=merge request !251, projectName=Konstantin Tskhovrebov / GitFox, isAuthorCurrentUser=true, isAssigneeCurrentUser=true), body=Fix/restore webview, date=2020-03-04T16:31:31.747Z, target=merge request, targetId=51623161, internal=TargetInternal(projectId=2977308, targetIid=251), badges=[Status(status=MERGED), Text(text=terrakok, target=user, targetId=62974, internal=null), Text(text=GitFox, target=project, targetId=2977308, internal=null)], action=[object Object]), Public(author=ShortUser(id=62974, state=active, name=Konstantin Tskhovrebov, webUrl=https://gitlab.com/terrakok, avatarUrl=https://assets.gitlab-static.net/upload…
console.kt:78:16

Работает!

Pure JavaScript SDK

JavaScript не знает про suspend

```
class CommitInteractor internal constructor(
    private val api: GitlabApi
) {

    suspend fun getCommit(projectId: Long, commitId: String): Commit =
        api.getRepositoryCommit(projectId, commitId)

    suspend fun getCommitDiffData(projectId: Long, commitId: String): List<DiffData> =
        api.getCommitDiffData(projectId, commitId)
}
```

d6fd9425

JavaScript Promise

```
class JsCommitInteractor internal constructor(
    private val interactor: CommitInteractor
) : CoroutineScope by CoroutineScope(Dispatchers.Main) {

    fun getCommit(
        projectId: Long,
        commitId: String
    ) = promise { interactor.getCommit(projectId, commitId) }

    fun getCommitDiffData(
        projectId: Long,
        commitId: String
    ) = promise { interactor.getCommitDiffData(projectId, commitId) }
}
```

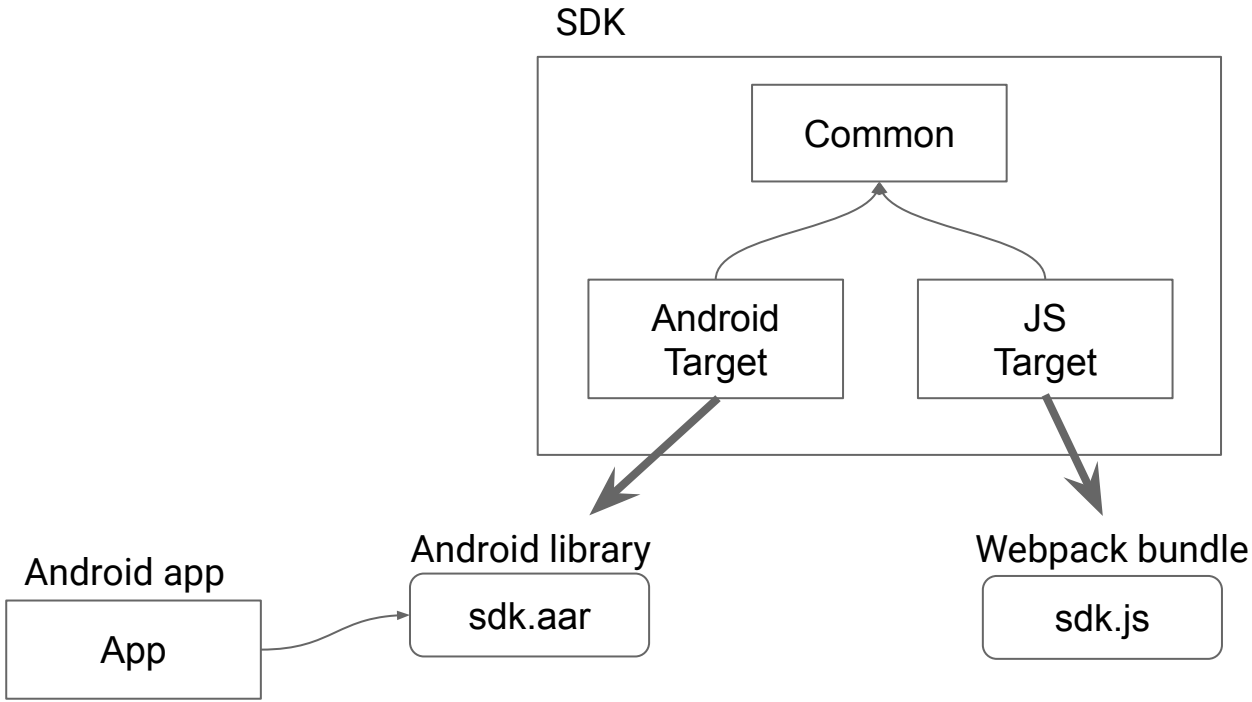
d6fd9425

```
const commitInteractor = new JsCommitInteractor();

commitInteractor.getCommit_puj7f4$(...).then( () => { … } );
```

JavaScript names

```
class JsCommitInteractor internal constructor(
    private val interactor: CommitInteractor
) : CoroutineScope by CoroutineScope(Dispatchers.Main) {

    @JsName("getCommit")
    fun getCommit(
        projectId: Long,
        commitId: String
    ) = promise {
        interactor.getCommit(projectId, commitId)
    }

    @JsName("getCommitDiffData")
    fun getCommitDiffData(
        projectId: Long,
        commitId: String
    ) = promise {
        interactor.getCommitDiffData(projectId, commitId)
    }
}
```

4ae587df


SDK
Common
Android
Target
JS
Target
Android library
sdk.aar
Webpack bundle
sdk.js
Android app
App

Проблема сборки на другой машине

```
git clone https://gitlab.com/terrakok/gitlab-client.git
cd ./gitlab-client/
./gradlew :sdk:jsBrowserDevelopmentWebpack
```

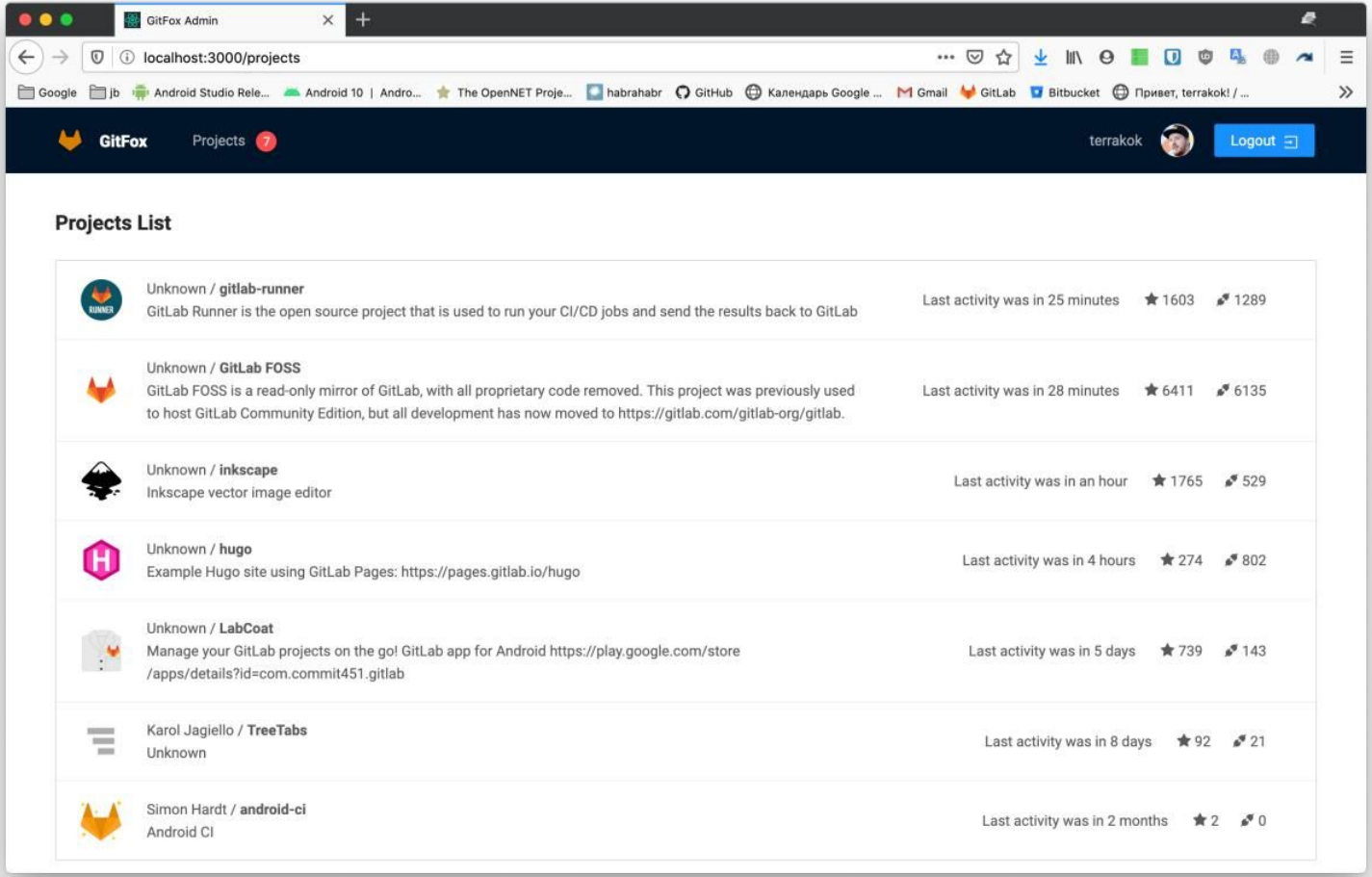
```
FAILED
Could not determine the dependencies of task ':sdk:jsPackageJson'.
> Cannot add a configuration with name 'jsNpm' as a configuration with
that name already exists
```

