

Maximize Android Mobility with
Kotlin and Reactive Paradigm





Muh Isfhani Ghiath

@isfaaghyth / isfa@daeng.id

Chief Warrior, **Daeng.id**
Community Lead, **Developer Student Clubs**
Software Engineer Intern, **Tokopedia**
ROC (Representative), **Google Crowdsource**
Github Campus Expert (GCE), **Github**

Find me at:
@isfaaghyth

iyth (Muh Isfhani Ghiath) x +

GitHub, Inc. [US] | https://github.com/isfaaghyth

Muh Isfhani Ghiath
isfaaghyth

creative enthusiasm and polyglot developer.

[Edit bio](#)

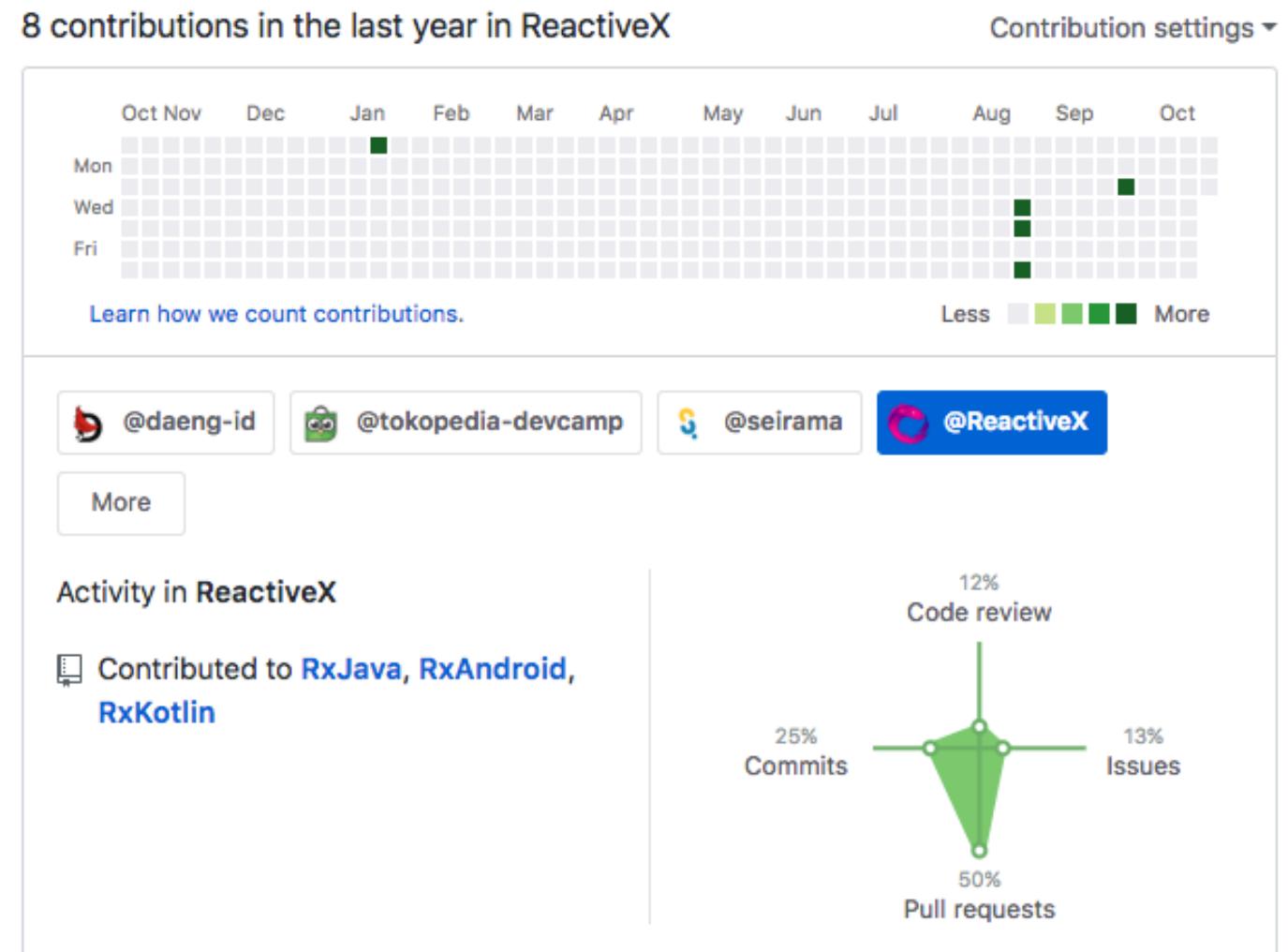
Developer Program Member

@daeng-id
Depok, Jawa Barat, Indonesia
✉️ isfaaghyth@gmail.com
🔗 http://isfaa.me/

Organizations

1,229 contributions in the last year [Contribution settings](#)

My contribution in Rx Project **RxJava, RxAndroid, RxKotlin.**



Find me at:
@isfaaghyth

isfaaghyth | Git Awards

Not Secure | git-awards.com/users/isfaaghyth

Git Awards GitHub username Search Top users by city Top users by country Top users worldwide About



java ranking Depok 3 / 55 🏆
Indonesia 13 / 2 031 🏆
Worldwide 5 192 / 745 015 🏆
Repos : 94 📂
Stars : 243 ★

isfaaghyth
Depok kotlin ranking Depok 1 / 5 🏆
Indonesia 6 / 127 🏆
Worldwide 1 041 / 12 904 🏆
Repos : 8 📂
Stars : 24 ★

This project is hosted by exoscale.ch thanks for their support !

