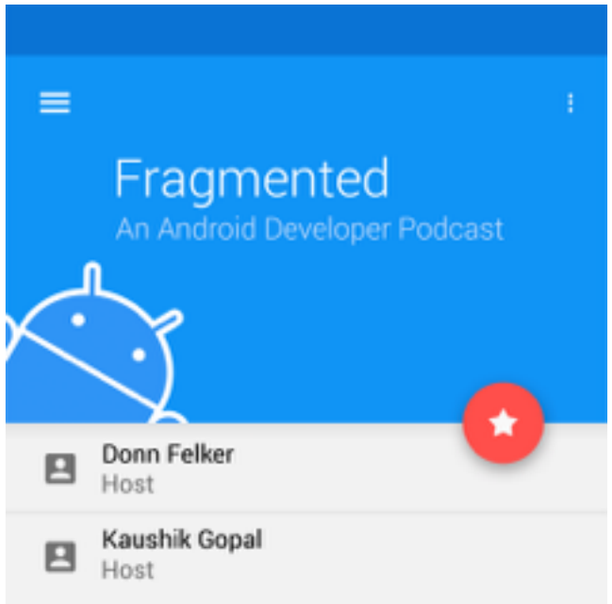
Learning RxJava (for Android) by example

Kaushik Gopal

Wedding Party

http://weddingpartyapp.com



Fragmented Podcast

http://fragmentedpodcast.com

Primer on RxJava

RxJava

```
compile 'io.reactivex:rxjava:1.0.11'
```

RxAndroid

```
compile 'io.reactivex:rxandroid:0.24.0'
```

Primer on RxJava

Anatomy

```
Observable.from(_doLongNetworkOp())
```

→ 1. Observable

```
.subscribeOn(Schedulers.io())
.observeOn(AndroidSchedulers.mainThread())
```

→ 3. Schedulers

```
.subscribe(_getObserver());
```

→ 2. Observer

Observable ⟷ Observer

=

4. Subscription

Primer on RxJava

Anatomy

```
Observable.from(
.subscribeOn(Schedulers.io())
.observeOn(AndroidSchedulers.mainThread())
```

→ 1. Observable

→ 3. Schedulers

```
.subscribe(_getObserver());
```

→ 2. Observer

```
/**
 * Observer that handles the result through the 3 important actions:
 *
 * 1. onCompleted
 * 2. onError
 * 3. onNext
 */
private Observer<Boolean> _getObserver() {
    return new Observer<Boolean>() {

        @Override
        public void onCompleted() {...}

        @Override
        public void onError(Throwable e) {...}

        @Override
        public void onNext(Boolean bool) {...}
    };
}
```

Primer on RxJava

Anatomy

```
Subscription s = Observable.from(_doLongNetworkOp())
                           .subscribeOn(Schedulers.io())
                           .observeOn(AndroidSchedulers.mainThread())
                           .subscribe(_getObserver());
```

1. s.unsubscribe()
2. CompositeSubscription cs; cs.add(s);
3. cs.unsubscribe()

Example 1

Death to AsyncTasks

```
getUserInfoATask.execute(_username.getText().toString())
;

AsyncTask getUserInfoATask =
  new AsyncTask<String, Void, User>() {

     @Override
     protected User doInBackground(String... params) {

        return _api.getUser(params[0]);
     }

     @Override
     protected void onPostExecute(User user) {

        _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
     }                  user.email));
};
```

1. Observable

```
vable.just(_username.getText().toString())

    .map(new Func1<String, User>() {
        @Override
        public User call(String username) {
            return _api.getUser(username);
        }
    })
```

Example 1

Death to AsyncTasks

```
getUserInfoATask.execute(_username.getText().toString());

AsyncTask getUserInfoATask =
  new AsyncTask<String, Void, User>() {

     @Override
     protected User doInBackground(String... params) {

        return _api.getUser(params[0]);

     }

     @Override
     protected void onPostExecute(User user) {

        _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
                        user.email));
     }
};
```

1.Observable

```
_api.user(username)
```

Yay Retrofit!

```
/**
 * See https://developer.github.com/v3/users/
 */
@GET("/users/{user}")
User getUser(@Path("user") String user);

@GET("/users/{user}")
Observable<User> user(@Path("user") String user);
```

