

Angular 2: Convert Observable to Promise

Q) How do I convert the following observable to a promise so I can call it with `.then(...)` ?

39

My method I want to convert to a promise:

```
this._APIService.getAssetTypes().subscribe(  
  assettypes => {  
    this._LocalStorageService.setAssetTypes(assettypes);  
  },  
  err => {  
    this._LogService.error(JSON.stringify(err))  
  },  
  () => {}  
);
```

The service method it calls:

```
getAssetTypes() {  
  var method = "assettype";  
  var url = this.apiUrl + method;  
  
  return this._http.get(url, {}).  
    .map(res => <AssetType[]>res.json())  
    .map((assettypes) => {  
      assettypes.forEach((assettypes) => {  
        // do anything here you might need....  
      });  
      return assettypes;  
    });  
}
```

Thanks!

typescript



angular

promise

observable

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)



The Most answers below have [toPromise\(\)](#) operator, that has been deprecated in RxJS 5.5+ – [Rohit Sharma](#) Sep 7 '18 at 20:32

4 Answers

**rxjs6**

68



<https://github.com/ReactiveX/rxjs/issues/2868#issuecomment-360633707>

Don't pipe. It's on the Observable object by default.

```
Observable.of('foo').toPromise(); // this
```

rxjs5

```
import 'rxjs/add/operator/toPromise';
import 'rxjs/add/operator/map';

...

this._APIService.getAssetTypes()
  .map(assettypes => {
    this._LocalStorageService.setAssetTypes(assettypes);
  })
  .toPromise()
  .catch(err => {
    this._LogService.error(JSON.stringify(err));
  });
```

edited Aug 3 '18 at 20:08

answered Apr 21 '16 at 18:08



[Günter Zöchbauer](#)

Join **Stack Overflow** to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook



What does `getAssetTypes()` return? – [Günter Zöchbauer](#) Apr 21 '16 at 18:29

- 2 @GünterZöchbauer has `import 'rxjs/operator/add/toPromise';` been changed to: `import 'rxjs/add/operator/toPromise';` ? – [Zze](#) Apr 24 '17 at 6:52
- 1 @Zze you're right, thanks for pointing out. – [Günter Zöchbauer](#) Apr 24 '17 at 6:56
- 2 `toPromise` is deprecated. How would we do this now? – [Jus10](#) Aug 3 '18 at 20:02

observable can be converted to promise like this:

16 `let promise=observable.toPromise();`

answered Jan 3 '18 at 11:46



[Luca C.](#)

4,286 43 47

If I conver this to promise I can also catch an error right? – [KnowledgeSeeker](#) Aug 17 '18 at 6:30

you can surround with try catch and even user `promise.catch()` – [Luca C.](#) Aug 17 '18 at 9:46

you dont really need to do this just do ...

11 `import 'rxjs/add/operator/first';`

```
this.esQueryService.getDocuments$.first().subscribe(() => {
  event.enableButtonsCallback();
},
(err: any) => console.error(err)
);
this.getDocuments(query, false);
```

Join Stack Overflow to learn, share knowledge, and build your career.

Sign up with email



Sign up with Google

Sign up with Facebook





danday74

21.6k 17 108 143

Correct, the promise is only resolved at the time the observables completes, using `.first()` or `.take(1)` will ensure it's the case after the first emit. – [Flavien Volken](#) Jan 11 '18 at 9:27

The problem here is we cannot use `async/await` pattern. – [Sampath](#) Mar 14 '18 at 11:27

The proper way to make Observable a Promise, in your case would be following

4

```
getAssetTypesPromise() Observable<any> {  
  return new Promise((resolve, reject) => {  
    this.getAssetTypes().subscribe((response: any) => {  
      resolve(response);  
    }, reject);  
  });  
}
```

[Run code snippet](#)[Expand snippet](#)

answered Oct 8 '18 at 13:26



Teodor Hirs

65 1 4

1 This can be further simplified to `return new Promise((resolve, reject) => this.getAssetTypes().subscribe(resolve, reject));` – [mopo922](#) Dec 18 '18 at 16:34

Join Stack Overflow to learn, share knowledge, and build your career.

[Sign up with email](#)[Sign up with Google](#)[Sign up with Facebook](#)