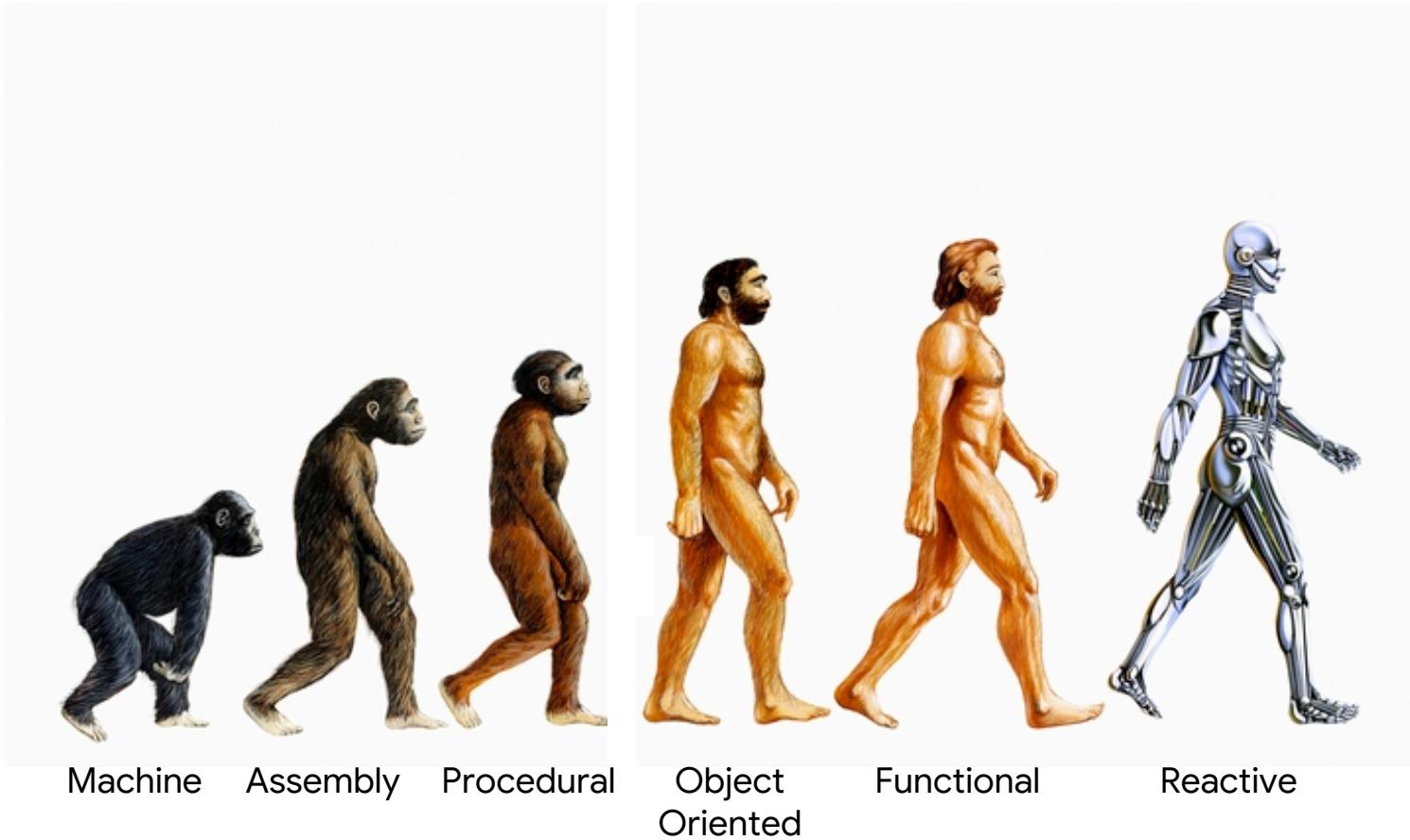
javascript ranking Depok 2 / 46 🏆
Indonesia 225 / 3 061 🏆



/rē'aktiv/



/rē'ak ney-tiv/



Reactive Programming

In computing, reactive programming is a declarative programming paradigm concerned with data streams and the propagation of change.

W



CHAPTER SIX

the possible criminal connection of the heavy men eventually taken her away. These people knew everyone and were not slow in discussing it. There was no reserve, self-preservation.

Rosemary had talked about it being natural that people assume she was gay since she was single and had a sister who was already 'out' with a partner who was a lawyer. Gertie had spoken about her husband's problems coping with drink and violence spoke as if Jack had been prone to getting chest colds in the winter. Colm had approached their table with a casual apology over incident as if it had not been the most excruciatingly embarrassing moment of her life. The two women had told her how they initially thought Ria was mad to go to America and leave children but they hoped it would all work out for the best.