Example 1

Death to AsyncTasks

1.0Observable

```
getUserInfoATask.execute(_username.getText().toString());

AsyncTask getUserInfoATask =
  new AsyncTask<String, Void, User>() {

    @Override
    protected User doInBackground(String... params) {

        return _api.getUser(params[0]);

    }
```

```
    @Override
    protected void onPostExecute(User user) {
        _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
                        user.email));
        }
    };
```

```
Observable.just(_username.getText().toString())

        .map(new Func1<String, User>() {
            @Override
            public User call(String username) {
                return _api.getUser(username);
            }
        })
```

2.0Observer

```
        .subscribe(new Observer<User>() {
            @Override
            public void onCompleted() {...}

            @Override
            public void onError(Throwable e) {...}

            @Override
            public void onNext(User user) {
                _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
                        user.email));
            }
        });
```

Example 1

Death to AsyncTasks

```
getUserInfoATask.execute(_username.getText().toString());

AsyncTask getUserInfoATask =
  new AsyncTask<String, Void, User>() {

    @Override
    protected User doInBackground(String... params) {

        return _api.getUser(params[0]);

    }
```

```
    @Override
    protected void onPostExecute(User user) {
        _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
                        user.email));
        }
    };
```

1.Observable

```
Observable.just(_username.getText().toString())

        .map(new Func1<String, User>() {
            @Override
            public User call(String username) {
                return _api.getUser(username);
            }
        })
```

3.Schedulers

```
        .subscribeOn(Schedulers.io())
        .observeOn(AndroidSchedulers.mainThread())
```

2.Observer

```
        .subscribe(new Observer<User>() {
            @Override
            public void onCompleted() {...}

            @Override
            public void onError(Throwable e) {...}

            @Override
            public void onNext(User user) {
                _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
                        user.email));
            }
        });
```

Example 1

Death to AsyncTasks

```
getUserInfoATask.execute(_username.getText().toString());

AsyncTask getUserInfoATask =
  new AsyncTask<String, Void, User>() {

    @Override
    protected User doInBackground(String... params) {

        return _api.getUser(params[0]);

    }

    @Override
    protected void onPostExecute(User user) {
        _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
                        user.email));
    }
};
```

1. Error handling
2. Lifecycle changes
3. Caching (rotation)
4. composing multiple calls

```
Observable.just(_username.getText().toString())

        .map(new Func1<String, User>() {
            @Override
            public User call(String username) {
                return _api.getUser(username);
            }
        })

.subscribeOn(Schedulers.io())
.observeOn(AndroidSchedulers.mainThread())


.subscribe(new Observer<User>() {
    @Override
    public void onCompleted() {...}

    @Override
    public void onError(Throwable e) {...}

    @Override
    public void onNext(User user) {
        _adapter.add(format("%s  = [%s: %s]",
                _username.getText(),
                user.name,
                user.email));
    }
});
```

Example 1

Death to AsyncTasks

```
getUserInfoATask.execute(_username.getText().toString());

AsyncTask getUserInfoATask =
  new AsyncTask<String, Void, User>() {

    @Override
    protected User doInBackground(String... params) {

        return _api.getUser(params[0]);

    }

    @Override
    protected void onPostExecute(User user) {
        _adapter.add(format("%s  = [%s: %s]",
                        _username.getText(),
                        user.name,
                        user.email));
        }
    };
```

1. Error handling
2. Lifecycle changes
3. Caching (rotation)
4. composing multiple calls

```
_api.user(username)
.subscribeOn(Schedulers.io())
.observeOn(AndroidSchedulers.mainThread())


.subscribe(new Observer<User>() {
    @Override
    public void onCompleted() {...}

    @Override
    public void onError(Throwable e) {...}

    @Override
    public void onNext(User user) {
        _adapter.add(format("%s  = [%s: %s]",
                _username.getText(),
                user.name,
                user.email));
    }
});
```

Example 2

Death to TimerTasks

```
int START_DELAY = 0;
int POLLING_INTERVAL = 3000;

final Handler handler = new Handler();
Timer timer = new Timer();

timer.scheduleAtFixedRate(new TimerTask() {
    public void run() {
        handler.post(new Runnable() {

            @Override
            public void run() {
                // do something here
            }
        });
    }
}, START_DELAY, POLLING_INTERVAL);



timer.cancel();
```