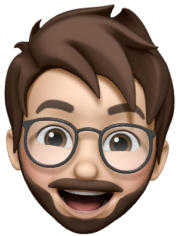
Проблема сборки на другой машине

local.properties

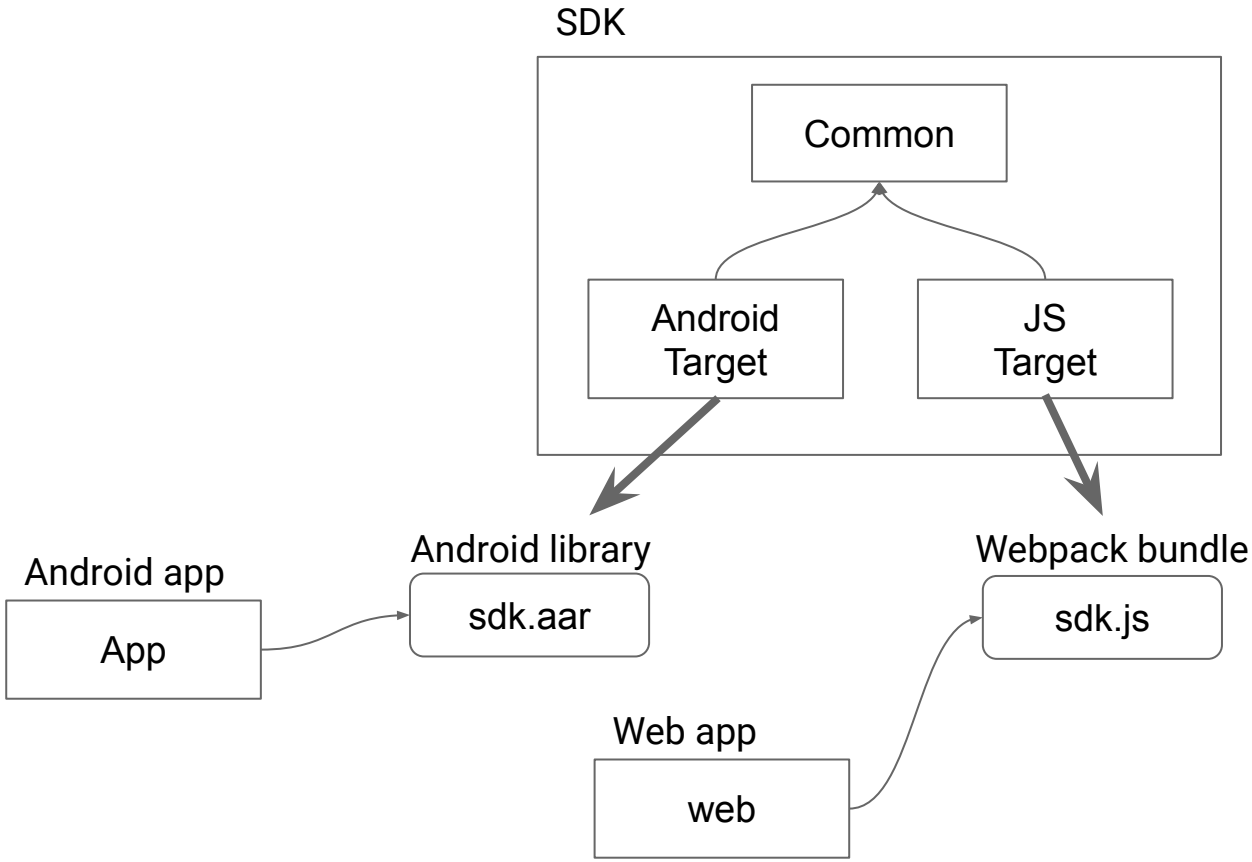
```
## This file must *NOT* be checked into Version Control Systems,
# as it contains information specific to your local configuration.
#
# Location of the SDK. This is only used by Gradle.
# For customization when using a Version Control System, please read the
# header note.
#Tue Mar 03 18:32:10 MSK 2020
sdk.dir=/Users/konstantin.tskhovrebov/Library/Android/sdk
```

¯_(ツ)_/¯





Dmitriy Dubotovkin


SDK
Common
Android
Target
JS
Target
Android library
sdk.aar
Webpack bundle
sdk.js
Android app
App
Web app
web

конец

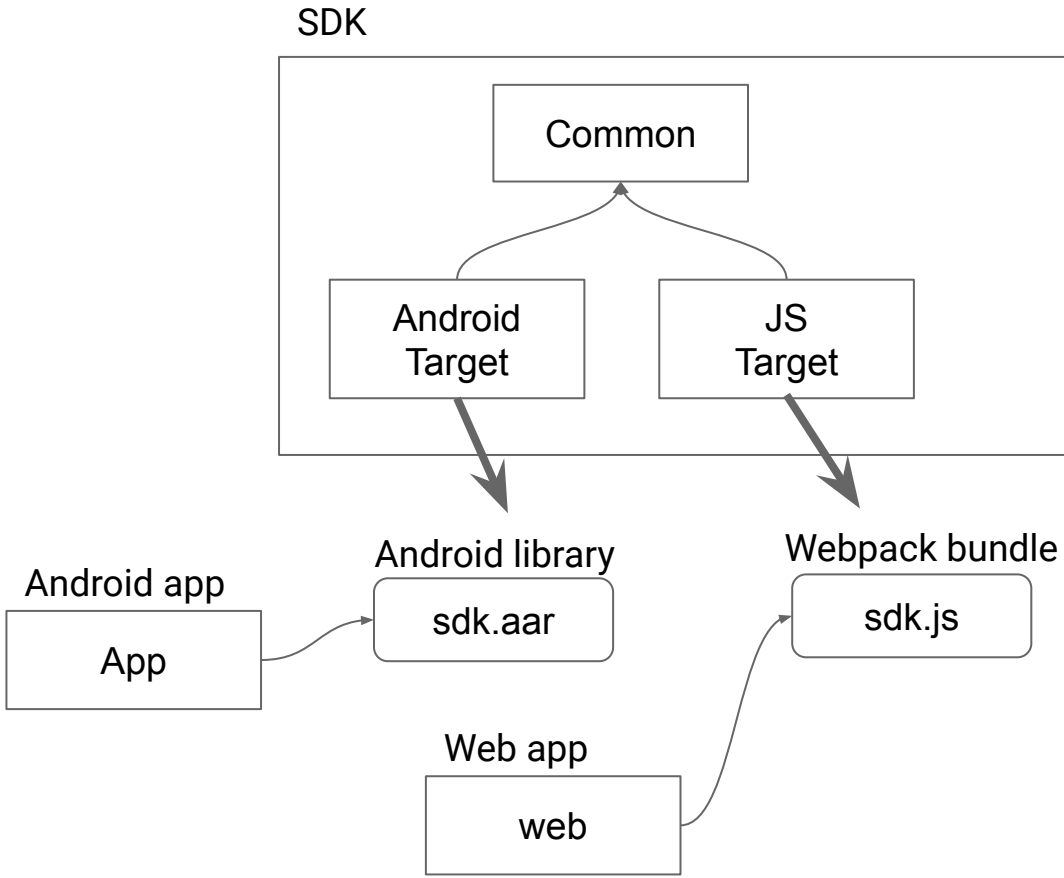
первой части

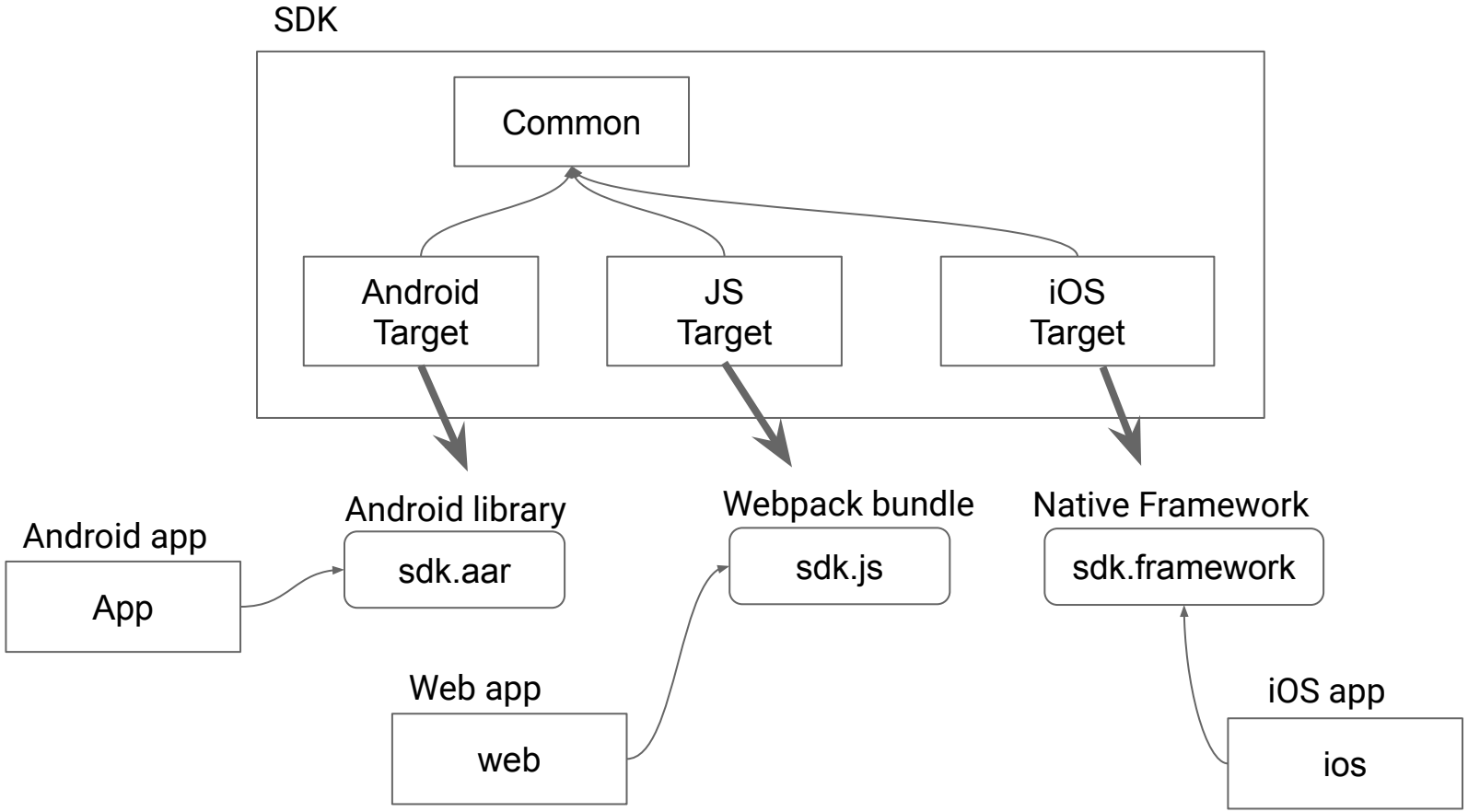
Запуск реального андроид приложения на iOS и в браузере

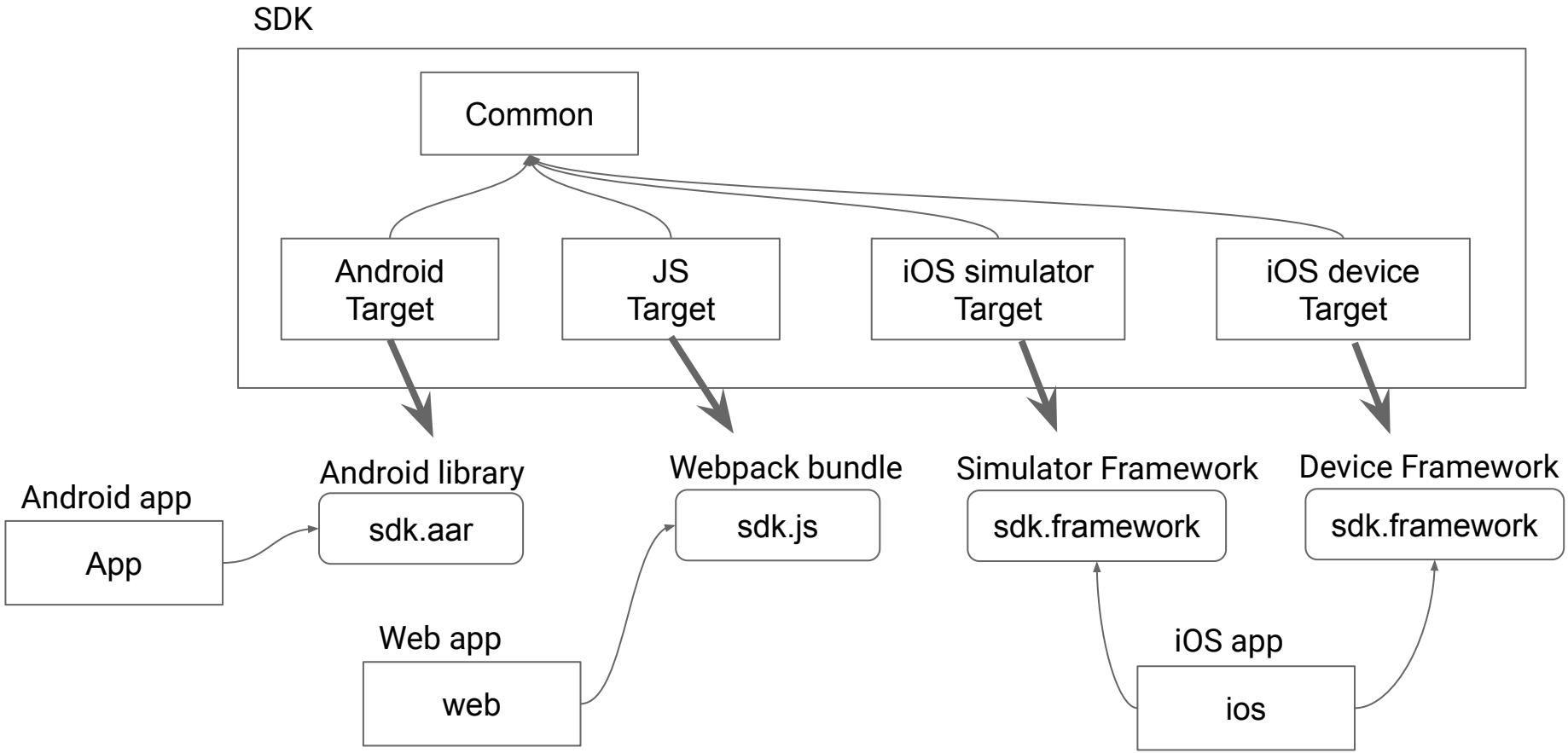
Константин Цховребов
JetBrains

Часть #2

Native (iOS)


