

## [CGA-JAVA-WFDA] Web Front-end Development Angular 2.1

[Dashboard](#) / [My courses](#) / [CGA-JAVA-WFDA-2.1](#) / [9. Kết nối với Back-end](#) / [\[Bài đọc\] Phân biệt Promise và Observable](#)

# [Bài đọc] Phân biệt Promise và Observable

Có rất nhiều điểm khác nhau giữa Observable và Promise.

Mặc dù Observable được khởi tạo, nhưng điều đó không có nghĩa là nó thực thi ngay lập tức. Nó sẽ không bao giờ được thực thi nếu như chưa được đăng ký. Còn Promise thì lại khác, nó được thực thi ngay khi nó được khởi tạo. Chính điều này khiến Observable có lợi thế hơn khi làm việc với bất đồng bộ.

Observable có thể chứa được nhiều giá trị trong nó, còn Promise thì không. Observable giống như một array, còn Promise giống như một giá trị đơn. Và điều đó khiến Observable linh động hơn trong việc lưu trữ dữ liệu so với Promise. Bên cạnh đó Observable cũng chính là một dòng chảy (stream), và dòng chảy này được thay đổi theo thời gian. Chúng ta có thể truyền bất cứ giá trị nào vào dòng chảy đó và ngay lập tức Observable sẽ emit cho chúng ta giá trị mới.

Dưới đây là một

## Tiền xử lý dữ liệu

Không giống như Promise, Observable có thể thực thi quá trình tiền xử lý dữ liệu trước khi chúng ta đăng ký. Ví dụ:

```
observable.pipe(map((x) => 2 * x));
```

Còn đối với Promise thì chúng ta chỉ có thể xử lý dữ liệu khi Promise trả về dữ liệu

```
promise.then((x) => 2 * x);
```

## Khả năng huỷ

Sau khi chúng ta đã đăng ký một Observable, chúng ta vẫn có thể huỷ nó đi được, nếu như không muốn nó chạy mãi. Còn Promise thì lại không hỗ trợ việc này.

```
const sub = obs.subscribe(...);
sub.unsubscribe();
```

## Hướng sự kiện

Một điểm nữa khiến Observable vượt trội hơn so với Promise là nó có thể làm được các công việc liên quan đến thao tác hướng sự kiện.

Ví dụ như khi click vào một button:

```
import { fromEvent } from "rxjs";
const buttonEl = document.querySelector("button");
const clicks$ = fromEvent(buttonEl, "click");
let subscription = clicks$.subscribe(e => console.log("clicked", e));
```

Để làm được việc đó, chúng ta cần sử dụng fromEvent của thư viện RxJS. fromEvent chính là một Observable, chúng ta tiến hành subscribe. Và khi chúng ta click vào button, Observable lúc này sẽ ngay lập tức emit giá trị về chúng ta. Ngoài ra, chúng ta cũng có thể huỷ sự kiện này nếu như không muốn thực thi nữa

```
subscription.unsubscribe();
```

Ngoài việc sử dụng Observable ra, thì chúng ta có thể làm theo cách thông thường là dùng addEventListener để lắng nghe sự kiện.

```
const buttonEl = document.querySelector("button");
const handler = e => console.log("clicked", e);
buttonEl.addEventListener("click", handler);
```

Và removeEventListener để huỷ việc lắng nghe sự kiện

```
button.removeEventListener('click', handler);
```

Last modified: Thursday, 25 February 2021, 10:05 AM

CHƯƠNG TRÌNH

- Career
- Premium
- Accelerator

TÀI NGUYÊN

- Blog
- Tạp chí Lập trình
- AgileBreakfast

Follow Us



CodeGym@2018. All rights reserved.

You are logged in as Dương Văn Thanh Sơn (Log out)