Example 2

Death to TimerTasks

```
int START_DELAY = 0;
int POLLING_INTERVAL = 3000;

final Handler handler = new Handler();
Timer timer = new Timer();

timer.scheduleAtFixedRate(new TimerTask() {
    public void run() {
        handler.post(new Runnable() {

            @Override
            public void run() {
                // do something here
            }
        });
    }

}, START_DELAY, POLLING_INTERVAL);
```

```
int ST_DELAY = 0;
int POLL_INTERVAL = 3;
```

```
Observable.timer(ST_DELAY,
                 POLL_INTERVAL, TimeUnit.SECONDS)
```

Example 2

Death to TimerTasks

```
int START_DELAY = 0;
int POLLING_INTERVAL = 3000;

final Handler handler = new Handler();
Timer timer = new Timer();

timer.scheduleAtFixedRate(new TimerTask() {

   public void run() {
       handler.post(new Runnable() {

           @Override
           public void run() {
               // do something here
           }
       });
   }

}, START_DELAY, POLLING_INTERVAL);
```

```
int ST_DELAY = 0;
int POLL_INTERVAL = 3;
```

```
Observable.timer(ST_DELAY,
                 POLL_INTERVAL, TimeUnit.SECONDS)
```

```
        .subscribe(new Observer<Long>() {
                   @Override
                   public void onCompleted() {...}

                   @Override
                   public void onError(Throwable e) {...}

                   @Override
                   public void onNext(Long number) {
                         // do something here
                   }
               });
```

Example 2

Death to TimerTasks

```
int START_DELAY = 0;
int POLLING_INTERVAL = 3000;

final Handler handler = new Handler();
Timer timer = new Timer();

timer.scheduleAtFixedRate(new TimerTask() {

   public void run() {
       handler.post(new Runnable() {

           @Override
           public void run() {
               // do something here
           }
       });
   }

}, START_DELAY, POLLING_INTERVAL);

 timer.cancel();
```

```
int ST_DELAY = 0;
int POLL_INTERVAL = 3;

_subscription =

Observable.timer(ST_DELAY,
                 POLL_INTERVAL, TimeUnit.SECONDS)

       .subscribe(new Observer<Long>() {
                  @Override
                  public void onCompleted() {...}

                  @Override
                  public void onError(Throwable e) {...}

                  @Override
                  public void onNext(Long number) {
                        // do something here
                  }
              });


_subscription.unsubscribe()
```

Example 2

Death to TimerTasks

```
// execute task once after a delay
Observable.timer(START_DELAY, TimeUnit.SECONDS)
```

```
// executing task with delay, every X seconds
Observable.timer(START_DELAY, POLL_INTERVAL, TimeUnit.SECONDS)
```

```
// nicer api
Observable.interval(POLL_INTERVAL, TimeUnit.SECONDS);
```

```
// execute at an interval but only 20 times
Observable.interval(POLL_INTERVAL, TimeUnit.SECONDS)
          .take(20)
          ...
          .flatMap()
          .map()
```

Demo!

Example 3

"For" Smarter Auto Complete

Demo!

```
Observable<OnTextChangeEvent> textObservable =
                                    WidgetObservable.text(_inputEditText);


textObservable
    .debounce(400, TimeUnit.MILLISECONDS, Schedulers.computation())//
    .observeOn(AndroidSchedulers.mainThread()))//

.subscribe(_getSearchObserver());
```