SDK
Common
Android
Target
JS
Target
Android library
sdk.aar
Webpack bundle
sdk.js
Android app
App
Web app
web


SDK
Common
Android Target
JS Target
iOS Target
Android library
sdk.aar
Webpack bundle
sdk.js
Native Framework
sdk.framework
Android app
App
Web app
web
iOS app
ios


SDK
Common
Android Target
JS Target
iOS simulator Target
iOS device Target
Android app
App
Android library
sdk.aar
Webpack bundle
sdk.js
Simulator Framework
sdk.framework
Device Framework
sdk.framework
Web app
web
iOS app
ios

Hands-on

[Targeting iOS and Android with Kotlin Multiplatform]

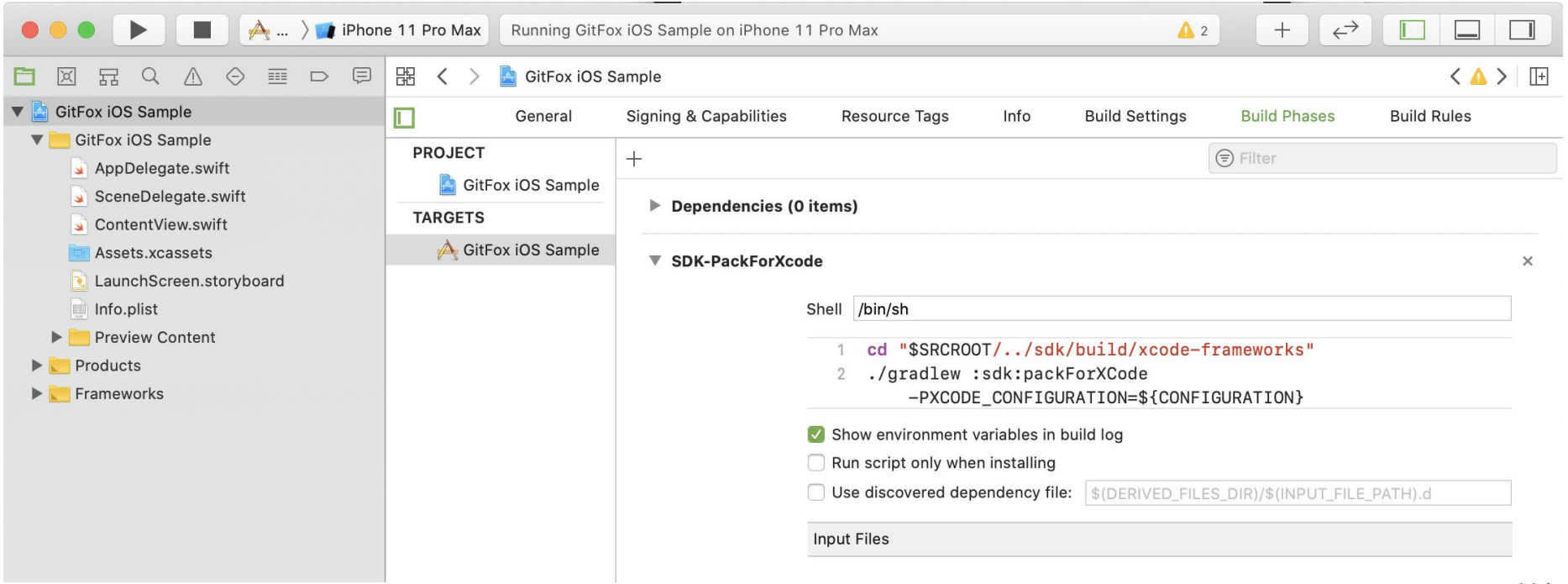
```
//select iOS target platform depending on the Xcode environment variables
val iOSTarget: (String, KotlinNativeTarget.() -> Unit) -> KotlinNativeTarget =
    if (System.getenv("SDK_NAME")?.startsWith("iphoneos") == true) ::iosArm64
    else ::iosX64

iOSTarget("ios") {
    binaries {
        framework {
            baseName = "GitFoxSDK"
        }
    }
}
```

Автоматический выбор таргета

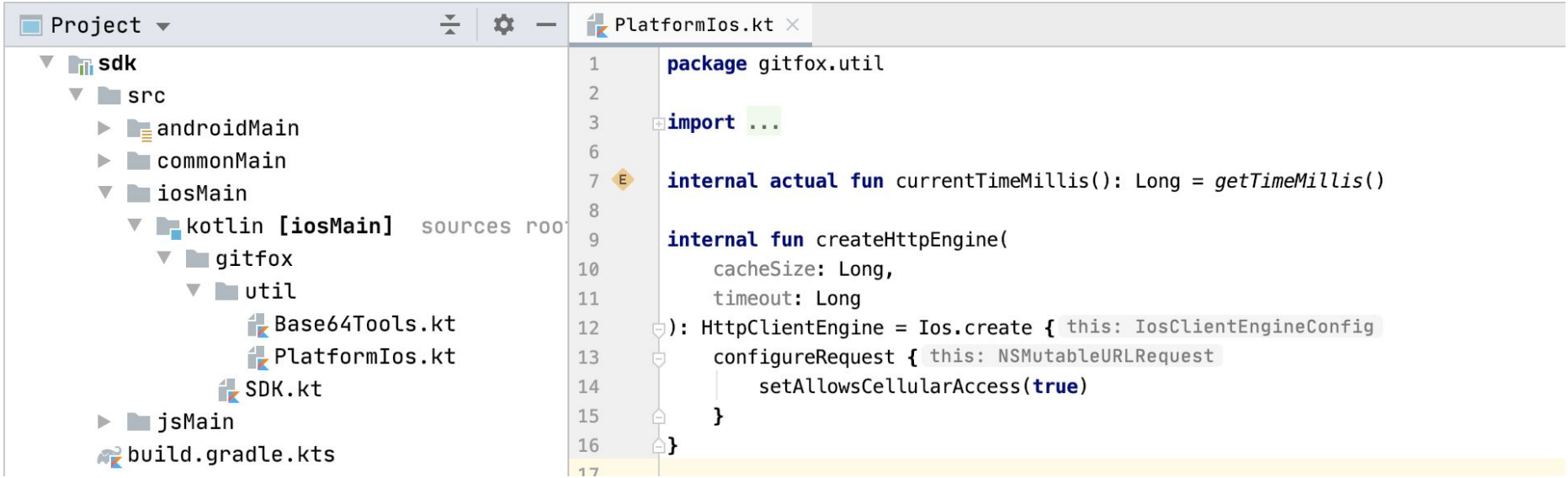
```
val packForXcode by tasks.creating(Sync::class) {
    val targetDir = File(buildDir, "xcode-frameworks")

    /// selecting the right configuration for the iOS
    /// framework depending on the environment
    /// variables set by Xcode build
    val mode = System.getenv("CONFIGURATION") ?: "DEBUG"
    val framework = kotlin.targets
        .getByName<KotlinNativeTarget>("ios")
        .binaries.getFramework(mode)
    inputs.property("mode", mode)
    dependsOn(framework.linkTask)

    from({ framework.outputDirectory })
    into(targetDir)

    /// generate a helpful ./gradlew wrapper with embedded Java path
    doLast {...}
}

tasks.getByName("build").dependsOn(packForXcode)
```

Вспомогательная gradle таска

Подключение к iOS проекту



Платформенная часть



./gradlew :sdk:linkDebugFrameworkIos

```
e: java.lang.IllegalStateException:
org.jetbrains.kotlin.ir.descriptors.WrappedPropertyDescriptor@21acea72
is not bound
```

Konstantin Tskhovrebov 10:54 PM

Привет! У меня что-то развалилось где не должно было 🙂

воспроизвести просто:

```
git clone git@gitlab.com:terrakok/gitlab-client.git
cd ./gitlab-client
git checkout sdk-ios
./gradlew :sdk:linkDebugFrameworkIos
```

(edited)

Untitled

```
▶ ./gradlew :sdk:linkDebugFrameworkIos

> Configure project :sdk
Kotlin Multiplatform Projects are an experimental feature.

```

 **51 replies** Last reply 29 days ago

Kotlinx-serialization

```
@Serializer(forClass = Color::class)
internal object ColorDeserializer : KSerializer<Color> {

    override fun deserialize(decoder: Decoder): Color =
        Color(decoder.decodeString())
}
```

7285d3af

Kotlinx-serialization

```
@Serializer(forClass = Color::class)
internal object ColorDeserializer : KSerializer<Color> {

    override fun deserialize(decoder: Decoder): Color =
        Color(decoder.decodeString())

    override fun serialize(encoder: Encoder, value: Color) {
        encoder.encodeString(value.name)
    }
}
```

7285d3af

Быстрая проверка

```
fun run() = runBlocking {
    sdk.getSessionInteractor().loginOnCustomServer(
        "https://gitlab.com/",
        "..."
    )
    sdk.getProjectInteractor().getProject(42)
}
```

```
//В элементарном иОС проекте

import GitFoxSDK

let sdk = IosSDK.init(...)
sdk.run()
```

запускаем симулятор из Xcode...

