IMMOBILIEN SCOUT 24

RxJava & Coroutines: A Practical Analysis

ashdavies.io - @askashdavies

IMMOBILIEN SCOUT 24

IMMOBILIEN SCOUT 24

IMMOBILIEN SCOUT 24

Adoption

IMMOBILIEN SCOUT 24

7.7% (2015)

IMMOBILIEN SCOUT 24

19.5% (2016)

7.7% (2015)

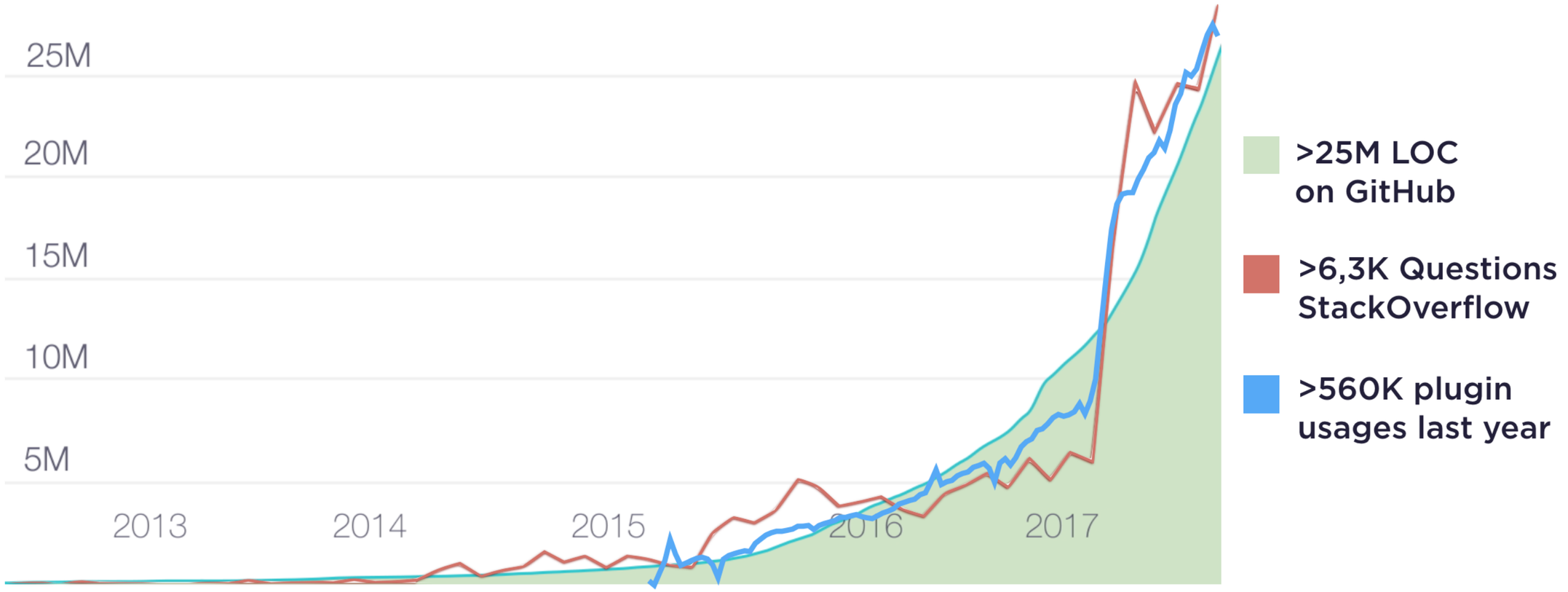
IMMOBILIEN SCOUT 24

46.8% (2017)

19.5% (2016)

7.7% (2015)

IMMOBILIEN SCOUT 24



blog.jetbrains.com/kotlin/2016/02/kotlin-1-0-released-pragmatic-language-for-jvm-and-android/

Multi Platform

Coroutines

IMMOBILIEN SCOUT 24

```
fun main() {
    GlobalScope.launch {
        delay(1000L)
        println("World!")
    }
    println("Hello,")
    Thread.sleep(2000L)
}
```

IMMOBILIEN SCOUT 24

```
fun main() {
    GlobalScope.launch {
        delay(1000L)
        println("World!")
    }
    println("Hello,")
    Thread.sleep(2000L)
}
```