Example 3

For Smarter Auto Complete

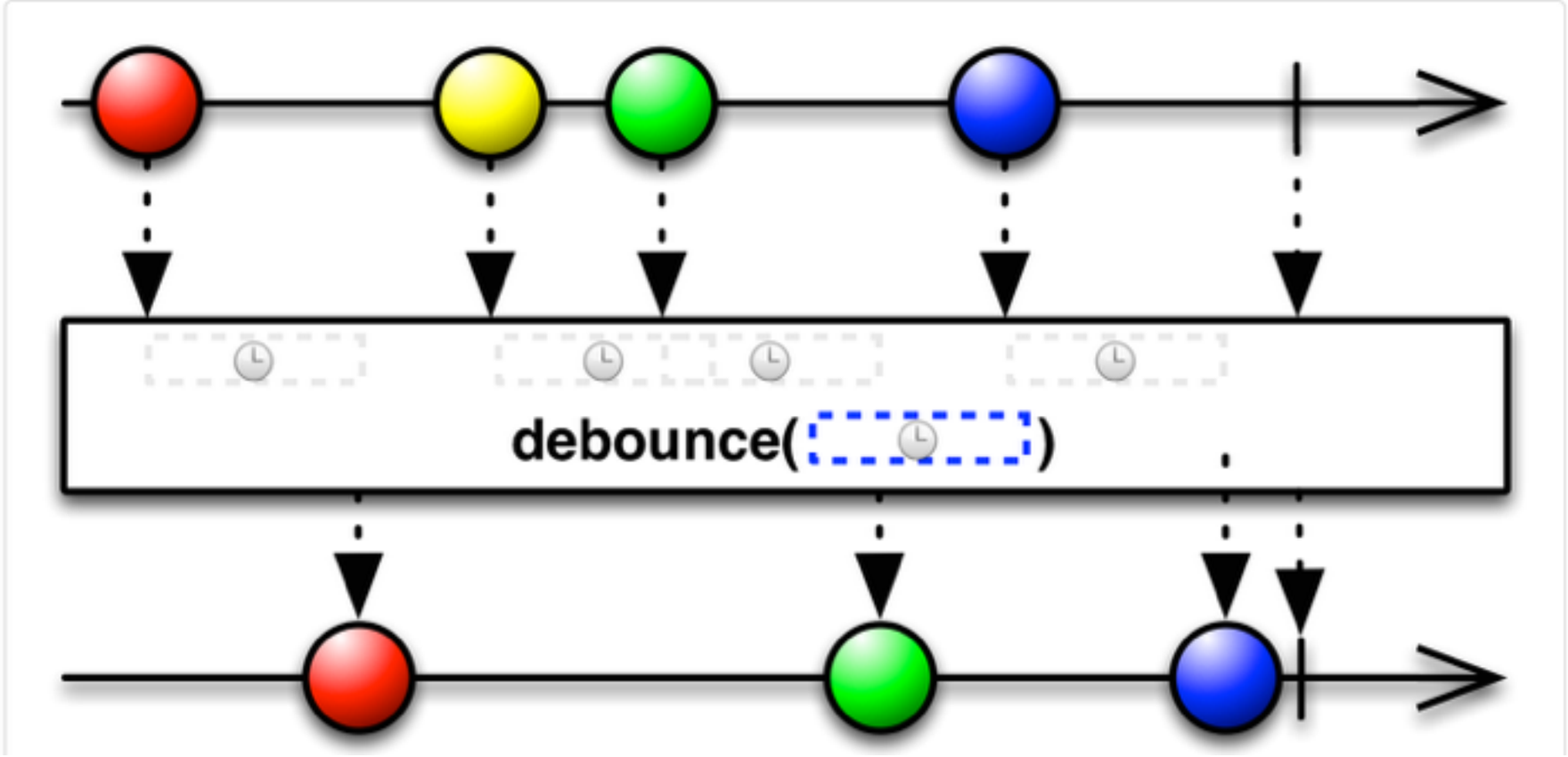
- `debounce( )` — only emit an item from the source Observable after a particular timespan has passed without the Observable emitting any other items

```
textObservable
    .debounce(400, TimeUnit.MILLISECONDS, Schedulers.io())//
```

Example 3

For Smarter Auto Complete

- `debounce( )` — only emit an item from the source Observable after a particular timespan has passed without the Observable emitting any other items



Example 3

For Smarter Auto Complete

```
textObservable
    .debounce(400, TimeUnit.MILLISECONDS, Schedulers.computation())

    .observeOn(AndroidSchedulers.mainThread()))

    .subscribe( new Observer<OnTextChangeEvent>() {
      @Override
      public void onCompleted() {}

      @Override
      public void onError(Throwable e) {}

      @Override
      public void onNext(OnTextChangeEvent onTextChangeEvent) {
          _log(format("You searched for %s",
                      onTextChangeEvent.text().toString()));
      }
});
```