InvalidMutabilityException

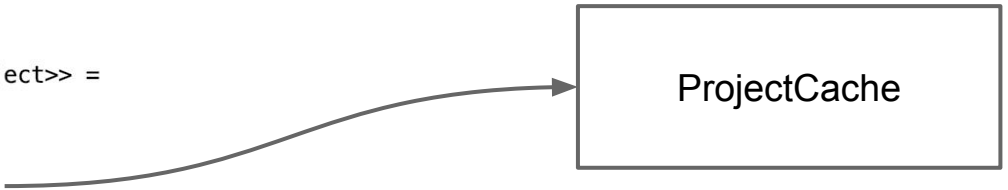
Concurrency in Kotlin/Native

[KotlinConf 2018:
Kotlin/Native Concurrency Model by Nikolay Igotti]

[KotlinConf 2019:
Kotlin Native Concurrency Explained by Kevin Galligan]

```
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
            .getProjects(...)
            .also { projectCache.put(it) }

    override suspend fun getProject(...): Project =
        projectCache.get(id) ?: serverApi.getProject(id, statistics)
            .also { projectCache.put(listOf(it)) }
}
```

```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }

    override fun getProject(
        id: Long,
        statistics: Boolean
    ): Single<Project> =
        Single
            .defer {
                val cachedProject = projectCache.get(id)
                if (cachedProject == null) {
                    serverApi.getProject(id, statistics)
                        .doOnSuccess { projectCache.put(listOf(it)) }
                } else {
                    Single.just(cachedProject)
                }
            }
}
```



Кеширующие запросы когда-то давно в приложении


```
private val cache = ConcurrentHashMap<Long, ProjectCacheItem>()
```

Синхронизация доступа на запись

Синхронизация доступа на запись

Блокировки на запись: **@Synchronized** или семафоры

Чтобы не писать самому, можно использовать готовые коллекции из **`java.util.concurrent.*`**

Синхронизация доступа на запись

Блокировки на запись: **@Synchronized** или семафоры

Чтобы не писать самому, можно использовать готовые коллекции из **`java.util.concurrent.*`**

Очереди исполнения на одном потоке.

Очередь исполнения на одном потоке.

```
private val cache = mutableMapOf<Long, Project>()
```

Очередь исполнения на одном потоке.

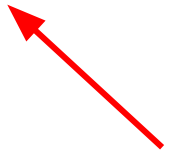
```
private val cache = mutableMapOf<Long, Project>()

suspend fun put(data: List<Project>) {
    withContext(cacheDispatcher) {
        map.putAll(data)
    }
}
```

Очередь исполнения на одном потоке.

```
private val cacheDispatcher = createSingleThreadDispatcher("cache")

private val cache = mutableMapOf<Long, Project>()

suspend fun put(data: List<Project>) {
    withContext(cacheDispatcher) {
        map.putAll(data)
    }
}
```

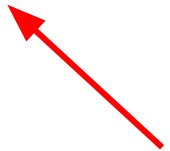
Очередь исполнения на одном потоке.

```
private val cacheDispatcher = createSingleThreadDispatcher("cache")

private val cache = mutableMapOf<Long, Project>()

suspend fun put(data: List<Project>) {
    withContext(cacheDispatcher) {
        map.putAll(data)
    }
}
```



Очередь исполнения на одном потоке.

```
private val cacheDispatcher = createSingleThreadDispatcher("cache")

private val cache = mutableMapOf<Long, Project>()

suspend fun put(data: List<Project>) {
    withContext(cacheDispatcher) {
        map.putAll(data)
    }
}

internal expect fun createSingleThreadDispatcher(name: String): CoroutineDispatcher
```



```
//JVM
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    newSingleThreadContext(name)
```

```
//JVM
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    newSingleThreadContext(name)

//JavaScript
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    Dispatchers.Main
```

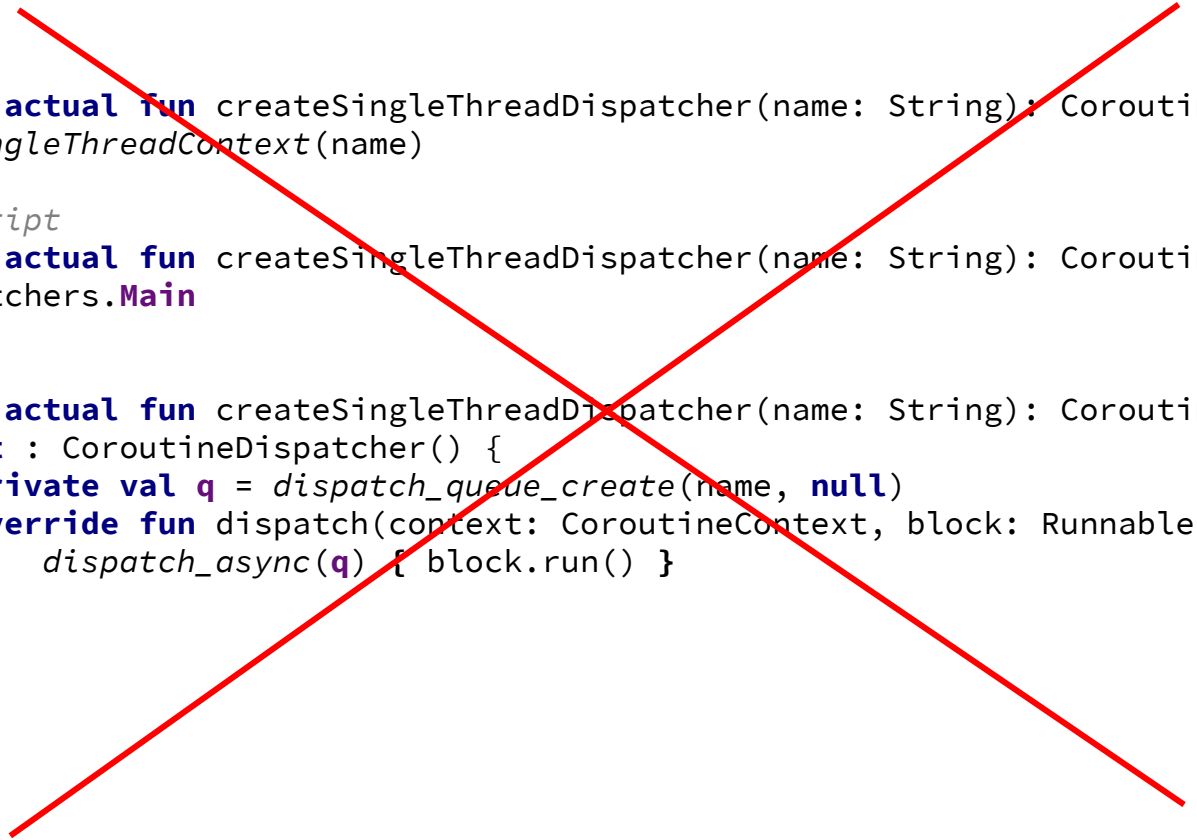
```
//JVM
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
   newSingleThreadContext(name)

//JavaScript
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
   Dispatchers.Main

//Native
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
   object : CoroutineDispatcher() {
       private val q = dispatch_queue_create(name, null)
       override fun dispatch(context: CoroutineContext, block: Runnable) {
           dispatch_async(q) { block.run() }
       }
   }
```

https://developer.apple.com/library/archive/documentation/General/Conceptual/ConcurrencyProgrammingGuide/OperationQueues/OperationQueues.html

```
//JVM
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    newSingleThreadContext(name)

//JavaScript
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    Dispatchers.Main

//Native
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    object : CoroutineDispatcher() {
        private val q = dispatch_queue_create(name, null)
        override fun dispatch(context: CoroutineContext, block: Runnable) {
            dispatch_async(q) { block.run() }
        }
    }
```

InvalidMutabilityException

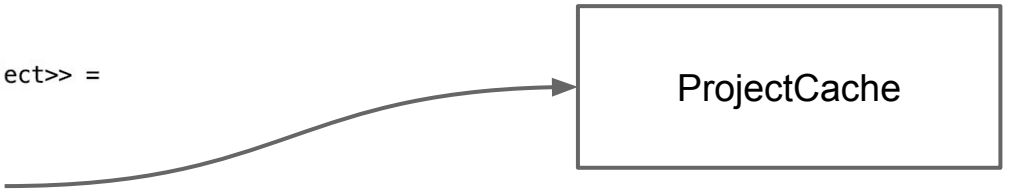
```
//JVM
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    newSingleThreadContext(name)

//JavaScript
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    Dispatchers.Main

//Native
internal actual fun createSingleThreadDispatcher(name: String): CoroutineDispatcher =
    object : CoroutineDispatcher() {
        private val q = dispatch_queue_create(name, null)
        override fun dispatch(context: CoroutineContext, block: Runnable) {
            dispatch_async(q) { block.run() }
        }
    }
```



SingleThreadDispatcher - не делайте так!!!

Потоки в RxJava

```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }
```



```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }
```

Кеширующие запросы когда-то давно в приложении

```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    apiWithProjectCache.getProjects(...)
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe { projects, err ->
            renderUI(projects)
        }
}
```

Кеширующие запросы когда-то давно в приложении

```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    apiWithProjectCache.getProjects(...)
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe { projects, err ->
            renderUI(projects)
        }
}
```

Кеширующие запросы когда-то давно в приложении

```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    apiWithProjectCache.getProjects(...)
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe { projects, err ->
            renderUI(projects)
        }
}
```

Кеширующие запросы когда-то давно в приложении

```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }
```

???

UI

```
fun getProjects(...) {
    apiWithProjectCache.getProjects(...)
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe { projects, err ->
            renderUI(projects)
        }
}
```

Кеширующие запросы когда-то давно в приложении

RxJava2 Adapter

An `Adapter` for adapting RxJava 2.x types.

Your service methods can now use any of the above types as their return type.

```
interface MyService {
  @GET("/user")
  Observable<User> getUser();
}
```

By default all reactive types execute their requests synchronously. There are multiple ways to control the threading on which a request occurs:

- Call `subscribeOn` on the returned reactive type with a `Scheduler` of your choice.

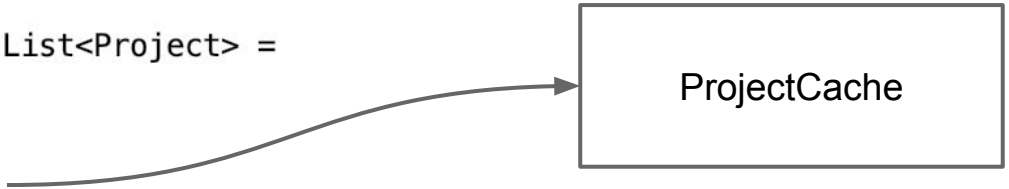
```
class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override fun getProjects(...): Single<List<Project>> =
        serverApi
            .getProjects(...)
            .doOnSuccess { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    apiWithProjectCache.getProjects(...)
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread())
        .subscribe { projects, err ->
            renderUI(projects)
        }
}
```

Потоки в Coroutines

```
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
            .getProjects(...)
            .also { projectCache.put(it) }
```



```kotlin
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
            .getProjects(...)
            .also { projectCache.put(it) }
```

```
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
            .getProjects(...)
            .also { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    UiScope.launch { this: CoroutineScope
        val projects = apiWithProjectCache.getProjects(...)
        renderUI(projects)
    }
}
```

```
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
            .getProjects(...)
            .also { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    UiScope.launch { this: CoroutineScope
        val projects = apiWithProjectCache.getProjects(...)
        renderUI(projects)
    }
}
```

```
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
            .getProjects(...)
            .also { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    UiScope.launch { this: CoroutineScope
        val projects = apiWithProjectCache.getProjects(...)
        renderUI(projects)
    }
}
```

```
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
???         .getProjects(...)
            .also { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    UiScope.launch { this: CoroutineScope
        val projects = apiWithProjectCache.getProjects(...)
        renderUI(projects)
    }
}
```

Search... (press 's' to focus, or '#' for sections)

Http Client

Estimated reading time: 2 minutes

Edit Page

In addition to HTTP serving, Ktor also includes a flexible asynchronous HTTP client. This client supports several configurable engines, and has its own set of features.