Marilyn could not take in the degree of involvement and indiscreetness that these people felt confident to have in everyone else's life. They thought nothing of discussing the motives and private sorrows of their friend with Marilyn who was after all a complete stranger, here purely because of an accidental home exchange. While she felt sympathy for Ria and all that had happened to her, she also felt a sense of annoyance.

Why had she not kept her dignity, and refused to allow all these people into her life? The only way to cope with tragedy and grief was to refuse to permit it to be articulated and acknowledged. Deny its existence and you had some hope of survival. Marilyn got out of bed and looked down on the messy garden and the other large red brick houses of the neighbourhood. She felt very lost and alone in this place where garrulous people wanted to know everything about you and expected you to need the details of their lives too!

She ached for the cool house and beautiful garden in Westville. If she were there now she could go and swim lengths of her pool safe in the knowledge that no one would call and burden her with post mortems about last night. Clement the cat who slept on her bed every night woke up and stretched and came over to her hopefully. He was purring loudly. The day was about to begin, he was expecting a game and a bowl of something.

Marilyn looked at him sadly. 'I don't usually talk to animals, Clement, but I'm making an exception in your case. I made the wrong decision coming here. It was the worst decision I ever made in my life.'

302

303

'Do you think when we're talking to Gr...

'...call her Noya?' Brian asked.

'What?' Annie looked up from her book.

'You know ... if we call Bernadette's mother b...

'...maybe we should do the same with Granny.' Brian said.

'No, Brian, and shut up,' said Annie.

'You always say shut up, you never say anything at all!'

'Who could say anything nice to you, Brian, ha...

'Well, some people do.'

'Who apart from Mam and Dad? And they ha...

'you're what they got.'

'Finola often says nice things.'

'Tell me one nice thing she said to you today.'

'She said it was good that I had remembered to...

'command the centre of the board.'

'And had you?' Annie still refused chess lessons accept that Brian had mastered it.

'Well, only by accident in a way. I just sort of pu...

'and they were commanding and she was very pi...

'Brian smiled at the triumph of it all.'

'Sometimes he was more pathetic than awful,

'you'd feel sorry for him. And he didn't really unde...

'lives were going to change. He thought that af...

ReactiveX

a library for composing asynchronous and event-based programs by using observables sequences.

Rx, which is a port of the Reactive-Extensions library from .NET, enables Android apps to be built in this 'style'.