Example 4

Form Validation - CombineLatest

Demo!

```
_emailChangeObservable = WidgetObservable.text(_emailEditText);
_passwordChangeObservable = WidgetObservable.text(_passwordEditText);
_numberChangeObservable = WidgetObservable.text(_numberEditText);
```

Example 4

Form Validation - CombineLatest

```
Observable.combineLatest(

        _emailChangeObservable,
        _passwordChangeObservable,
        _numberChangeObservable,
        (Func3) (onEmailChangeEvent, onPasswordChangeEvent, onNumberChangeEvent)
```

Observable

Example 4

Form Validation - CombineLatest

Observable

```
Observable.combineLatest(
        _emailChangeObservable,
        _passwordChangeObservable,
        _numberChangeObservable,
        (Func3) (onEmailChangeEvent, onPasswordChangeEvent, onNumberChangeEvent)

         boolean emailValid =
                 !isNullOrEmpty(onEmailChangeEvent.text()) &&
                 EMAIL_ADDRESS.matcher(onEmailChangeEvent.text()).matches();

        if (!emailValid)
             _email.setError("Invalid Email!");
```

Example 4

Form Validation - CombineLatest

Observable

```
Observable.combineLatest(

        _emailChangeObservable,
        _passwordChangeObservable,
        _numberChangeObservable,
        (Func3) (onEmailChangeEvent, onPasswordChangeEvent, onNumberChangeEvent)

         boolean emailValid =
                !isNullOrEmpty(onEmailChangeEvent.text()) &&
                EMAIL_ADDRESS.matcher(onEmailChangeEvent.text()).matches();

        if (!emailValid)
             _email.setError("Invalid Email!");

        boolean passValid =
           !isNullOrEmpty(onPasswordChangeEvent.text()) &&
           onPasswordChangeEvent.text().length() > 8;

        if (!passValid)
           _password.setError("Invalid Password!");
```

Example 4

Form Validation - CombineLatest

Observable

```
Observable.combineLatest(

        _emailChangeObservable,
        _passwordChangeObservable,
        _numberChangeObservable,
        (Func3) (onEmailChangeEvent, onPasswordChangeEvent, onNumberChangeEvent)

         boolean emailValid =
                !isNullOrEmpty(onEmailChangeEvent.text()) &&
                EMAIL_ADDRESS.matcher(onEmailChangeEvent.text()).matches();

        if (!emailValid)
            _email.setError("Invalid Email!");

        boolean passValid =
           !isNullOrEmpty(onPasswordChangeEvent.text()) &&
           onPasswordChangeEvent.text().length() > 8;

        if (!passValid)
           _password.setError("Invalid Password!");

        // numValid
```

Example 4

Form Validation - CombineLatest

Observable

```
Observable.combineLatest(

        _emailChangeObservable,
        _passwordChangeObservable,
        _numberChangeObservable,
        (Func3) (onEmailChangeEvent, onPasswordChangeEvent, onNumberChangeEvent) -> {

            boolean emailValid =
                    !isNullOrEmpty(onEmailChangeEvent.text()) &&
                    EMAIL_ADDRESS.matcher(onEmailChangeEvent.text()).matches();

            if (!emailValid)
                _email.setError("Invalid Email!");

            boolean passValid =
                !isNullOrEmpty(onPasswordChangeEvent.text()) &&
                onPasswordChangeEvent.text().length() > 8;

            if (!passValid)
                _password.setError("Invalid Password!");

            // numValid

            return emailValid && passValid && numValid;
        })
```

Example 4

Form Validation - CombineLatest

Observable ⟷ Observer

```
.subscribe(new Observer<Boolean>() {
    @Override
    public void onCompleted() {
        Timber.d("completed");
    }

    @Override
    public void onError(Throwable e) {
        Timber.e(e, "there was an error");
    }

    @Override
    public void onNext(Boolean formValid) {
        if (formValid) {
            _btnValidIndicator.turnValid();
        } else {
            _btnValidIndicator.turnInvalid();
        }
    }
});
```