> ARTIFACT
>
> The main functionality is available through the `io.ktor:ktor-client-core:$ktor_version` artifact. And each engine, is provided in separate artifacts.

Table of contents:

Calls: Requests and Responses

You can check how to make requests, and how to receive responses in their respective sections.

Concurrency

Remember that requests are asynchronous, but when performing requests, the API suspends further requests and your function will be suspended until done. If you want to perform several requests at once in the same block, you can use `launch` or `async` functions and later get the results. For example:

```
/**
 * Create call context and use it as a coroutine context to [execute] request.
 */
private suspend fun executeWithinCallContext(requestData: HttpRequestData): HttpResponseData {
    val callContext = createCallContext(requestData.executionContext)

    return async( context: callContext + KtorCallContextElement(callContext)) { this: CoroutineScope
        if (closed) {
            throw ClientEngineClosedException()
        }

        execute(requestData) ^async
    }.await()
}
```

```
/**
 * Create call context and use it as a coroutine context to [execute] request.
 */
private suspend fun executeWithinCallContext(requestData: HttpRequestData): HttpResponseData {
    val callContext = createCallContext(requestData.executionContext)

    return async( context: callContext + KtorCallContextElement(callContext)) { this: CoroutineScope
        if (closed) {
            throw ClientEngineClosedException()
        }

        execute(requestData) ^async
    }.await()
}
```

```
internal class ApiWithProjectCache(
    private val serverApi: GitlabApi,
    private val projectCache: ProjectCache
) : GitlabApi by serverApi {

    override suspend fun getProjects(...): List<Project> =
        serverApi
            .getProjects(...)
            .also { projectCache.put(it) }
```

UI

```
fun getProjects(...) {
    UiScope.launch { this: CoroutineScope
        val projects = apiWithProjectCache.getProjects(...)
        renderUI(projects)
    }
}
```

From RxJava 2 to Kotlin Flow: Threading

Comparing threading in RxJava 2 and Kotlin Flow

Vasya Drobushkov Follow

Jan 25 · 15 min read

https://proandroiddev.com/from-rxjava-2-to-kotlin-flow-threading-8618867e1955

Для большинства задач
в мобильных клиентах
нет необходимости
работать с
многопоточностью

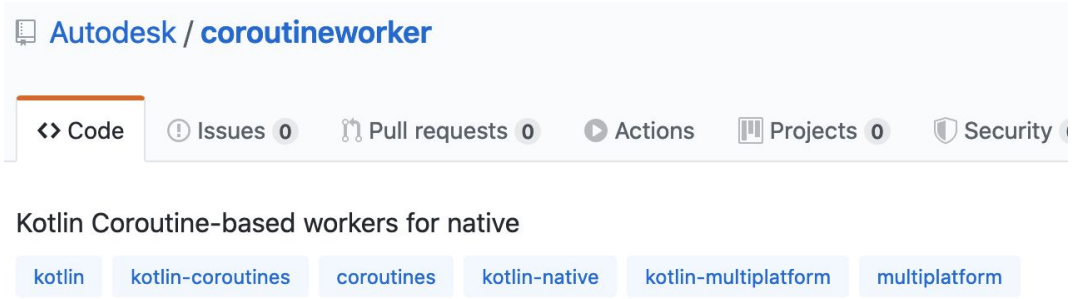
- На однопоточном JavaScript работают клиенты Slack, Spotify и другие, а они совсем не простые.

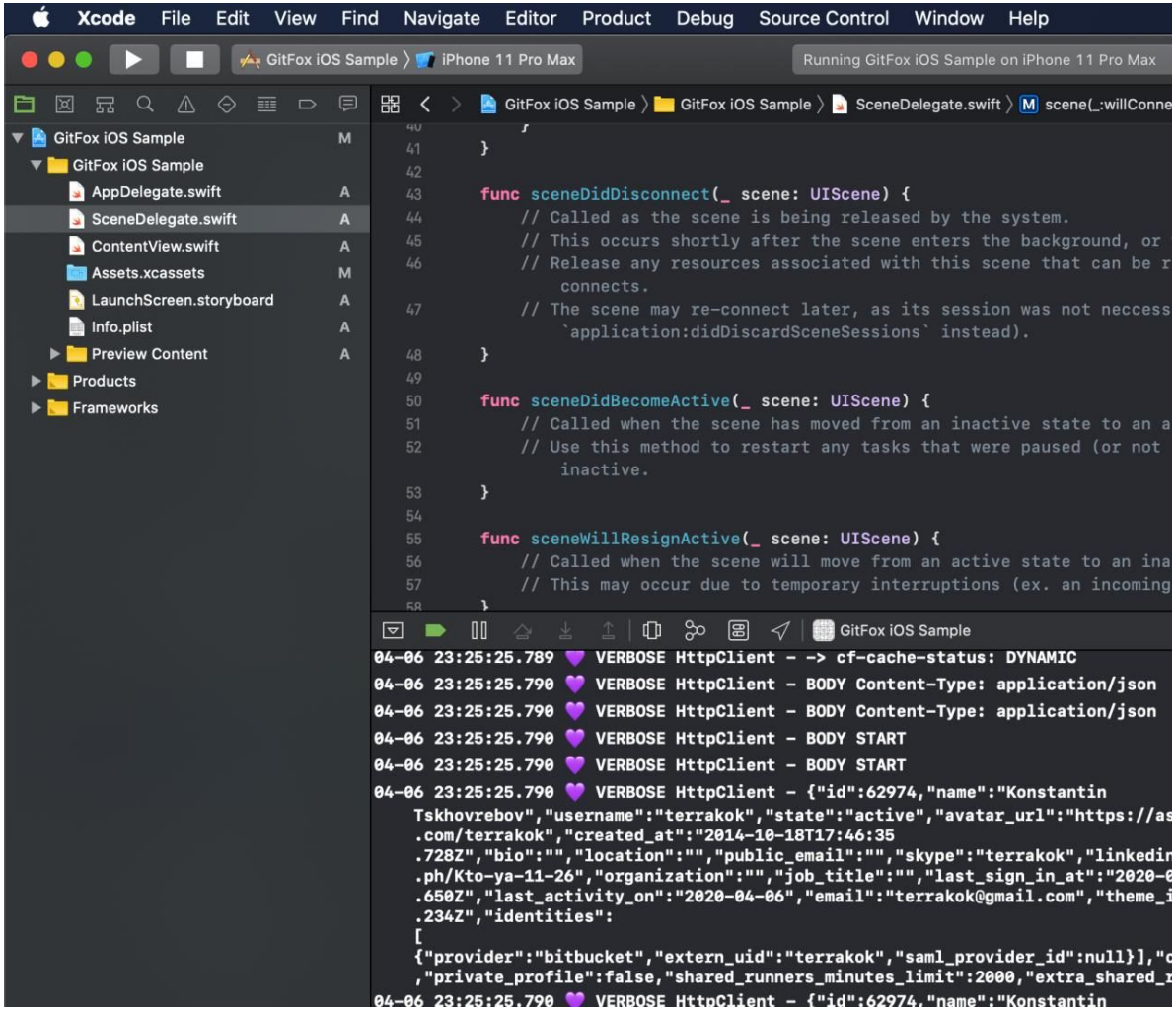
- На однопоточном JavaScript работают клиенты Slack, Spotify и другие, а они совсем не простые.
- Большой пользы не будет от парсинга мелких json’ов параллельно, а может быть и хуже.

- На однопоточном JavaScript работают клиенты Slack, Spotify и другие, а они совсем не простые.
- Большой пользы не будет от парсинга мелких json’ов параллельно, а может быть и хуже.
- Достаточно вынести работу с сетью и файлами на отдельные потоки, а для простых расчетов хватит и UI фреймов.

- На однопоточном JavaScript работают клиенты Slack, Spotify и другие, а они совсем не простые.
- Большой пользы не будет от парсинга мелких json’ов параллельно, а может быть и хуже.
- Достаточно вынести работу с сетью и файлами на отдельные потоки, а для простых расчетов хватит и UI фреймов.

А если не хватит, то всегда можно реализовать “многопоточность”, просто коробочное решение для корутин пока экспериментальное.




Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
GitFox iOS Sample 〉 iPhone 11 Pro Max
Running GitFox iOS Sample on iPhone 11 Pro Max
GitFox iOS Sample M
GitFox iOS Sample
AppDelegate.swift A
SceneDelegate.swift A
ContentView.swift A
Assets.xcassets M
LaunchScreen.storyboard A
Info.plist A
Preview Content A
Products
Frameworks
GitFox iOS Sample 〉 GitFox iOS Sample 〉 SceneDelegate.swift 〉 scene(_:willConne
}
func sceneDidDisconnect(_ scene: UIScene) {
// Called as the scene is being released by the system.
// This occurs shortly after the scene enters the background, or
// Release any resources associated with this scene that can be r
connects.
// The scene may re-connect later, as its session was not neccess
`application:didDiscardSceneSessions` instead).
}
func sceneDidBecomeActive(_ scene: UIScene) {
// Called when the scene has moved from an inactive state to an a
// Use this method to restart any tasks that were paused (or not
inactive.
}
func sceneWillResignActive(_ scene: UIScene) {
// Called when the scene will move from an active state to an ina
// This may occur due to temporary interruptions (ex. an incoming
GitFox iOS Sample
04-06 23:25:25.789 VERBOSE HttpClient - -> cf-cache-status: DYNAMIC
04-06 23:25:25.790 VERBOSE HttpClient - BODY Content-Type: application/json
04-06 23:25:25.790 VERBOSE HttpClient - BODY Content-Type: application/json
04-06 23:25:25.790 VERBOSE HttpClient - BODY START
04-06 23:25:25.790 VERBOSE HttpClient - BODY START
04-06 23:25:25.790 VERBOSE HttpClient - {"id":62974,"name":"Konstantin
Tskhovrebov","username":"terrakok","state":"active","avatar_url":"https://as
.com/terrakok","created_at":"2014-10-18T17:46:35
.728Z","bio":"","location":"","public_email":"","skype":"terrakok","linkedin
.ph/Kto-ya-11-26","organization":"","job_title":"","last_sign_in_at":"2020-0
.650Z","last_activity_on":"2020-04-06","email":"terrakok@gmail.com","theme_i
.234Z","identities":
[
{"provider":"bitbucket","extern_uid":"terrakok","saml_provider_id":null}],"c
,"private_profile":false,"shared_runners_minutes_limit":2000,"extra_shared_r
04-06 23:25:25.790 VERBOSE HttpClient - {"id":62974,"name":"Konstantin

Framework для iOS

iOS не знает про suspend

```
class CommitInteractor internal constructor(
    private val api: GitlabApi
) {

    suspend fun getCommit(projectId: Long, commitId: String): Commit =
        api.getRepositoryCommit(projectId, commitId)

    suspend fun getCommitDiffData(projectId: Long, commitId: String): List<DiffData> =
        api.getCommitDiffData(projectId, commitId)
}
```

iOS callbacks

```
class IosCommitInteractor internal constructor(private val interactor: CommitInteractor) {

    fun getCommit(
        projectId: Long,
        commitId: String,
        callback: (result: Commit?, error: Exception?) -> Unit
    ) {
        //...
    }

    fun getCommitDiffData(
        projectId: Long,
        commitId: String,
        callback: (result: List<DiffData>?, error: Exception?) -> Unit
    ) {
        //...
    }
}
```

