[Cùng Học Angular 2] Căn bản về RxJS trong Angular 2

angular2 4 rxjs 4

Sang Nguyễn Đắc viết ngày 07/01/2017

Angular 2 ra mắt kèm theo rất nhiều tính năng khá hữu ích

Một trong số đó là sự xuất hiện của RxJS. Vậy bài này chúng ta sẽ cùng tìm hiểu căn bản về **RxJS** để bắt đầu với **Angular 2**, tức là mình sẽ chỉ giới thiệu sơ lược về RxJS đủ để các bạn làm quen và code được **Angular 2** ở mức căn bản.

Đầu tiên, **RxJS** là viết tắt của ***Reactive Extensions Library for JavaScript*** và là phiên bản **thứ 5**. Các bạn có thể xem chi tiết hơn tại repo chính thức của project này. **Reactive Extensions** không chỉ hỗ trợ cho **Javascript** mà cho rất nhiều ngôn ngữ khác như **Java** hay **C#**, nhưng **Angular 2** sử dụng **RxJS**, vì thế trong phạm vi bài này chúng ta sẽ chỉ tìm hiểu về ***Reactive Extensions Library for JavaScript*** mà thôi.

RxJS đơn giản là một thư viện xử lý bất đồng bộ (asynchronous data streams) hay nói đúng hơn là nó không hề xa lạ, hầu hết các sự kiện mà bạn làm việc cùng đều là một *"luồng"* các sự kiện khác nhau và **RxJS** sẽ giúp bạn tạo được các luồng sự kiện này từ *"mọi thứ"* (ít nhất là đối với javascript).

Để giải thích rõ hơn về RxJS thì cần rất nhiều giấy mực nhưng có một bài viết rất hay tại đây mà các bạn rất nên đọc, việc dịch thuật sang tiếng Việt khá là khó khăn nên mình sẽ không nói trong phạm vi bài viết này.

Còn nếu các bạn muốn hiểu một cách đơn giản nhất thì Rx có nghĩa là tạo một cái "ống nước" chứa dữ liệu, sự kiện mà bạn có thể nắm bắt và điều khiển được cách mà cái ống nước này chạy. Và nhân nói về việc này thì nó bao gồm 2 giai đoạn, thứ nhất là tạo ra cái ống nước và thứ hai mở cho ống nước này chạy tức là sẽ không có chuyện gì xảy ra nếu ống nước này không được mở, các bạn sẽ thấy điều này

rõ hơn khi mình demo. RxJS cung cấp một bộ các công cụ cực kì đầy đủ để các bạn làm được việc này.

Lý thuyết thế đủ rồi, bây giờ là lúc cho thực hành!

Cách để học RxJS đơn giản nhất là các bạn vào jsbin và chọn "Add library" tìm đến mục RxJS và chọn bản mới nhất (tức là 5.x), việc này sẽ tạo một biến toàn cục Rx.

Một note nhỏ là code trong bài viết này được viết bằng ES6 nên nếu bạn cảm thấy syntax lạ thì hãy học một chút về ES6 trước, thực sự là nó cực kì quan trọng nếu bạn muốn code được Angular 2.

Vậy hãy cùng bắt đầu với những lệnh cơ bản nhất, mở tab **Javascript** và **Console**, chuyển tab javascript sang sử dụng **ES6/Babel**.

```
const testInterval = Rx.Observable.interval(1000);
```

Ý nghĩa của đoạn này là gì, đó là các bạn đã thiết lập một ống nước, **Rx** là biến toàn cục mà ta có được khi add thư viện **RxJS** vào, **Observable** nôm na là một luồng dữ liệu và interval có nghĩa là cứ sau một khoảng thời gian (1s) thì nó sẽ được kích hoạt một lần.

Nhưng như mình đã nói ở trên, nếu bản mở console thì sẽ thấy là không hề có chuyện gì xuất hiện, bởi vì đơn giản là cái ống nước này chưa hề được mở, vậy hãy thêm đoạn này.

```
test.subscribe(val => console.log(val));
```

Lúc này thì mở console và bạn sẽ thấy việc các số được tăng dần sau mỗi 1s. Việc này nghĩa là ta mở cho ống nước chạy. Nhưng những dữ liệu này từ đâu ra, chính

bản thân lệnh interval chứa giá trị số nguyên và tăng lên sau mỗi một thời gian nhất định.

Tiếp theo, hãy đến với một ví dụ rõ ràng hơn:

```
nst arrObservable = Rx.Observable.from(arr).subscribe(val => console.log(val));
```

Vậy là bạn đã thấy là chúng ta hoàn toàn có thể tạo ra một **Observable** ngay từ một mảng đơn giản, nhưng hiện tại tất cả những gì đoạn code này làm được là... in cái mảng này ra, thế nên hãy làm tiếp một số ví dụ với đoạn code này.

```
rvable.from(arr).delay(3000)
ubscribe(val => console.log(val));
```

Đúng như bạn đang nghĩ, đoạn code này sẽ bị delay 3s rồi mới được kích hoạt.

```
rvable.from(arr).map(val => val*2)
ubscribe(val => console.log(val));
```

Bây giờ thì thay vì in ra mảng như bình thường thì tất cả các phần tử đều sẽ được nhân 2.

```
able = Rx.Observable.from(arr)
subscribe(val => console.log(val));
```

Bây giờ sẽ chỉ có các phần tử lớn hơn 4 mới được in ra. Vậy sau ví dụ này, bạn sẽ thấy thực sự rất dễ dàng điều khiển được luồng dữ liệu này thông qua Rx và điều thứ hai là các operator đều sẽ được kích hoạt theo thứ tự: map sẽ được kích hoạt trước rồi mới đến filter.