IMMOBILIEN SCOUT 24

```
fun main() {
    GlobalScope.launch {
        delay(1000L)
        println("World!")
    }
    println("Hello,")
    Thread.sleep(2000L)
}
```

IMMOBILIEN SCOUT 24

Stability

IMMOBILIEN SCOUT 24

async / launch

```
val deferred: Deferred<String> = async { "Hello" }
```

```
val result: String = deferred.await()
```

```
val job: Job = launch { "Hello" }
```

```
job.join()
```

IMMOBILIEN SCOUT 24

Annotations

bit.ly/2BrxgKv

IMMOBILIEN SCOUT 24

Here be dragons

IMMOBILIEN SCOUT 24

⚠️ `@ExperimentalCoroutinesApi`

IMMOBILIEN SCOUT 24

⚠️ @ObsoleteCoroutinesApi

IMMOBILIEN SCOUT 24

`@InternalCoroutinesApi`

IMMOBILIEN SCOUT 24

elizarov commented on 28 Sep

Collaborator

From the point of usage `ObsoleteCoroutinesApi` is just like experimental. The different is in intent. When API is experimental it might change or might graduate as is. For obsolete API we already know that there are problems with designs that will force us to deprecated this API in the future when we have a better replacement.

Right now there is no replacement for channel operations, so you have no choice but continue using them. However, we plan to replace them with mechanism based on lazy/cold streams (#254) in the future, which will get you much better performance.