43ee05fc

Suspend -> callback

```
internal fun <T> CoroutineScope.wrap(
    callback: (result: T?, error: Exception?) -> Unit,
    block: suspend () -> T
) {
    launch {
        try {
            callback(block(), null)
        } catch (e: Exception) {
            callback(null, e)
        }
    }
}
```

43ee05fc

MainLoopDispatcher

```
import platform.darwin.*

internal object MainLoopDispatcher : CoroutineDispatcher(), Delay {

    override fun dispatch(context: CoroutineContext, block: Runnable) {
        dispatch_async(dispatch_get_main_queue()) {
            try {
                block.run()
            } catch (err: Throwable) {
                Napier.e("UNCAUGHT " + err.message, err)
                throw err
            }
        }
    }
    //...
}
```

43ee05fc

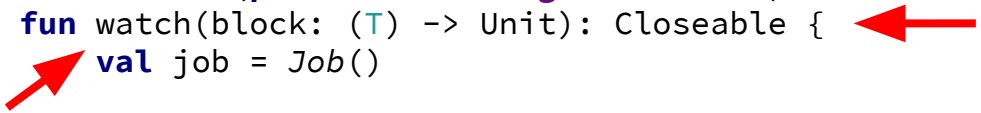
iOS callbacks

```
class IosCommitInteractor internal constructor(
    private val interactor: CommitInteractor
) : CoroutineScope by CoroutineScope(MainLoopDispatcher) {

    fun getCommit(
        projectId: Long,
        commitId: String,
        callback: (result: Commit?, error: Exception?) -> Unit
    ) {
        wrap(callback) { interactor.getCommit(projectId, commitId) }
    }

    fun getCommitDiffData(
        projectId: Long,
        commitId: String,
        callback: (result: List<DiffData>?, error: Exception?) -> Unit
    ) {
        wrap(callback) { interactor.getCommitDiffData(projectId, commitId) }
    }
}
```

43ee05fc

Flow в iOS

```
class CFlow<T>(private val origin: Flow<T>) : Flow<T> by origin {
    fun watch(block: (T) -> Unit): Closeable {
        val job = Job()

        onEach {
            block(it)
        }.launchIn(CoroutineScope(MainLoopDispatcher + job))

        return object : Closeable {
            override fun close() {
                job.cancel()
            }
        }
    }
}

internal fun <T> Flow<T>.wrap(): CFlow<T> = CFlow(this)
```

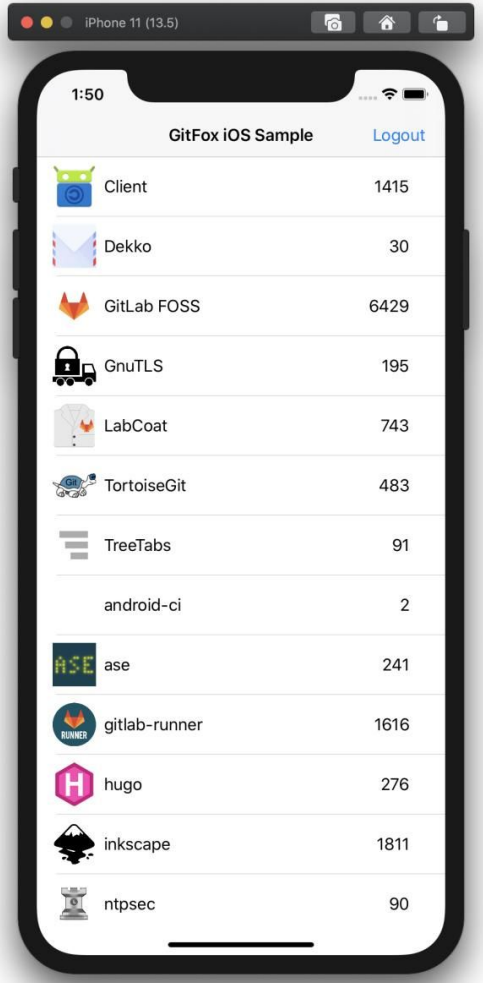


ef86c7ce

Вызов из iOS приложения

```
import GitFoxSDK

let sdk = IosSDK.init(
    oAuthParams: OAuthParams.init(
        endpoint: "https://gitlab.com/",
        appId: "appId",
        appKey: "appKey",
        redirectUrl: "redirectUrl"
   ),
   isDebug: true
)

sdk.getSessionInteractor().loginOnCustomServer(
    serverPath: "https://gitlab.com/",
    token: "put real private token!"
) { result, err in
    if err == nil {
        //at the moment you are logged in
        sdk.getProjectInteractor().getProject(id: 2977308) { result, err in
            if let project = result {
                print(project)
            } else {
                print("error: " + err!.message!)
            }
        }
    } else {
        print("error: " + err!.message!)
    }
}
```

Ilyas Siraev



Выводы

Выводы

- это возможно!

Выводы

- это возможно!
- стандартный стек уже есть в виде библиотек

Выводы

- это возможно!
- стандартный стек уже есть в виде библиотек
- проблемы можно решить относительно небольшими компромиссами

Выводы

- это возможно!
- стандартный стек уже есть в виде библиотек
- проблемы можно решить относительно небольшими компромиссами
- команда JetBrains открыта для вопросов

Выводы

- это возможно!
- стандартный стек уже есть в виде библиотек
- проблемы можно решить относительно небольшими компромиссами
- команда JetBrains открыта для вопросов
- большие перспективы у технологии

Выводы

- это возможно!
- стандартный стек уже есть в виде библиотек
- проблемы можно решить относительно небольшими компромиссами
- команда JetBrains открыта для вопросов
- большие перспективы у технологии
- относительно безопасный эксперимент

Выводы

- это возможно!
- стандартный стек уже есть в виде библиотек
- проблемы можно решить относительно небольшими компромиссами
- команда JetBrains открыта для вопросов
- большие перспективы у технологии
- относительно безопасный эксперимент
- теперь есть последовательный пример без “белых пятен”

Логичные вопросы

- Оправданно ли для тонких клиентов?
- Можно ведь было все на Flutter написать!

Оправданно ли для тонких клиентов?

- меньше писать бойлерплейт

Оправданно ли для тонких клиентов?

- меньше писать бойлерплейт
- меньше багов, так как только одна кодовая база

Оправданно ли для тонких клиентов?

- меньше писать бойлерплейт
- меньше багов, так как только одна кодовая база
- можно сосредоточиться на хорошем UI и UX

Оправданно ли для тонких клиентов?

- меньше писать бойлерплейт
- меньше багов, так как только одна кодовая база
- можно сосредоточиться на хорошем UI и UX
- даже в тонком клиенте есть логика:
  - кеширование
  - управление сессиями
  - оптимизация запросов

Оправданно ли для тонких клиентов?

- меньше писать бойлерплейт
- меньше багов, так как только одна кодовая база
- можно сосредоточиться на хорошем UI и UX
- даже в тонком клиенте есть логика:
  - кеширование
  - управление сессиями
  - оптимизация запросов
- минимальный оверхед для поддержки будущей мультиплатформы с самого начала в андроид проекте

Можно ведь было все на Flutter написать!

- я не думал о других платформах, кроме Android

Можно ведь было все на Flutter написать!

- я не думал о других платформах, кроме Android
- я не знаю Dart

Можно ведь было все на Flutter написать!

- я не думал о других платформах, кроме Android
- я не знаю Dart
- уже есть рабочий клиент, зачем писать с нуля?

Можно ведь было все на Flutter написать!

- я не думал о других платформах, кроме Android
- я не знаю Dart
- уже есть рабочий клиент, зачем писать с нуля?
- точно не будет проблем с производительностью UI

Можно ведь было все на Flutter написать!

- я не думал о других платформах, кроме Android
- я не знаю Dart
- уже есть рабочий клиент, зачем писать с нуля?
- точно не будет проблем с производительностью UI
- это не фреймворк, можно собрать и под другие платформы

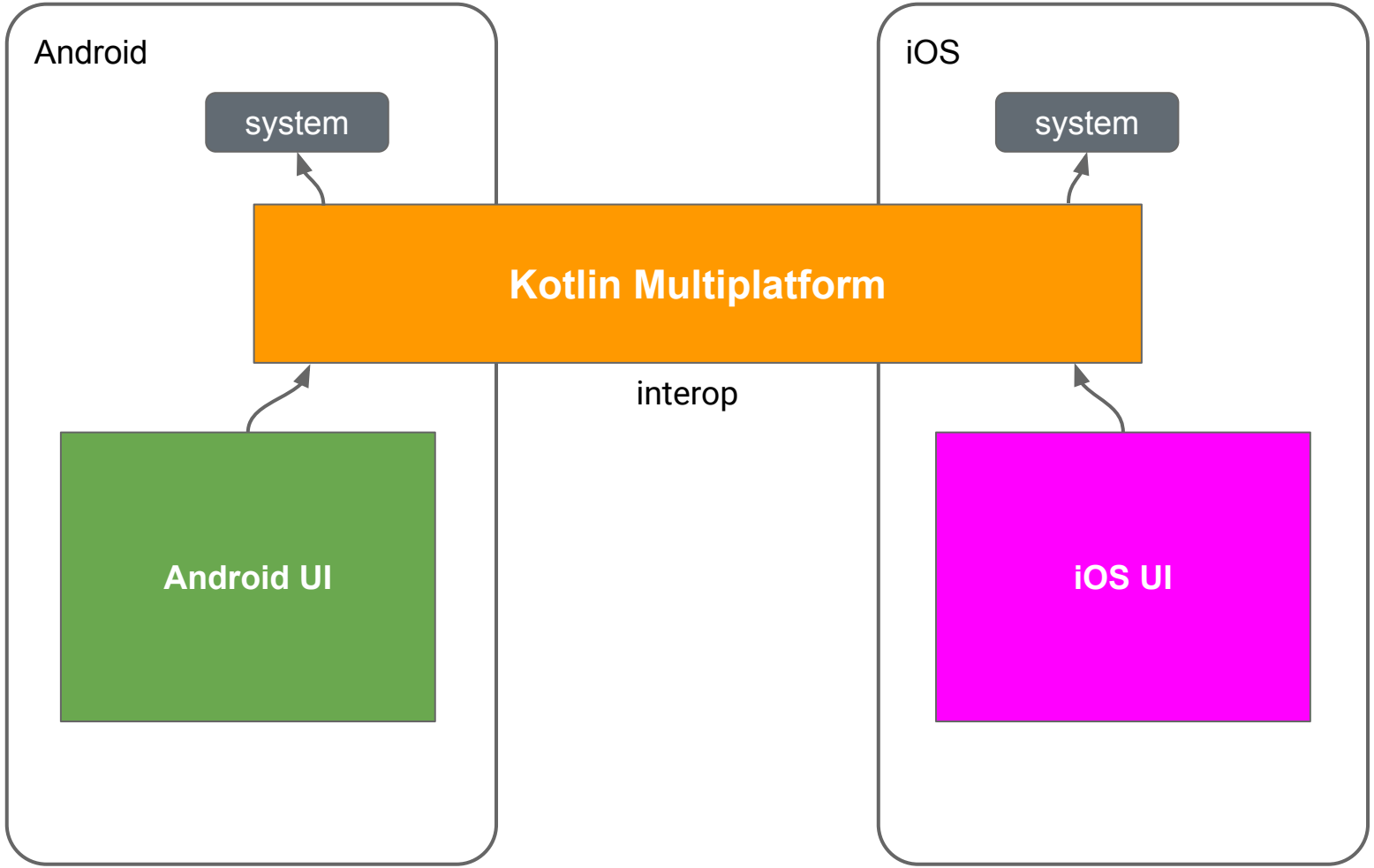
Можно ведь было все на Flutter написать!

