“Observable” Creation, Composition, and Filtering

Russell Elledge
@russellelledge

hardcore dev and IT training

Agenda

“Observable” Creation

Composition

Filtering

Observable Creation Lifecycle

Observer

onNext(T)

onCompleted(T)

onError(Throwable t)

<interface>
Observer

Subscription

unsubscribe()

<inteface>
Subscription

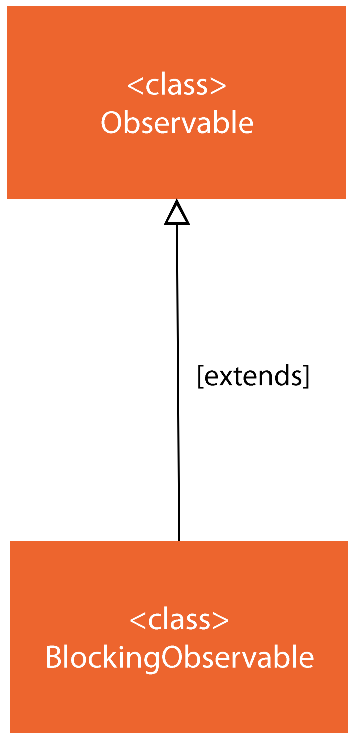
Observable Creation - Types of Observables

Non-Blocking Observables

- Asynchronous execution supported
- Unsubscribe at any point in the event stream

Blocking Observables

- BlockingObservable subclass
- Events are synchronous
- No ability to unsubscribe



Observable Creation - Schedulers

<class>

Schedulers

- computation()
- currentThread()
- immediate()
- io()
- newThread()
- executor(Executor)
- executor(ScheduledExecutor)

<class>

Observable

- subscribeOn(Scheduler)
- observeOn(Scheduler)

Observable Creation

<Static Factory Method>
Observable.from(…)

```
<T> Observable<T> Observable.from( T object [ , Scheduler s ] )
<T> Observable<T> Observable.from( Iterable<T> list [ , Scheduler s ] )
<T> Observable<T> Observable.from( T[] array [ , Scheduler s ] )
<T> Observable<T> Observable.from( Future<T> future [ , Scheduler s ] )
```

<Static Factory Method>
BlockingObservable.from(…)

```
<T> Observable<T> BlockingObservable.from( Observable o )
```

“Observable” Creation

Demonstration

Review - “Observable” Creation

Q: What type of object is used to tell RxJava which thread to execute your Observer code on?

A: Scheduler

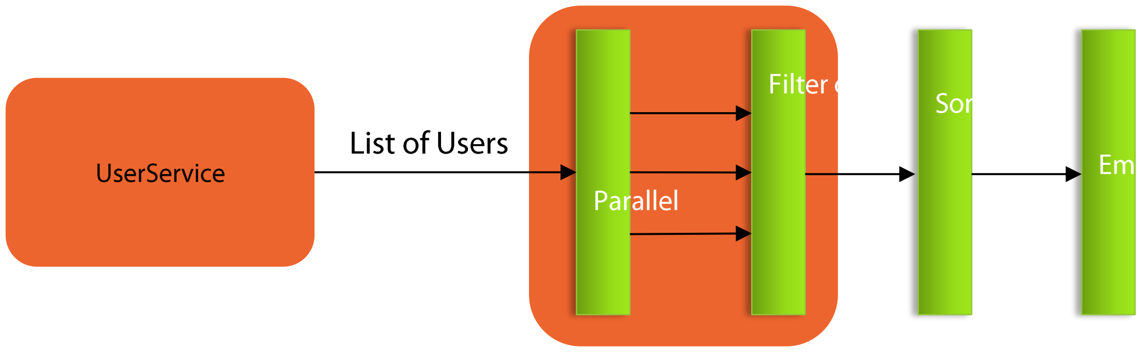
Q: What method is called on an Observer interface when there is an unhandled exception?

A: onError

Q: What are the two types of Observables within the RxJava library?

A: Blocking and Non-Blocking

Composition



1. Fetch a list of users from a "UserService".
2. Add a parallel operation so that our next step is executed using our computational scheduler.
3. In parallel, we filter our any user that is an administrator.
4. Next we will order the user list based on each user's security level.
5. Finally, emit JSON for each user.

Composition

Demonstration

Review - Composition

Q: What is the pattern used by the RxJava library to build up simple event handling functions into complex programming logic?

A: Composition

Q: What operation can be used to ensure that the code of an event handler is performed concurrently?

A: parallel

Q: True or False? By default, RxJava is single threaded.

A: True

Filtering

Predicates

Positional Filters

Time Based Filters

Demonstration

Review - Filtering

Q: What type of filter operations can be performed in order to extract events from an observable at a specific location in the stream of events?

A: Positional Filters

Q: What type of filtering operations can be used to change how events are reported by Observables over time?

A: Time Based Filters

Q: When the "timeout" operation is triggered because there have been no events within the specified timeout period, what observer method is called?
A: onError()

Review

“Observable” Creation

Composition

Filtering