👍 1

Coroutines

 Best thing since sliced bread

IMMOBILIEN SCOUT 24

💪 Coroutines

IMMOBILIEN SCOUT 24

Native first-party library

IMMOBILIEN SCOUT 24

😊 Easy-to-use

IMMOBILIEN SCOUT 24

👌 `suspend fun`

IMMOBILIEN SCOUT 24

Dispatchers.Main

 History of Android

IMMOBILIEN SCOUT 24

Background Processes

IMMOBILIEN SCOUT 24

🏛 Runnable / Handler

IMMOBILIEN SCOUT 24

 AsyncTask

IntentService

IMMOBILIEN SCOUT 24

 Loader<T>

IMMOBILIEN SCOUT 24

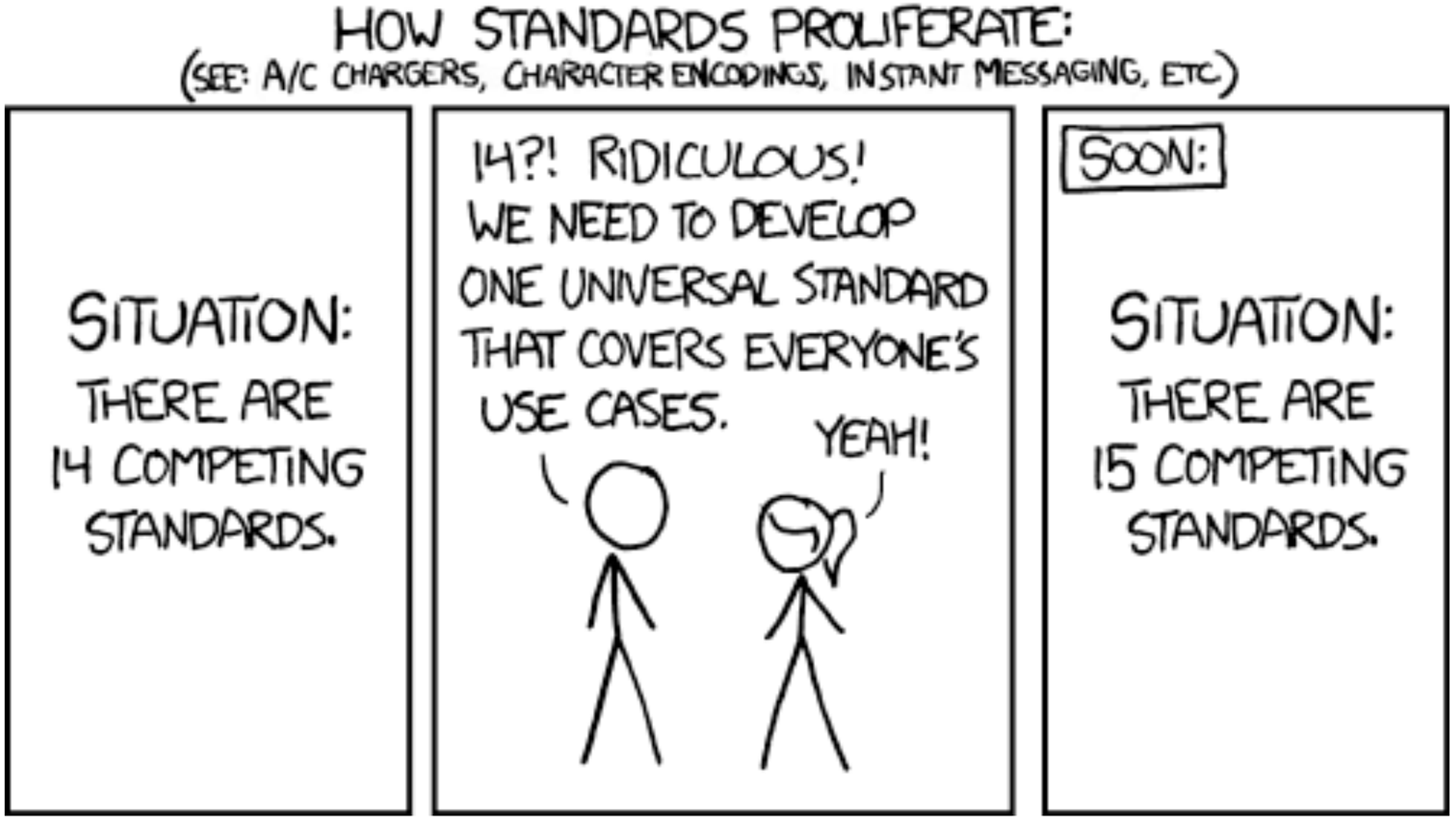
WorkManager

IMMOBILIEN SCOUT 24

Background Processes

`AlarmManager`, `AsyncTask`, `CountDownTimer`, `FutureTask<T>`, `GcmNetworkManager`, `Handler`, `HandlerThread`, `IntentService`, `JobDispatcher`, `JobScheduler`, `Loader<T>`, `ScheduledThreadPoolExecutor`, `Timer`, `Task<T>`, `WorkManager`

IMMOBILIEN SCOUT 24



https://xkcd.com/927/

RxJava to the rescue

IMMOBILIEN SCOUT 24

⛓️ Chained operations

IMMOBILIEN SCOUT 24

⬆️⬇️ Abstracted threading

IMMOBILIEN SCOUT 24

😮 Reactive programming

IMMOBILIEN SCOUT 24

😮

IMMOBILIEN SCOUT 24

RxFingerprint: Android Fingerprint Authentication and Encryption in RxJava2

RxFingerprint wraps the Android Fingerprint APIs (introduced in Android Marshmallow) and makes it easy to:

RxTuples2

RxTuples2 is a library to smooth RxJava2 usage by adding simple Tuple creation functions.

For the RxJava 1.X version, please go to RxTuples.

RxMocks Apache License 2.0

Mocks/Assertions for RxJava testing

RxPermissions

JitPack 0.10.2 | nuget v0.10.2 ▾ 1.2k | build passing

This library allows the usage of RxJava with the new Android M permission model.

RxLoader

License Apache 2.0 | API 16+ | JCenter 2.1.0

An Android Loader that wraps an RxJava Observable.

RxAndroidBle build passing | maven-central v1.7.1

IMMOBILIEN SCOUT 24

RxAnimations

Download 0.9.1 API 15+

RxAnimations is a library with the main goal to make android animations more solid and cohesive.

RxFingerprint: Android Fingerprint Authentication and Encryption in RxJava2

RxFingerprint wraps the Android Fingerprint APIs (introduced in Android Marshmallow) and makes it easy to:

RxTuples2

RxTuples2 is a library to smooth RxJava2 usage by adding simple Tuple creation functions.

For the RxJava 1.X version, please go to RxTuples.

Rx Preferences

Reactive `SharedPreferences` for Android.