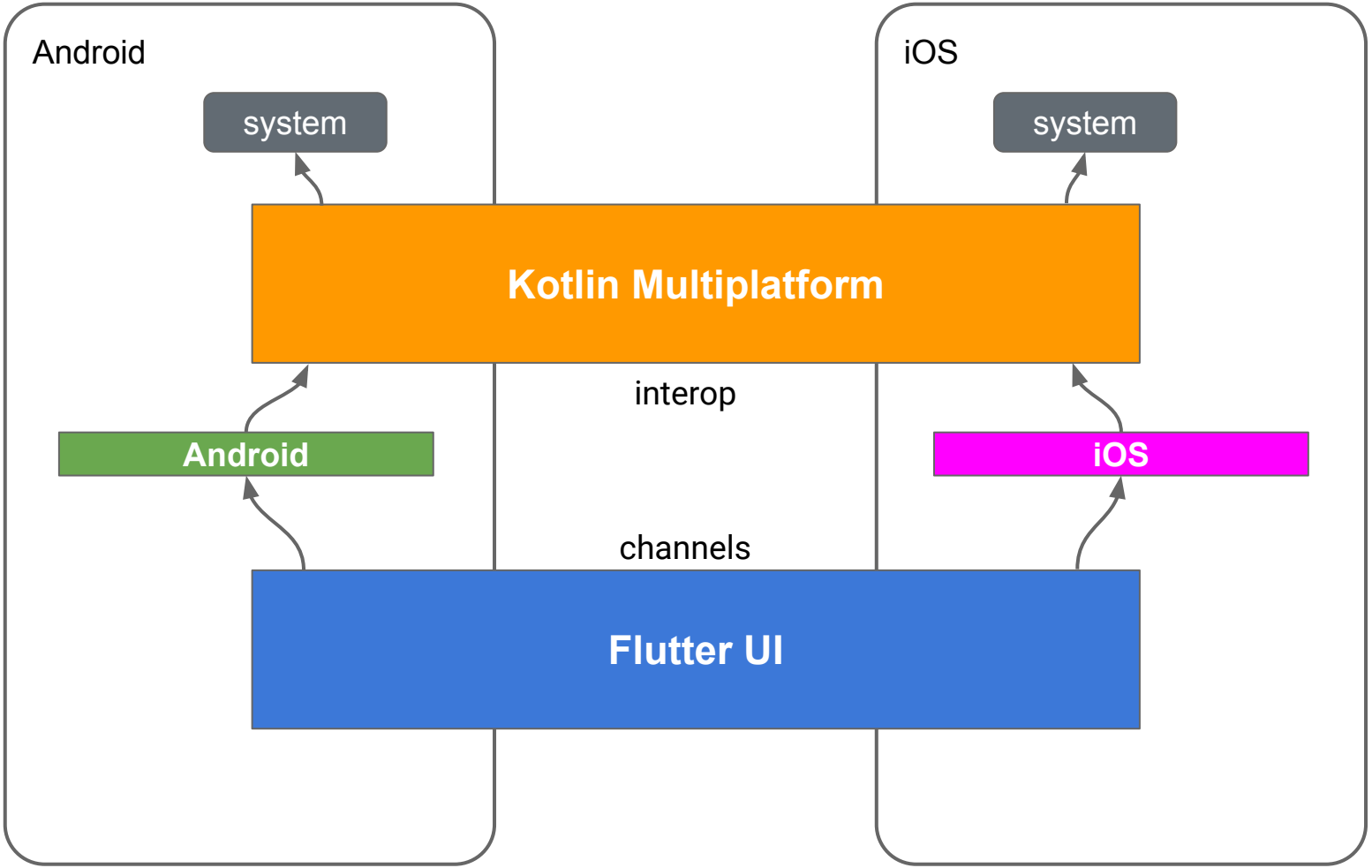
- я не думал о других платформах, кроме Android
- я не знаю Dart
- уже есть рабочий клиент, зачем писать с нуля?
- точно не будет проблем с производительностью UI
- это не фреймворк, можно собрать и под другие платформы
- на Flutter я не могу написать только часть приложения

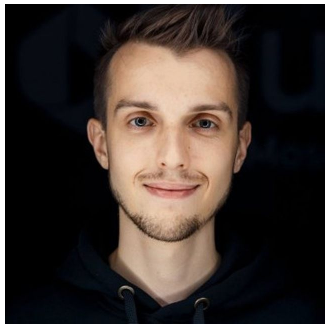
Можно ведь было все на Flutter написать!

- я не думал о других платформах, кроме Android
- я не знаю Dart
- уже есть рабочий клиент, зачем писать с нуля?
- точно не будет проблем с производительностью UI
- это не фреймворк, можно собрать и под другие платформы
- на Flutter я не могу написать только часть приложения
- на Flutter делают полностью приложение, а не библиотеку к сервису

Бонус #1: общий UI

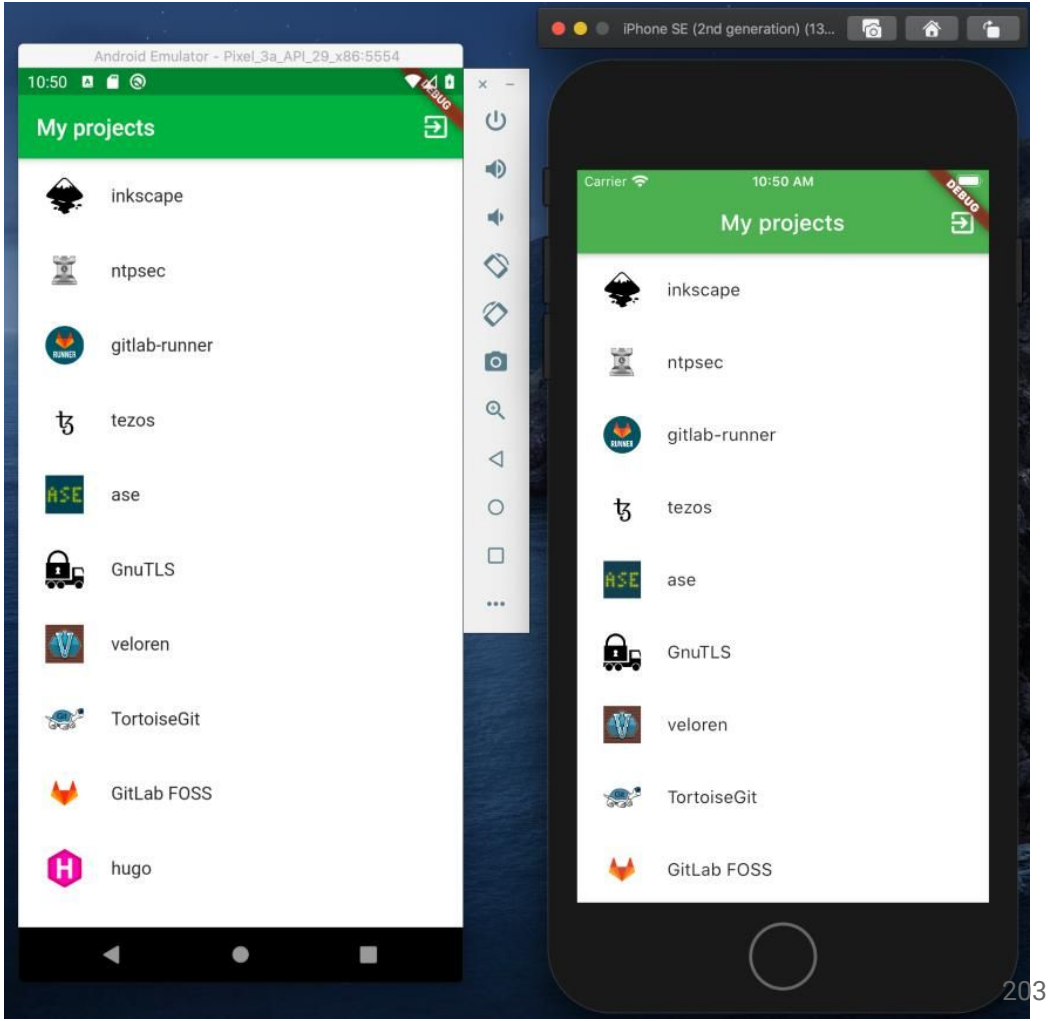

Android
system
iOS
system
Kotlin Multiplatform
interop
Android UI
iOS UI


Android
system
iOS
system
Kotlin Multiplatform
interop
Android
iOS
channels
Flutter UI

