Chaining Observables in RxJS

Asked 3 years, 1 month ago   Active 2 years, 10 months ago   Viewed 72k times

▲

50

▼

★

13

I'm learning RxJS and Angular 2. Let's say I have a promise chain with multiple async function calls which depend on the previous one's result which looks like:

```javascript
var promiseChain = new Promise((resolve, reject) => {
  setTimeout(() => {
    resolve(1);
  }, 1000);
}).then((result) => {
  console.log(result);

  return new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve(result + 2);
    }, 1000);
  });
}).then((result) => {
  console.log(result);

  return new Promise((resolve, reject) => {
    setTimeout(() => {
    resolve(result + 3);
      }, 1000);
  });
});

promiseChain.then((finalResult) => {
  console.log(finalResult);
});
```

My attempts at doing the same solely using RxJS without the use of promises produced the following:

```javascript
var observableChain = Observable.create((observer) => {
  setTimeout(() => {
    observer.next(1);
    observer.complete();
  }, 1000);
}).flatMap((result) => {
  console.log(result);
```

```javascript
    return Observable.create((observer) => {
      setTimeout(() => {
        observer.next(result + 2);
        observer.complete()
      }, 1000);
    });
  }).flatMap((result) => {
    console.log(result);

    return Observable.create((observer) => {
      setTimeout(() => {
        observer.next(result + 3);
        observer.complete()
      }, 1000);
    });
  });

  observableChain.subscribe((finalResult) => {
    console.log(finalResult);
  });
```

It yields the same output as the promise chain. My questions are

1. Am I doing this right? Are there any RxJS related improvements that I can make to the above code

2. How do I get this observable chain to execute repeatedly? i.e. Adding another subscription at the end just produces an additional 6 though I expect it to print 1, 3 and 6.

   observableChain.subscribe((finalResult) => { console.log(finalResult); });

   observableChain.subscribe((finalResult) => { console.log(finalResult); });

   1 3 6 6

javascript   promise   reactive-programming   rxjs   rxjs5

edited Sep 7 '16 at 11:43   asked Jun 12 '16 at 7:42

user3743222   Harindaka
14.1k   5   54   65   **1,785**   5   32   47

Have put a working example of chaining promises using rxjs6 here: stackoverflow.com/a/55991374/1882064 – arcseldon May 5 at 11:25

1 Answer

About promise composition vs. Rxjs, as this is a frequently asked question, you can refer to a number of previously asked questions on SO, among which :

40

- [How to do the chain sequence in rxjs](#)

- [RxJS Promise Composition (passing data)](#)

- [RxJS sequence equvalent to promise.then()?](#)

✔

Basically, `flatMap` is the equivalent of `Promise.then` .

For your second question, do you want to replay values already emitted, or do you want to process new values as they arrive? In the first case, check the `publishReplay` operator. In the second case, standard subscription is enough. However you might need to be aware of the cold. vs. hot dichotomy depending on your source (cf. [Hot and Cold observables : are there 'hot' and 'cold' operators?](#) for an illustrated explanation of the concept)

edited May 23 '17 at 12:34   answered Jun 12 '16 at 17:55

Community ♦   user3743222
1   1   **14.1k**   5   54   65

Got a question that you can't ask on public Stack Overflow? Learn more about sharing private information with Stack Overflow for Teams.   ✕