Hãy đến với một ví dụ khác, giờ hãy thêm một thư viện khác là **jQuery** tương tự như cách chúng ta đã làm với RxJS và thêm một button với id *'button'*.

```
utton, 'click')
    .subscribe(event => console.log('Clicked'));
```

Bây giờ, thử kích vào button, và ta đã có một **observable** ngay từ một event, thử một số sự kết hợp khác nào.

```
bservable
ent(button, 'click')
.subscribe(event => console.log('Clicked'));
```

Bây giờ, dù bạn có kích chuột bao nhiêu lần vào button thì đoạn log cũng chỉ xuất hiện 2 lần.

```
bservable
lick')
ceTime(1000)
subscribe
),
or'),
 => console.log('Done')
```

Hay là bây giờ, kích chuột thật nhanh vào button, điều gì đang xảy ra, đoạn log chỉ xuất hiện mỗi khi bạn thả chuột cách 1s. Nhưng sau 3 lần thì đoạn log Done sẽ hiện ra, và phần code của **subscribe** cũng đã khác trước. Điều này có nghĩa là thực

ra **subscribe** chứa 3 đối số khác nhau là **onNext (next)**, **onError (error)** và **onComplete ()** nên sau khi đã hoàn thành (take(3)) thì đoạn log Done sẽ cho ta biết là luồng dữ liệu này đã kết thúc. Ta sẽ không thấy được lỗi vì đơn giản đây chỉ là một event click đơn giản.

Bây giờ, hãy tạo một input có id là input, chúng ta sẽ đến với ví dụ cuối cùng.

```
const input = $('#input');

rvable

ıp')

d()

ubscribe(value => console.log(value));
```

Đây là một đoạn code khác phức tạp với sự kết hợp của khá nhiều operator, nhưng bạn có thể hiểu như thế này: tạo **observable** từ một event keyup của input, sau đó lấy về giá trị của input, chỉ kích hoạt khi giá trị này thay đổi và cuối cùng là chỉ kích hoạt mỗi 0.3s. Đây là một đoạn code cơ bản cho search box.

Đơn giản vậy thôi, bạn chỉ cần vào trang chủ và tìm những operators hoặc effects mà mình cần, lắp nối chúng lại với nhau. Muốn quen được RxJS thì chỉ cần làm nhiều và bạn sẽ thấy có những operators phù hợp với công việc này hơn là những operator khác. Nhưng cũng nên lưu ý là có hàng trăm cách khác nhau để làm cùng một việc với RxJS, và cách nào là do cách mà bạn lắp ghép mà thôi, RxJS như một bộ đồ chơi lego mà bạn có thể thỏa sức sáng tạo.

Bài viết ở trên chỉ bao gồm những operator căn bản nhất để làm việc với **DOM**, còn về phần **Ajax** thì mình sẽ đề cập tới khi viết một bài về app **Angular 2** nhỏ. Cuối cùng, bạn cũng có thể học thêm về những operator nâng cao ở trang này. Sử dụng JSbin, thêm thư viện RxJS và bắt tay vào học thôi.

Nhật Khánh - VTeam - VBlog

Chia sẻ bài viết với bạn bè nữa nhé!

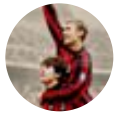
FACEBOOK GOOGLE PLUS TWITTER

Bình luận

Duc Hoang gần 2 năm

Bài viết rất hay, cám ơn bạn nhiều

Hay
0



Toan Van gần 2 năm

thanks tac gia nhe 💯

Hay
0

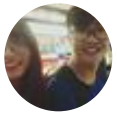
Đăng nhập để bình luận :)

Sang Nguyễn Đắc

6 bài viết. 13 người follow

Kipalog Follow

Cùng một tác giả



18 10

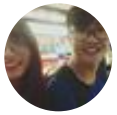
Cài đặt và thiết lập môi trường làm việc cơ bản cho Frontend Web Developer

Web frontend

Mở đầu series bài viết tìm hiểu về Frontend Web mình sẽ hướng dẫn các bạn cài đặt và thiết lập môi trường làm việc hiệu quả. Các ứng dụng cần th...

Sang Nguyễn Đắc viết hơn 2 năm trước

18 10

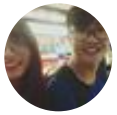


11 2

Viết CSS hiệu quả hơn với Sass và Gulp (Phần 1)

Sass frontend gulp js

I. Sass và Gulp là gì? 1. Sass (Link) là một CSS PreProcessor (Tương tự Less, Stylus, …). Sass được sinh ra để giúp bạn viết CSS nhanh và rõ ràn...
Sang Nguyễn Đắc viết hơn 2 năm trước
11 2



5 5

[Cùng học Angular 2] Cài đặt môi trường

angular2

Mình viết bài này vì muốn giới thiệu cho các bạn cách cài đặt một môi trường cơ bản nhất để làm quen với Angular 2 dành cho các bạn mới bắt đầu hoặ...
Sang Nguyễn Đắc viết hơn 2 năm trước
5 5

Bài viết liên quan

4 1

Thử dùng RxJS thay thế Redux

TIL rxjs redux reactjs

Trải qua 1 thời gian khá dài trải nghiệm Angular, mình thấy RxJS khá hay. Bài này sẽ quay trở lại với bài toán Counter thần kì của React/Redux, như...

cdxf viết 11 tháng trước

4 1

4 0

Rxjs Và Reactive Programming

rxjs tiepphan.com laptrinhthatkydieu

Hẳn các bạn vẫn còn nhớ trong một số bài trước chúng ta có nói về Observable trong ứng dụng Angular, vậy Observable là gì, nó có quan hệ gì với Ang...

Tiep Phan viết gần 2 năm trước

4 0