Android Arsenal RxPaparazzo

RxJava extension for Android to take photos using the camera, select files or photos from the device and optionally crop or rotate any selected images.

RxMocks Apache License 2.0

Mocks/Assertions for RxJava testing

RxPaparazzo

RxPermissions

JitPack 0.10.2 nuget v0.10.2 1.2k build passing

This library allows the usage of RxJava with the new Android M permission model.

downloads 245/month

Android Arsenal RxCache

RxCache

Android Arsenal RxActivityResult

chat on gitter

> 中文文档

For a more reactive approach go here.

RxBinding

RxJava binding APIs for Android UI widgets from the platform and support libraries.

RxActivityResult

RxLoader

License Apache 2.0 API 16+ JCenter 2.1.0

An Android Loader that wraps an RxJava Observable.

RxAndroidBle build passing maven-central v1.7.1

RxLifecycle

This library allows one to automatically complete sequences based on a second lifecycle stream.

This capability is useful in Android, where incomplete subscriptions can cause memory leaks.

IMMOBILIEN SCOUT 24

RxAnimations

Download 0.9.1 API 15+

RxAnimations is a library with the main goal to make ar

RxPresso

build passing Download 0.2.0 license unknown

Easy Espresso UI testing for Android applications using RxJava.

RxFingerprint: Android Fingerprint Authentication and Encryption in RxJava2

RxTuples2

RxTuples2 is a library to smooth RxJava2 usage by adding simple Tuple creation

For the RxJava 1.X version, please go to RxTuples.

Reactive Wrapper for Java8 WatchService

nakes it easy to:

RxFileWatcher allows you to observe directories (recursively or not) for file system events with a RxJava ot based on the JDK WatchService, but it is much more convenient.

Rx Preferences

Reactive `SharedPreferences` for Android.

Android Arsenal RxPaparazzo

RxJava extension for Android to take photos using the camera, select files or photos from the device and optionally crop or rotate any selected images.

RxMocks

Apache License 2.0

ocks/Assertions fo

Reactive Fit API Library for Android

build passing codecov unknown Download 2.0.1 Android Arsenal RxFit API 9+

RxPaparazzo

RxCupboard

RxCupboard brings the excellent Android Cupboard library into the world of RxJava. Usin `Flowable` s, you can fluently store and retrieve streams of POJOs from your database.

RxPermissions

RxFileObserver

Android Arsenal RxFileObserver

Reactive wrapper around Android's FileObserver

downloads 245/month

ndroid Arsenal RxCache

JitPack 0.10.2 nug

This library allows tl

RxBonjour

RxActivityResult

build failing

A reactive wrapper around network service discovery functionalities for Kotlin and Java.

RxCache

中文文档

For a more reactive approach go here.

RxBinding

RxActivityResult

RxJava binding APIs for Android UI widgets from the platform and support libraries.

RxLoader

License Apache 2.0 API 16+ JCenter 2.1.0

An Android Loader that wraps an RxJava Observable.

RxAndroidBle

build passing maven-central v1.7.1

RxLifecycle

This library allows one to automatically complete sequences based on a second lifecycle stream.

This capability is useful in Android, where incomplete subscriptions can cause memory leaks.

RxAnimations

Download 0.9.1 API 15+

RxAnimations is a library with the main goal to make android anin

RxPresso

build passing Download 0.2.0 license unknown

Easy Espresso UI testing for Android applications using RxJava.

RxFingerprint: Android Fingerprint Authentication and Encryption in RxJava2

RxTuples2

RxTuples2 is a library to smooth RxJava2 usage by adding simple Tuple creation functio

For the RxJava 1.X version, please go to RxTuples.

Reactive Wrapper for Java8 WatchService

nakes it easy to:

RxFileWatcher allows you to observe directories (recursively or not) for file system events with a RxJava ob based on the JDK WatchService, but it is much more convenient.

RxPalette

RxJava binding APIs for the Palette Android library.

Rx Preferences

Reactive `SharedPreferences` for Android.

Android Arsenal RxPaparazzo

RxJava extension for Android to take photos using the camera, select files or photos from the device and optionally crop or rotate any selected images.