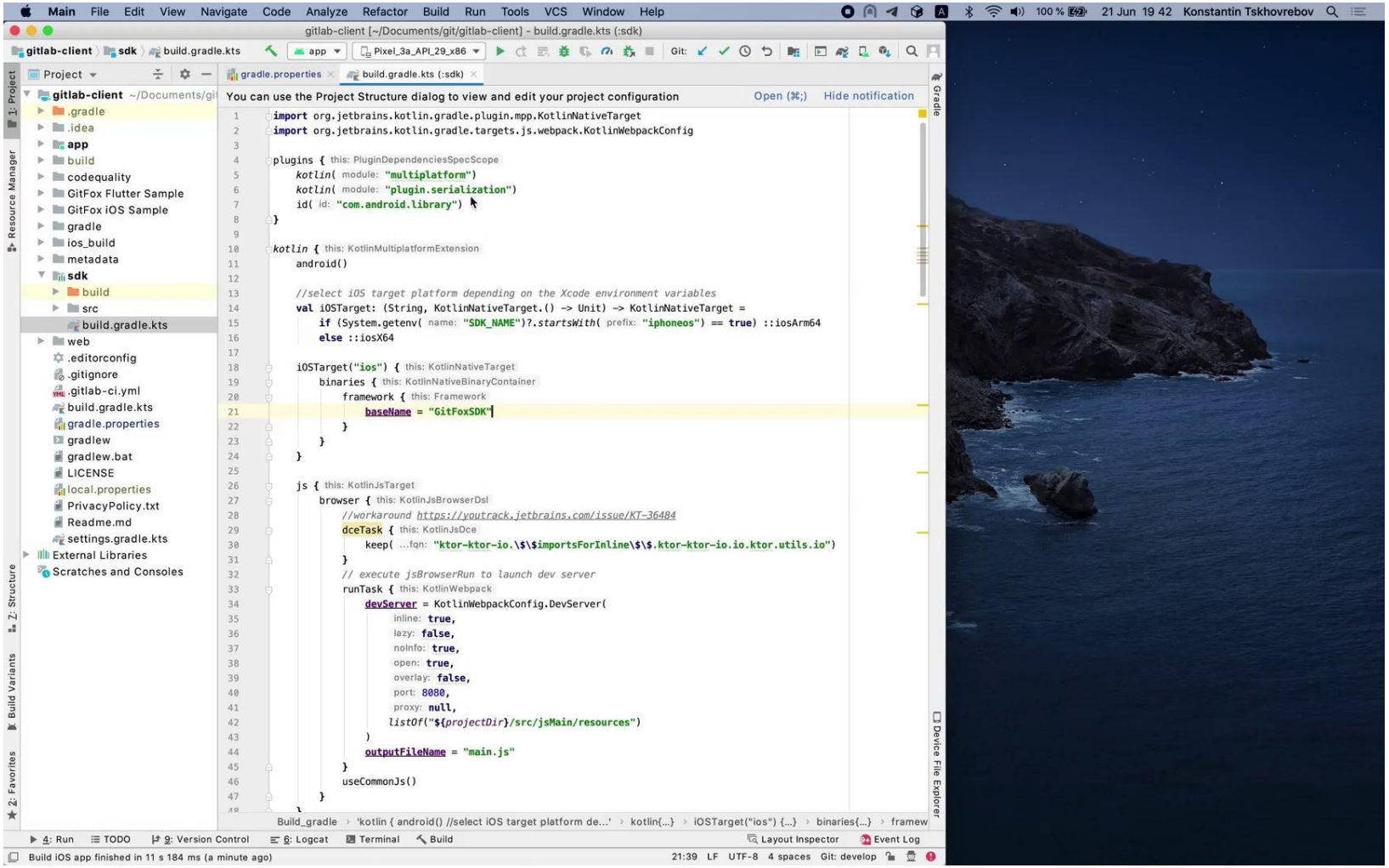


Eugene Saturov

@flutterdevpodcast news



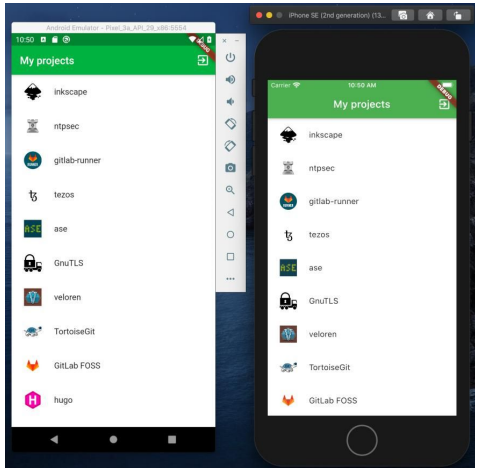
Бонус #2: top secret

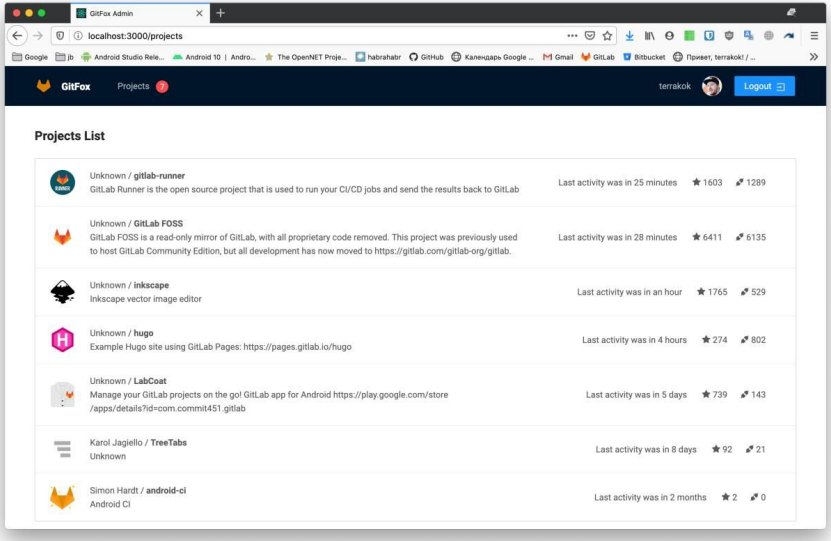

Main File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
21 Jun 19 42 Konstantin Tskhovrebov
gitlab-client [~/Documents/git/gitlab-client] - build.gradle.kts (:sdk)
gitlab-client sdk build.gradle.kts
app Pixel_3a_API_29_x86
Git:
Project
gradle.properties
build.gradle.kts (:sdk)
You can use the Project Structure dialog to view and edit your project configuration
Open (⌘;) Hide notification
gitlab-client ~/Documents/git
.gradle
.idea
app
build
codequality
GitFox Flutter Sample
GitFox iOS Sample
gradle
ios_build
metadata
sdk
build
src
build.gradle.kts
web
.editorconfig
.gitignore
.gitlab-ci.yml
build.gradle.kts
gradle.properties
gradlew
gradlew.bat
LICENSE
local.properties
PrivacyPolicy.txt
Readme.md
settings.gradle.kts
External Libraries
Scratches and Consoles
import org.jetbrains.kotlin.gradle.plugin.mpp.KotlinNativeTarget
import org.jetbrains.kotlin.gradle.targets.js.webpack.KotlinWebpackConfig
plugins { this: PluginDependenciesSpecScope
kotlin(module: "multiplatform")
kotlin(module: "plugin.serialization")
id(id: "com.android.library")
}
kotlin { this: KotlinMultiplatformExtension
android()
//select iOS target platform depending on the Xcode environment variables
val iOSTarget: (String, KotlinNativeTarget.() -> Unit) -> KotlinNativeTarget =
if (System.getenv(name: "SDK_NAME")?.startsWith(prefix: "iphoneos") == true) ::iosArm64
else ::iosX64
iOSTarget("ios") { this: KotlinNativeTarget
binaries { this: KotlinNativeBinaryContainer
framework { this: Framework
baseName = "GitFoxSDK"
}
}
}
js { this: KotlinJsTarget
browser { this: KotlinJsBrowserDsl
//workaround https://youtrack.jetbrains.com/issue/KT-36484
dceTask { this: KotlinJsDce
keep(...fqn: "ktor-ktor-io.$$importsForInline$$.ktor-ktor-io.io.ktor.utils.io")
}
// execute jsBrowserRun to launch dev server
runTask { this: KotlinWebpack
devServer = KotlinWebpackConfig.DevServer(
inline: true,
lazy: false,
noInfo: true,
open: true,
overlay: false,
port: 8080,
proxy: null,
listOf("${projectDir}/src/jsMain/resources")
)
outputFileName = "main.js"
}
useCommonJs()
}
Build_gradle › 'kotlin { android() //select iOS target platform de...' › kotlin{...} › iOSTarget("ios") {...} › binaries{...} › framew
4: Run TODO 9: Version Control 6: Logcat Terminal Build Layout Inspector Event Log
Build iOS app finished in 11 s 184 ms (a minute ago)
21:39 LF UTF-8 4 spaces Git: develop
Gradle
Device File Explorer
1: Project
Resource Manager
7: Structure
Build Variants
2: Favorites

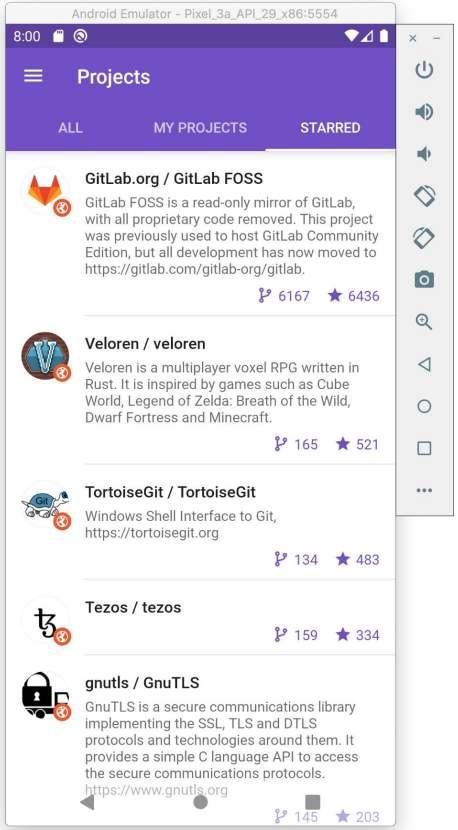
спасибо!

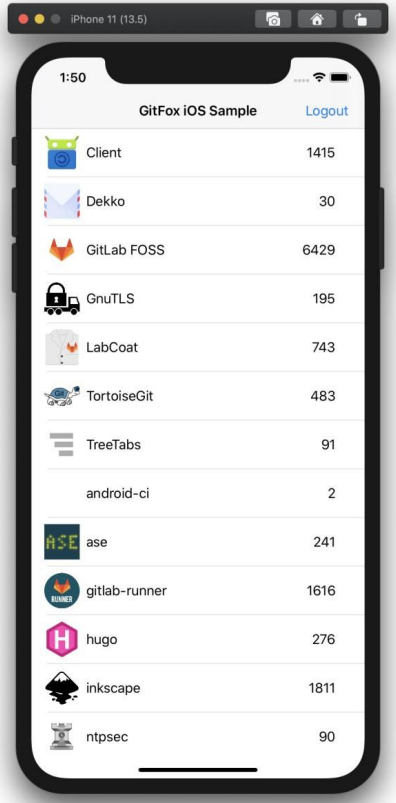
https://t.me/terrakok

https://t.me/gitfox


My projects
inkscape
ntpsec
gitlab-runner
tezos
ase
GnuTLS
veloren
TortoiseGit
GitLab FOSS
hugo


GitFox Admin
localhost:3000/projects
GitFox
Projects
terrakok
Logout
Projects List
Unknown / gitlab-runner
GitLab Runner is the open source project that is used to run your CI/CD jobs and send the results back to GitLab
Last activity was in 25 minutes
Unknown / GitLab FOSS
GitLab FOSS is a read-only mirror of GitLab, with all proprietary code removed. This project was previously used to host GitLab Community Edition, but all development has now moved to https://gitlab.com/gitlab-org/gitlab.
Last activity was in 28 minutes
Unknown / inkscape
Inkscape vector image editor
Last activity was in an hour
Unknown / hugo
Example Hugo site using GitLab Pages: https://pages.gitlab.io/hugo
Last activity was in 4 hours
Unknown / LabCoat
Manage your GitLab projects on the go! GitLab app for Android https://play.google.com/store /apps/details?id=com.commit451.gitlab
Last activity was in 5 days
Karol Jagiello / TreeTabs
Unknown
Last activity was in 8 days
Simon Hardt / android-ci
Android CI
Last activity was in 2 months


Android Emulator - Pixel_3a_API_29_x86:5554
8:00
Projects
ALL
MY PROJECTS
STARRED
GitLab.org / GitLab FOSS
GitLab FOSS is a read-only mirror of GitLab, with all proprietary code removed. This project was previously used to host GitLab Community Edition, but all development has now moved to https://gitlab.com/gitlab-org/gitlab.
6167
6436
Veloren / veloren
Veloren is a multiplayer voxel RPG written in Rust. It is inspired by games such as Cube World, Legend of Zelda: Breath of the Wild, Dwarf Fortress and Minecraft.
165
521
TortoiseGit / TortoiseGit
Windows Shell Interface to Git, https://tortoisegit.org
134
483
Tezos / tezos
159
334
gnutls / GnuTLS
GnuTLS is a secure communications library implementing the SSL, TLS and DTLS protocols and technologies around them. It provides a simple C language API to access the secure communications protocols.
https://www.gnutls.org
145
203


iPhone 11 (13.5)
1:50
GitFox iOS Sample
Logout
Client 1415
Dekko 30
GitLab FOSS 6429
GnuTLS 195
LabCoat 743
TortoiseGit 483
TreeTabs 91
android-ci 2
ase 241
gitlab-runner 1616
hugo 276
inkscape 1811
ntpsec 90