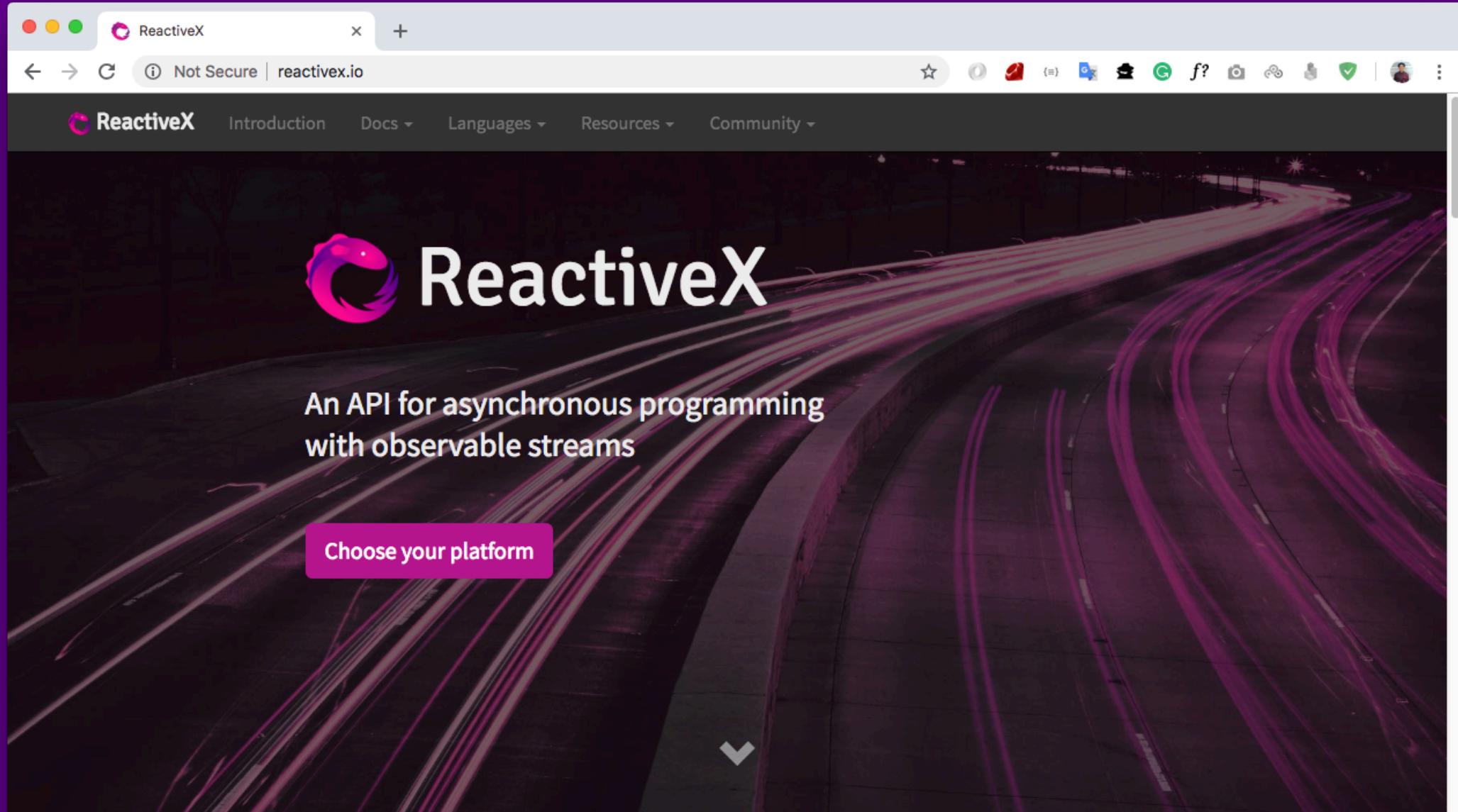


ReactiveX?

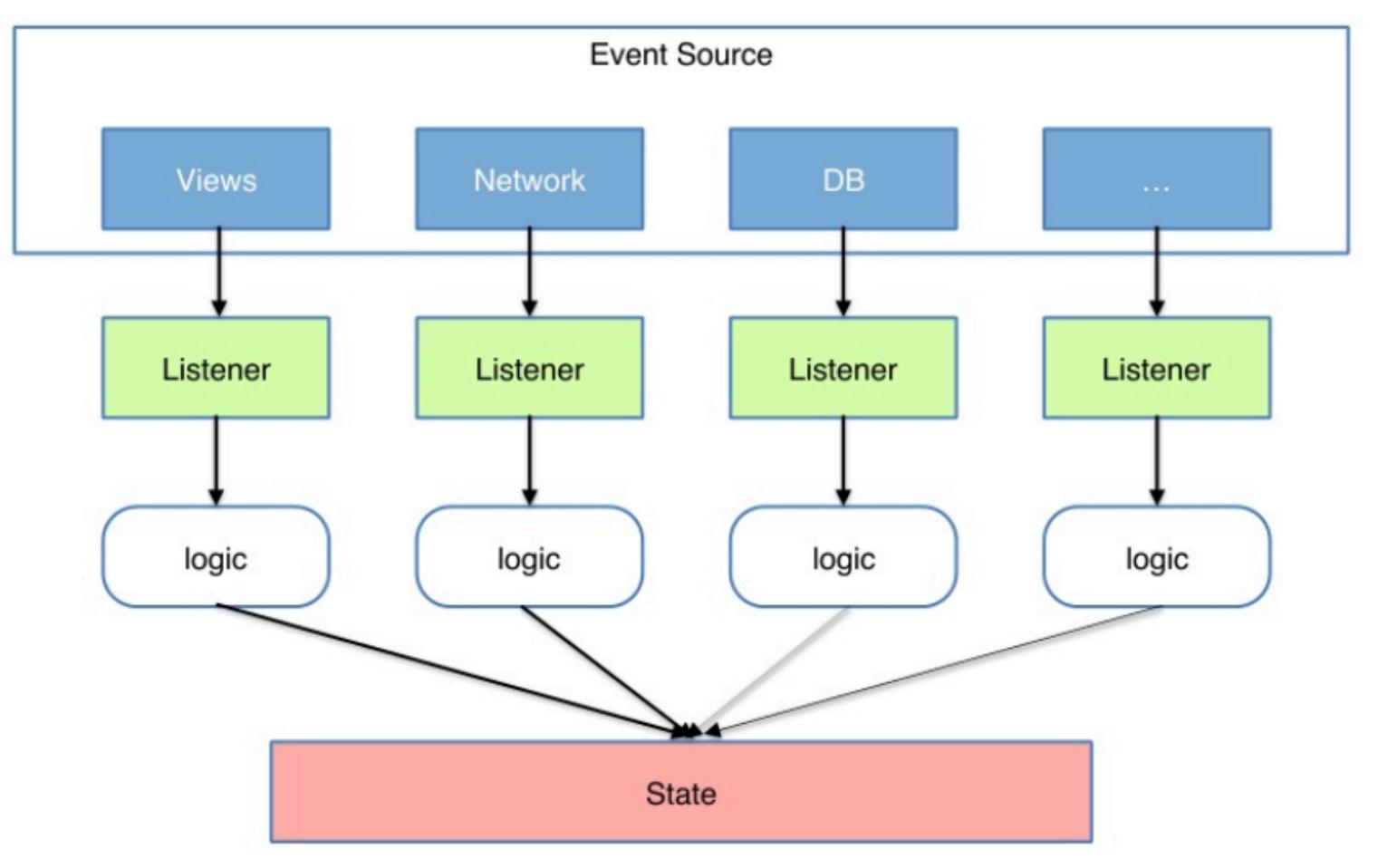
- composable data flow
- push concept
- combination of
 - observer pattern
 - iterator pattern
 - function programming