RxMocks

Apache License 2.0

RxPaparazzo

RxCupboard

RxCupboard brings the excellent Android Cupboard library into the world of RxJava. Usin `Flowable` s, you can fluently store and retrieve streams of POJOs from your database.

ocks/Assertions fo

Reactive Fit API Library for Android

build passing codecov unknown Download 2.0.1 Android Arsenal RxFit API 9+

RxPermissions

JitPack 0.10.2 nuget v0.10.2

This library allows the usage

RxFileObserver

Android Arsenal RxFileObserver

Reactive wrapper around Android's FileObserver

downloads 245/month

Android Arsenal RxCache

RxBonjou

build failing

Reactive Wearable API Library for Android

build passing codecov unknown Download 1.3.0 Android Arsenal RxWear API 9+

Arsenal RxActivityResult

gitter

A reactive wrapper around network service discovery functionalities for Kotlin and Java.

RxCache

中文文档

For a more reactive approach go here.

RxLoader

License Apache 2.0 API 16+ JCenter 2.1.0

RxBinding

RxJava binding APIs for Android UI widgets from the platform and support libraries.

RxActivityResult

RxRelay

an RxJava Observable.

Relays are RxJava types which are both an `Observable` and a `Consumer` .

Basically: A `Subject` except without

Android RxFirebase

build error coverage unknown bintray invalid license apache 2.0

RxJava wrapper for use with the Android Firebase client

RxAndroidBle

build passing maven

RxLifecycle

Android RxTasks

build passing coverage unknown Download 2.0.0 license apache 2.0

ibrary allows one to automatically complete sequences based on a second lifecycle stream.

capability is useful in Android, where incomplete subscriptions can cause memory leaks.

ashdavies / **rx-tasks**

IMMOBILIEN SCOUT 24

IMMOBILIEN SCOUT 24

👍 Asynchronous API's

IMMOBILIEN SCOUT 24

👎 Synchronous API's

IMMOBILIEN SCOUT 24

```
Observable.fromIterable()
```

IMMOBILIEN SCOUT 24

Flowable / Observable / Single / Completable / Maybe

IMMOBILIEN SCOUT 24

From \ To	Flowable	Observable	Maybe	Single	Completable
Flowable		toObservable()	reduce() elementAt() firstElement() lastElement() singleElement()	scan() elementAt() first()/firstOrError() last()/lastOrError() single/singleOrError() all()/any()/count() (and more)	ignoreElements()
Observable	toFlowable()		reduce() elementAt() firstElement() lastElement() singleElement()	scan() elementAt() first()/firstOrError() last()/lastOrError() single/singleOrError() all()/any()/count() (and more)	ignoreElements()
Maybe	toFlowable()	toObservable()		toSingle() sequenceEqual()	toCompletable()
Single	toFlowable()	toObservable()	toMaybe()		toCompletable()
Completable	toFlowable()	toObservable()	toMaybe()	toSingle() toSingleDefault()	

IMMOBILIEN SCOUT 24

```
Observable
  .fromIterable(resourceDraft.getResources())
  .flatMap(resourceServiceApiClient::createUploadContainer)
  .zipWith(Observable.fromIterable(resourceDraft.getResources()), Pair::create)
  .flatMap(uploadResources())
  .toList()
  .toObservable()
  .flatMapMaybe(resourceCache.getResourceCachedItem())
  .defaultIfEmpty(Resource.getDefaultItem())
  .flatMap(postResource(resourceId, resourceDraft.getText(), currentUser, resourceDraft.getIntent()))
  .observeOn(AndroidSchedulers.mainThread())
  .subscribeOn(Schedulers.io())
  .subscribe(
      resource -> repository.setResource(resourceId, resource, provisionalResourceId),
      resourceUploadError(resourceId, resourceDraft, provisionalResourceId)
  );
```

IMMOBILIEN SCOUT 24


RON BURGUNDY
IS #1 IN SAN DIEGO
IMMOBILIEN
SCOUT 24

Hidden complexity

IMMOBILIEN SCOUT 24

😭 Hidden gotchas

IMMOBILIEN SCOUT 24

💾 Memory footprint

IMMOBILIEN SCOUT 24

 Steep learning curve

IMMOBILIEN SCOUT 24

http://explosm.net/comics/3185/


IMMOBILIEN
SCOUT 24

#RxMustDie

pca.st/7IJG

"When all you have is a hammer, everything looks like a nail"