Example 4

Form Validation - CombineLatest


```
Observable.combineLatest(
        _emailChangeObservable,
          _passwordChangeObservable,
          _numberChangeObservable,

    (Func3) (onEmailChangeEvent,
             onPasswordChangeEvent,
             onNumberChangeEvent)

  boolean emailValid =
      !isNullOrEmpty(onEmailChangeEvent.text()) &&
      EMAIL_ADDRESS.matcher(onEmailChangeEvent.text()).matches();

  if (!emailValid)
      _email.setError("Invalid Email!");

  boolean passValid =
         !isNullOrEmpty(onPasswordChangeEvent.text()) &&
         onPasswordChangeEvent.text().length() > 8;

   if (!passValid)
       _password.setError("Invalid Password!");

    // numValid

    return emailValid && passValid && numValid;
})
```

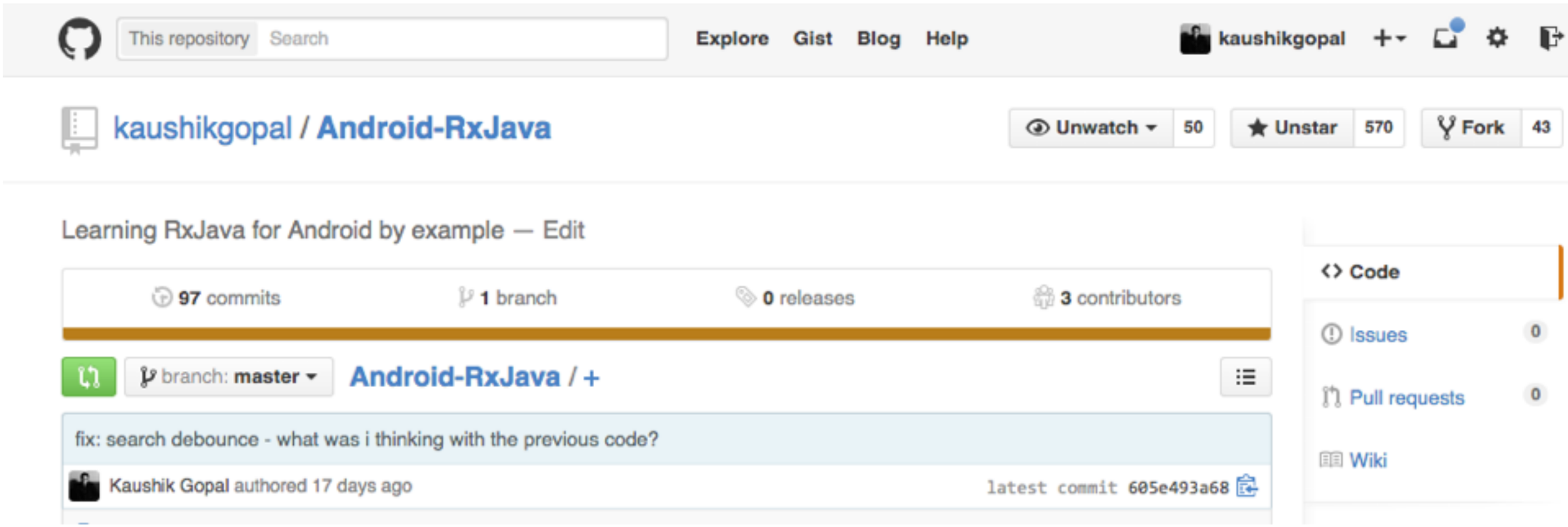
```
.subscribe(new Observer<Boolean>() {
    @Override
    public void onCompleted() {
        Timber.d("completed");
    }

    @Override
    public void onError(Throwable e) {
        Timber.e(e, "there was an eroor");
    }

    @Override
    public void onNext(Boolean formValid) {
        if (formValid) {
            _btnValidIndicator.turnValid();
        } else {
            _btnValidIndicator.turnInvalid();
        }
    }
});
```

Go forth and use RxJava

1. AsyncTask
2. TimerTask (all timing/interval)
3. Smarter AutoComplete
4. Form validations

More examples ?

https://github.com/kaushikgopal/Android-Rxjava



1. Look at the examples
2. Send PRS to clean my old crappy code
3. Contribute more examples