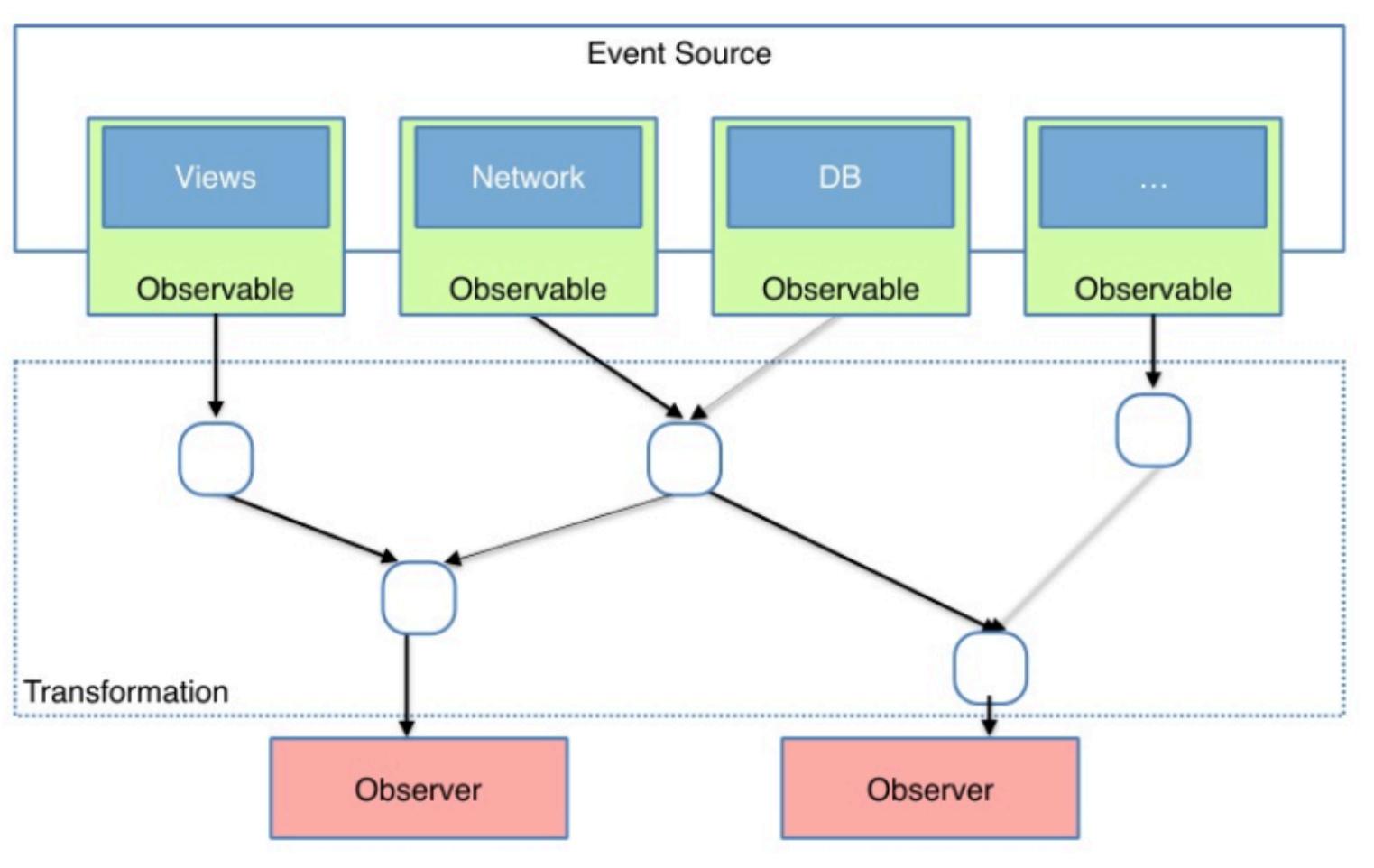
www.reactivex.io

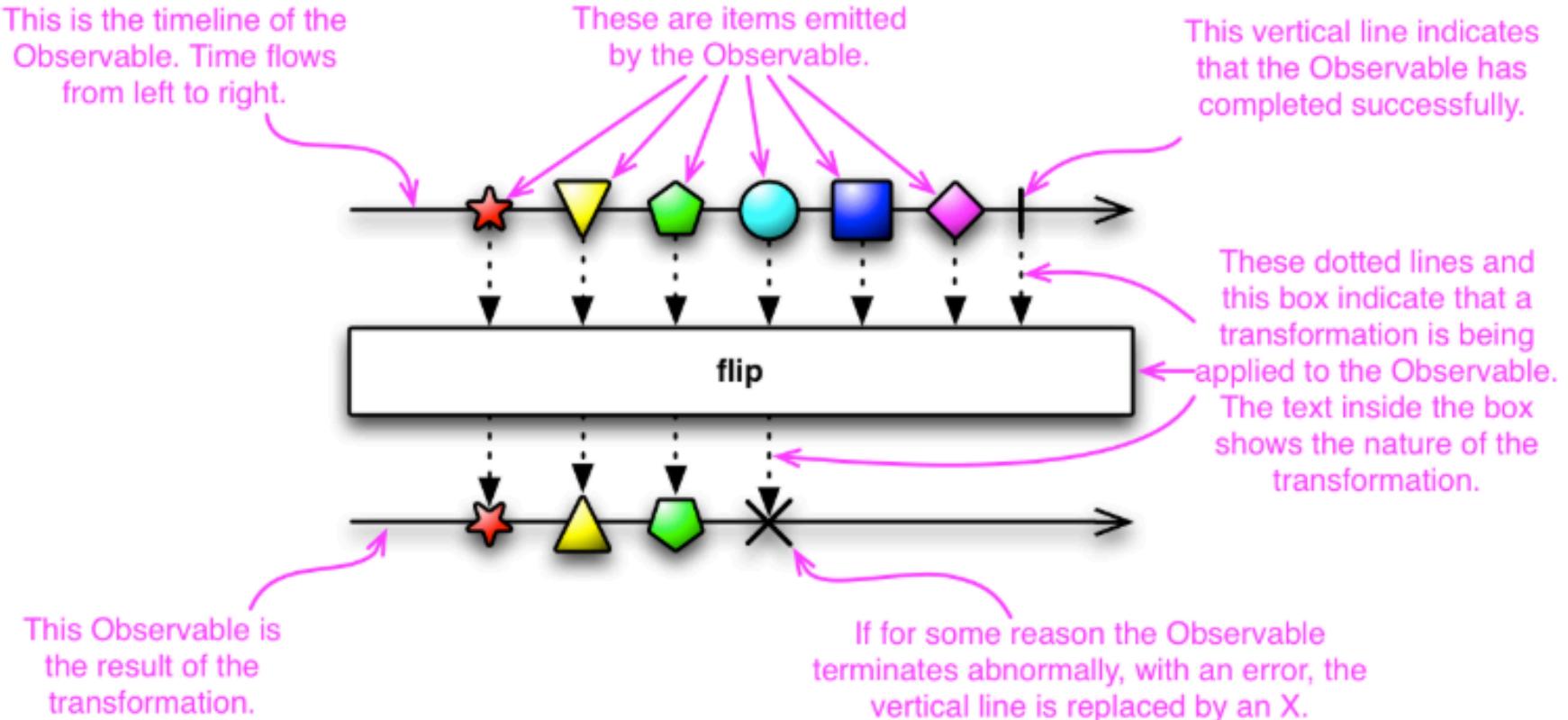


Typical Approach



Reactive Approach





(Java) Simple String Transform with Rx

```
@Test private void test() {
    Observable.just("Isfa", "Ganteng")
        .map(new Func1<String, String>() {
            @Override public String call(String s) {
                return s.toLowerCase();
            }
        })
        .reduce(new Func2<String, String, String>() {
            @Override public String call(String s1, String s2) {
                return s1 + "-" + s2;
            }
        })
        .subscribe(new Consumer<String>() {
            @Override public void call(String result) {
                Log.d("DSC", result);
            }
        });
}
```

(Kt) Simple String Transform with Rx and Functional

```
@Test fun test() {  
    Observable.just("Isfa", "Ganteng")  
        .map { s -> s.toLowerCase() }  
        .reduce { (s1, s2) -> "$s1-$s2" }  
        .subscribe { result -> Log.d("DSC", result) }  
}
```

Data source:
“Isfa” dan “Ganteng”

Transformation

Subscription

Result:
“isfa-ganteng”

KeyPart

- Observable
- Observer or Subscriber
 - onNext(T)
 - onComplete(T)
 - onError(Throwable)
- Subject



What is ReactiveX good for?



Functional

Avoid intricate stateful programs, using clean input/output functions over observable streams.



Less is more

ReactiveX's operators often reduce what was once an elaborate challenge into a few lines of code.



Async error handling

Traditional try/catch is powerless for errors in asynchronous computations, but ReactiveX is equipped with proper mechanisms for handling errors.



Concurrency made easy

Observables and Schedulers in ReactiveX allow the programmer to abstract away low-level threading, synchronization, and concurrency issues.



What is ReactiveX good for?

- Making code simple and readable
- Async Processing
 - (no AsyncTask, AsyncTaskLoader, ...)
- Async Composition
 - RxJava offers simple chaining of async operations
 - Eliminates callback hell



GET /dscindonesia

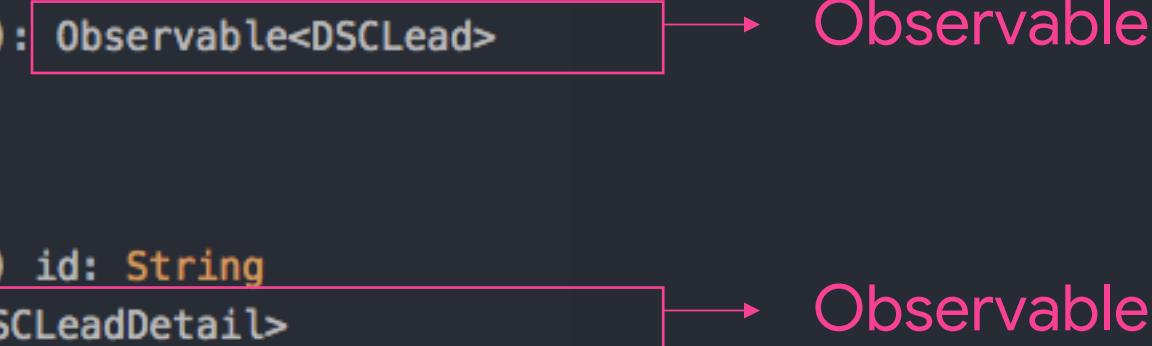
```
{  
  lead: [  
    {  
      name: "M.Isfha",  
      detail: 1278  
    },  
    {...}  
  ]  
}
```