Ivan Morgillo (@hamen)

IMMOBILIEN SCOUT 24

Reactive & Imperative programming

IMMOBILIEN SCOUT 24

Coroutine Use Cases

IMMOBILIEN SCOUT 24

Network Call Handling

```
// RxJava2
fun getUser(): Single<User> = Single.fromCallable { /* ... */ }

// Coroutine
suspend fun getUser(): User = /* ... */
```

IMMOBILIEN SCOUT 24

Cache Retrieval

```
// RxJava2
fun getUser(): Maybe<User> = Maybe.fromCallable { /* ... */ }

// Coroutine
suspend fun getUser(): User? = /* ... */
```

IMMOBILIEN SCOUT 24

Background Task Handling

```
// RxJava2
fun storeUser(user: User): Completable.fromCallable { /* ... */ }

// Coroutine
suspend fun storeUser(user: User) { /* ... */ }
```

IMMOBILIEN SCOUT 24

RxJava2 / Coroutines

Single<T> / T

Maybe<T> / T?

Completable / Unit

IMMOBILIEN SCOUT 24

Thread Handling

```
// RxJava2
getUser()
  .subscribeOn(Schedulers.computation())
  .observeOn(AndroidSchedulers.mainThread())
  .subscribe { /* Do something */ }

// Coroutine
launch(Dispatchers.Main) {
  val user = getUser()
  /* Do something */
}
```

IMMOBILIEN SCOUT 24

💪 ~~FlatMap~~

```
// RxJava2
getUser()
  .flatMap { doSomethingWithUser(it) }
  .subscribeOn(Schedulers.computation())
  .observeOn(AndroidSchedulers.mainThread())
  .subscribe { /* Do something else */ }

// Coroutine
launch(Dispatchers.Main) {
  val user = getUser()
  val smth = doSomethingWithUser(user)

  /* Do something else */
}
```

IMMOBILIEN SCOUT 24

Callback Consumption

```
// RxJava2
fun getSingle(): Single = Single.create<T> { emitter ->
  doSomethingAsync(object: Callback<T> {
    override fun onComplete(result: T) = emitter.onSuccess(result)
    override fun onException(exception: Exception) = emitter.onError(exception)
  })
}

// Coroutine
suspend fun getCoroutine() : T = suspendCoroutine { continuation ->
  doSomethingAsync(object : Callback<Baz> {
    override fun onComplete(result: T) = continuation.resume(result)
    override fun onException(exception: Exception) = continuation.resumeWithException(exception)
  })
}
```

IMMOBILIEN SCOUT 24

Task Cancellation

```
// RxJava2
val disposable: Disposable = Single.create { /* Do something */ }
    .subscribe { /* Do something else */ }
disposable.dispose()

// Coroutine
val parent: Job = Job()
launch(Dispatchers.Main + parent) { /* Do something */ }
parent.cancelChildren()
```

IMMOBILIEN SCOUT 24

Lifecycle Task Cancellation

bit.ly/2POmNBJ

IMMOBILIEN SCOUT 24

Lifecycle Task Cancellation

```
class MainActivity : AppCompatActivity {

  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)

    lifecycle.coroutineScope.launch {
      someSuspendFunction()
      someOtherSuspendFunction()
      someCancellableSuspendFunction()
    }
  }
}
```

IMMOBILIEN SCOUT 24

Value Streams

```
// RxJava2
val publisher = PublishSubject()
publisher.subscribe { /* Do something */ }
publisher.onNext("Hello")

// Coroutine
val channel = Channel<String>()
launch { channel.send("Hello") }
launch { channel.consumeEach { /* Do something */ } }
```

IMMOBILIEN SCOUT 24

💪 Backpressure

IMMOBILIEN SCOUT 24

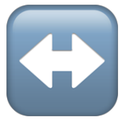
~~Channels~~

bit.ly/2DQU7lb

IMMOBILIEN SCOUT 24

RxJava Use Cases

IMMOBILIEN SCOUT 24



Channels

IMMOBILIEN SCOUT 24

Cold Streams

IMMOBILIEN SCOUT 24

Complex Business Logic

IMMOBILIEN SCOUT 24

Coroutines & RxJava: An Asynchronicity Comparison

Manuel Vivo (@manuelvicnt)

bit.ly/2R27stP

IMMOBILIEN SCOUT 24

Is Coroutines a replacement for RxJava?

