

Promise.all behavior with RxJS Observables?

Asked 3 years, 6 months ago Active 4 months ago Viewed 34k times



In Angular 1.x I would sometimes need to make multiple `http` requests and do something with all the responses. I would throw all the promises in an array and call `Promise.all(promises).then(function (results) {...})`.

67



Angular 2 best practices seem to point towards the use of RxJS's `observable` as a replacement to promises in `http` requests. If I have two or more different Observables created from `http` requests, is there an equivalent to `Promise.all()`?



10

javascript

angularjs

rxjs

observable

edited Jun 16 '16 at 11:56



user3743222

14.2k 5 54 66

asked Feb 24 '16 at 16:44



Corey Ogburn

12.4k 20 92 157

4 Answers



The more straightforward alternative for emulating `Promise.all` is to use the `forkJoin` operator (it starts all observables in parallel and join their last elements):

66



- [documentation](#)
- related link: Cf. [RxJS: concat three promises, distinguish results](#)



A bit out of scope, but in case it helps, on the subject of chaining promises, you can use a simple `flatMap`: Cf. [RxJS Promise Composition \(passing data\)](#)

edited May 23 '17 at 12:10



Community ♦

1 1

answered Feb 24 '16 at 17:20



user3743222

14.2k 5 54 66

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Please help, forkJoin doesn't work when observables passed as parameter don't emit values. I have void Observables and still want to use forkJoin functionality but it is not working – [Goga Koreli](#) Nov 16 '18 at 12:39



12

forkJoin works fine too, but I'd prefer [combineLatest](#) since you don't need to worry about it taking the last value of observables. This way, you can just get updated whenever any of them emit a new value too (e.g. you fetch on an interval or something).

answered Feb 25 '16 at 4:54

[kakigoori](#)

1,030 9 20

1 This doesn't meet my current needs but I will definitely be using that soon. – [Corey Ogburn](#) Feb 25 '16 at 5:34

2 That doesn't achieve the same behavior as Promise.all(), but it's similar to Promise.any() – [Purrell](#) Sep 8 '16 at 20:26

if i have 2 calls one return promise and another return observable can i user forkjoin ? or promise.all() ? or no one , i have to let the 2 functions return same type either promises or observable ? – [Joe Sleiman](#) Jun 12 '17 at 7:40

@JoeSleiman a bit late, but you can pick your side: [Observable.fromPromise\(\)](#) together with [Observable.zip\(\)](#), or [Observable.toPromise\(\)](#) with Promise.all() – [Arnaud P](#) Jul 28 '17 at 15:59



11

On [reactivex.io](#) *forkJoin* actually points to [Zip](#), which did the job for me:

```
let subscription = Observable.zip(obs1, obs2, ...).subscribe(...);
```

answered Jul 28 '17 at 16:11

[Arnaud P](#)

7,337 2 37 49



Update May 2019 using RxJs v6

6

Found the other answers useful, and wished to offer an example for the answer offered by Arnaud about `zip` usage.

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```
import { zip } from "rxjs";

const the_weather = new Promise(resolve => {
  setTimeout(() => {
    resolve({ temp: 29, conditions: "Sunny with Clouds" });
  }, 2000);
});

const the_tweets = new Promise(resolve => {
  setTimeout(() => {
    resolve(["I like cake", "BBQ is good too!"]);
  }, 500);
});

// Using RxJs
let source$ = zip(the_weather, the_tweets);
source$.subscribe([weatherInfo, tweetInfo]) =>
  console.log(weatherInfo, tweetInfo)
);

// Using ES6 Promises
Promise.all([the_weather, the_tweets]).then(responses => {
  const [weatherInfo, tweetInfo] = responses;
  console.log(weatherInfo, tweetInfo);
});
```

The output from both are the same. Running the above gives:

```
{ temp: 29, conditions: 'Sunny with Clouds' } [ 'I like cake', 'BBQ is good too!' ]
{ temp: 29, conditions: 'Sunny with Clouds' } [ 'I like cake', 'BBQ is good too!' ]
```

answered May 5 at 14:52



[arcseldon](#)

22.6k 6 75 94