GET /detail?id=

```
{  
  img: "http://gavatar.com/isfha.png",  
  chapter: "STT Nurul Fikri",  
  region: "Jakarta"  
}
```

Chaining Multiple Request with `flatMap`

```
interface Services {  
  
    @GET("/dscindonesia")  
    fun getDSCList(): Observable<DSCLead>  
  
    @GET("/detail")  
    fun getDetail(  
        @Query("id") id: String  
    ): Observable<DSCLeadDetail>  
}
```



The code defines an interface named Services. It contains two methods: getDSCList() which returns an Observable<DSCLead> and getDetail() which takes an id as a query parameter and returns an Observable<DSCLeadDetail>. Two arrows point from the method signatures to the word 'Observable'.

Chaining Multiple Request with `flatMap`

```
@Test fun chaining() {  
    service().getDSCList()  
        .flatMap { res1 -> service().getDetail(res1.detailId) }  
        .SchedulerOn(Schedulers.io())  
        .ObserveOn(AndroidThreads.mainThread())  
        .subscribe()  
}
```

Operator
Schedulers

450+
operators

10+
Supported
Language



My First Library with Kt+Rx

The screenshot shows a GitHub repository page for the user 'isfaaghyth' named 'notify'. The repository has 15 commits, 2 branches, 1 release, and 1 contributor. It uses the RxJava2 library and follows the MIT license.

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Description: a micro-library to simplifies a simple communication between activity, fragment, services, etc. through reactive-pattern.

Topics: notification, rxjava2, communicator, Manage topics

Metrics: 15 commits, 2 branches, 1 release, 1 contributor, MIT

Pubsub easily with Notify()

```
private val subject: Subject<Any> = PublishSubject.create()
```

→ PublishSubject operator

```
fun <T> listen(tClass: Class<T>, subscriber: EventProvider,  
                onNext: Consumer<T>): Disposable {  
    return subject  
        .subscribeOn(subscriber.io())  
        .observeOn(subscriber.mainThread())  
        .filter { o -> o.javaClass == tClass }  
        .map { o -> o as T }  
        .subscribe(onNext)  
}
```

→ Schedulers

→ Subscriber

Keystore library with Rx

The screenshot shows a GitHub repository page for the project 'Rak'. The repository is owned by 'isfaaghyt'. The main navigation bar includes links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. The repository description is 'Data Storage Library for Android.' and it has an 'Edit' button. Below the description, there are buttons for 'database', 'android', 'nosqlite', and 'Manage topics'. Key statistics at the bottom include 15 commits, 2 branches, 1 release, 1 contributor, and an MIT license. A horizontal bar at the bottom is divided into orange and brown segments.

isfaaghyt / Rak

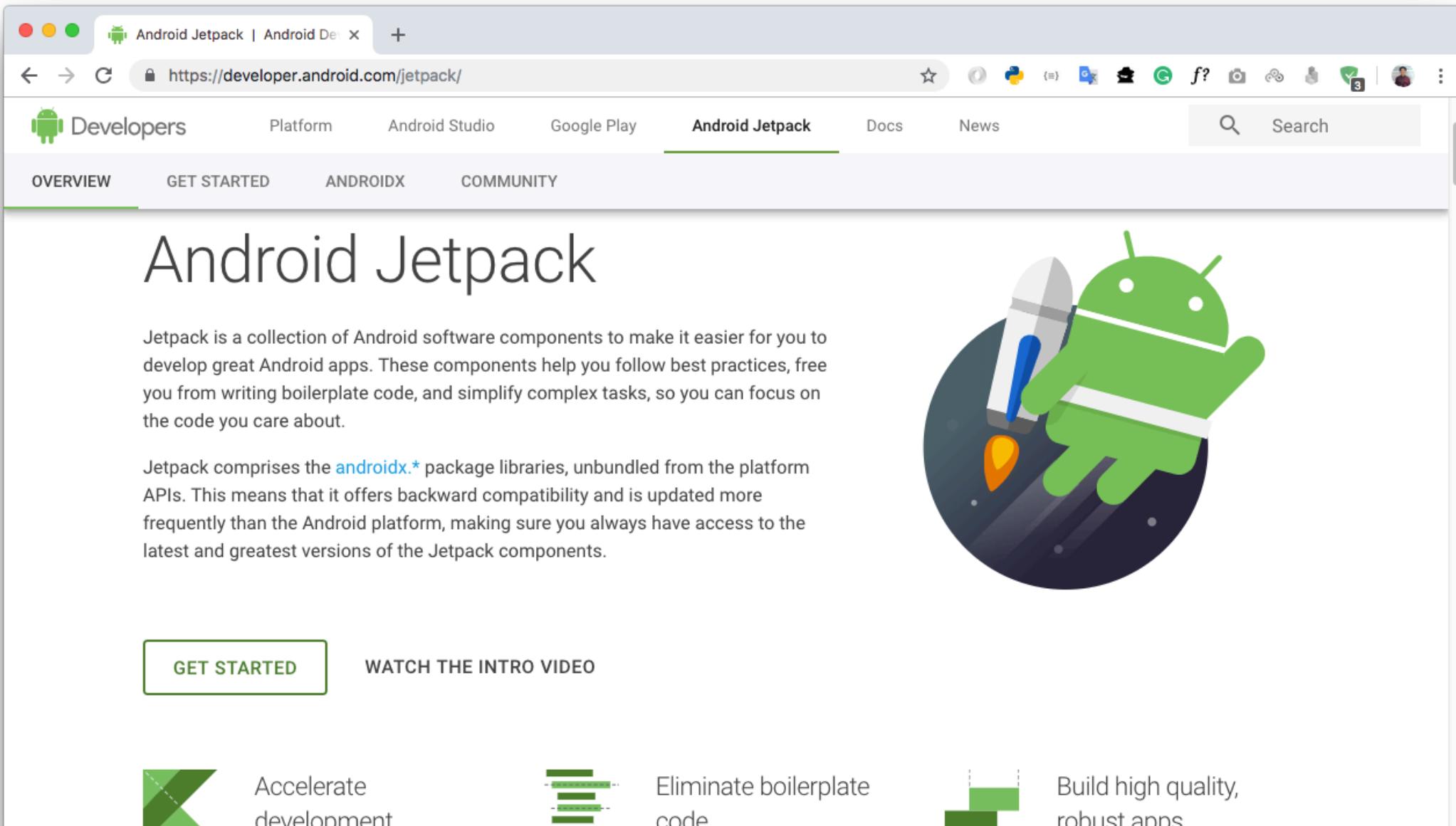
Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Data Storage Library for Android. Edit