IMMOBILIEN SCOUT 24

Maybe...

IMMOBILIEN SCOUT 24

Should I migrate to Coroutines?

IMMOBILIEN SCOUT 24

Probably not...

IMMOBILIEN SCOUT 24

"Don't fix what ain't broke"

Some Guy

IMMOBILIEN SCOUT 24


IMMOBILIEN
SCOUT 24

Did "I" migrate to Coroutines?

IMMOBILIEN SCOUT 24

 Yes!

IMMOBILIEN SCOUT 24

How could I migrate to Coroutines?

IMMOBILIEN SCOUT 24

🤝 Migration Policy

IMMOBILIEN SCOUT 24

Retrofit Services

IMMOBILIEN SCOUT 24

Retrofit2 Coroutines Adapter bit.ly/2TyGXOh

`com.jakewharton.retrofit:retrofit2-kotlin-coroutines-adapter:+`

IMMOBILIEN SCOUT 24

```
interface UserService {

  @GET("/user")
  fun getUser(): Deferred<User>
}

val retrofit = Retrofit.Builder()
    .baseUrl("https://example.com/")
    .addCallAdapterFactory(CoroutineCallAdapterFactory())
    .build()

GlobalScope.launch {
    val user = retrofit
        .create<UserService>() // >= 2.5.0
        .getUser()
        .await()
}
```

IMMOBILIEN SCOUT 24

```
interface UserService {

  @GET("/user")
  fun getUser(): Deferred<User>
}

val retrofit = Retrofit.Builder()
    .baseUrl("https://example.com/")
    .addCallAdapterFactory(CoroutineCallAdapterFactory())
    .build()

GlobalScope.launch {
    val user = retrofit
        .create<UserService>() // >= 2.5.0
        .getUser()
        .await()
}
```

IMMOBILIEN SCOUT 24

```
interface UserService {

  @GET("/user")
  fun getUser(): Deferred<User>
}

val retrofit = Retrofit.Builder()
    .baseUrl("https://example.com/")
    .addCallAdapterFactory(CoroutineCallAdapterFactory())
    .build()

GlobalScope.launch {
    val user = retrofit
        .create<UserService>() // >= 2.5.0
        .getUser()
        .await()
}
```

IMMOBILIEN SCOUT 24

```
interface UserService {

  @GET("/user")
  fun getUser(): Deferred<User>
}

val retrofit = Retrofit.Builder()
    .baseUrl("https://example.com/")
    .addCallAdapterFactory(CoroutineCallAdapterFactory())
    .build()

GlobalScope.launch {
    val user = retrofit
        .create<UserService>() // >= 2.5.0
        .getUser()
        .await()
}
```

IMMOBILIEN SCOUT 24

Coroutines RxJava2 bit.ly/2DQ2ZYn

`org.jetbrains.kotlinx:kotlinx-coroutines-rx2:+`

IMMOBILIEN SCOUT 24

Name	Result	Scope	Description
[rxCompletable]	`Completable`	[CoroutineScope]	Cold completable that starts coroutine on subscribe
[rxMaybe]	`Maybe`	[CoroutineScope]	Cold maybe that starts coroutine on subscribe
[rxSingle]	`Single`	[CoroutineScope]	Cold single that starts coroutine on subscribe
[rxObservable]	`Observable`	[ProducerScope]	Cold observable that starts coroutine on subscribe
[rxFlowable]	`Flowable`	[ProducerScope]	Cold observable that starts coroutine on subscribe with **backpressure** support

```
val service: UserService = /* ... */

GlobalScope.rxSingle { service.getUser() }
    .observeOn(AndroidSchedulers.mainThread())
    .subscribeOn(Schedulers.io())
    .subscribe(
        { /* Do something with user */ },
        { /* Handle error ... maybe */ }
    )
```

IMMOBILIEN SCOUT 24

😅 Conclusion

IMMOBILIEN SCOUT 24

THE END

IMMOBILIEN SCOUT 24

Cheers! 


ITS OVER. GO HOME!