database android nosqlite Manage topics

15 commits 2 branches 1 release 1 contributor MIT

Android Architecture Components



The screenshot shows a web browser displaying the official Android Jetpack documentation at <https://developer.android.com/jetpack/>. The page has a light gray header with the Android logo and the text "Android Jetpack | Android De X". Below the header is a navigation bar with links for "Developers", "Platform", "Android Studio", "Google Play", "Android Jetpack" (which is underlined in green), "Docs", and "News". A search bar is also present. The main content area features a large title "Android Jetpack" and a paragraph describing Jetpack as a collection of components to simplify app development. To the right of the text is a large, stylized green Android character wearing a white spacesuit and holding a white rocket. At the bottom of the page are two buttons: "GET STARTED" and "WATCH THE INTRO VIDEO". Below these buttons are three cards with icons and text: "Accelerate development" (with a green Kotlin icon), "Eliminate boilerplate code" (with a green code icon), and "Build high quality, robust apps" (with a green square icon).

Android Jetpack | Android De X

https://developer.android.com/jetpack/

Developers Platform Android Studio Google Play **Android Jetpack** Docs News

Search

OVERVIEW GET STARTED ANDROIDX COMMUNITY

Android Jetpack

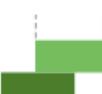
Jetpack is a collection of Android software components to make it easier for you to develop great Android apps. These components help you follow best practices, free you from writing boilerplate code, and simplify complex tasks, so you can focus on the code you care about.

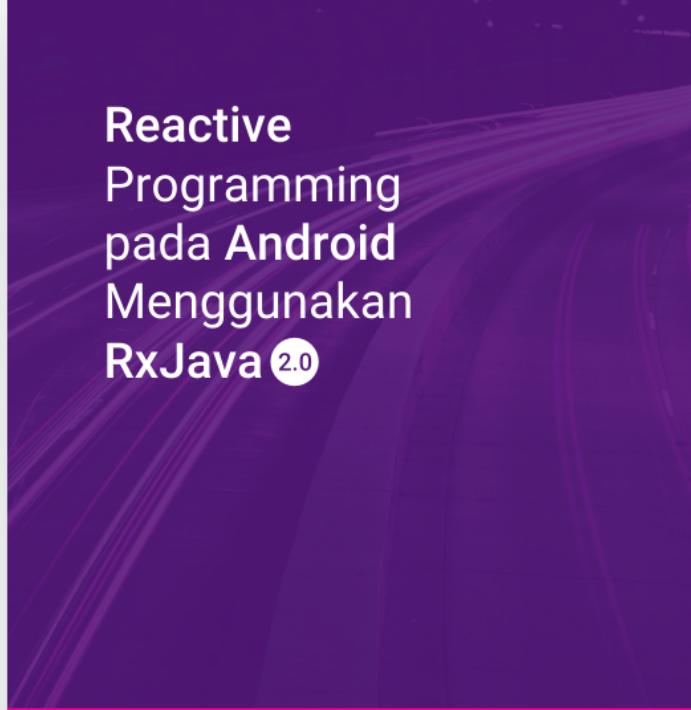
Jetpack comprises the `androidx.*` package libraries, unbundled from the platform APIs. This means that it offers backward compatibility and is updated more frequently than the Android platform, making sure you always have access to the latest and greatest versions of the Jetpack components.

GET STARTED **WATCH THE INTRO VIDEO**

 Accelerate development

 Eliminate boilerplate code

 Build high quality, robust apps



Reactive Programming pada Android Menggunakan RxJava 2.0



Muh Isfhani Ghiath

Let's develop your next app with Rx!

@isfaaghyth