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

[Bài đọc] Promises

1. Promise là gì ?

Promise sinh ra để xử lý kết quả của một hành động cụ thể, kết quả của mỗi hành động sẽ là thành công hoặc thất bại và Promise sẽ giúp chúng ta giải quyết câu hỏi "Nếu thành công thì làm gì? Nếu thất bại thì làm gì?". Cả hai câu hỏi này ta gọi là một hành động gọi lại (callback action)

2. Các trạng thái của promise

Khi một Promise được khởi tạo thì nó có một trong ba trạng thái sau:

- `Fulfilled`: hành động xử lý xong và thành công
- `Rejected`: hành động xử lý xong và thất bại
- `Pending`: hành động đang chờ xử lý hoặc bị từ chối

Trong đó hai trạng thái `Reject` và `Fulfilled` ta gọi là `Settled`, tức là đã xử lý xong.

3. Cách tạo 1 promise

- Để tạo 1 promise sẽ sử dụng cú pháp sau:

```
var promise = new Promise(function (resolve, reject) {
});
```

```
Trong đó:
* `resolve` là một hàm callback xử lý cho hành động thành công.
* `reject` là một hàm callback xử lý cho hành động thất bại.
```

4. Thenable trong promise

`Thenable` là một phương thức ghi nhận kết quả của trạng thái (thành công hoặc thất bại) mà ta khai báo ở `Reject` và `Resolve`.
Nó có hai tham số truyền vào là 2 callback function.
Tham số thứ nhất xử lý cho `Resolve` và tham số thứ 2 xử lý cho `Reject`.

```
var promise = new Promise(function(resolve, reject){
 resolve('Success');
 // OR
 reject('Error');
 });
```

```
promise.then(
 function(success){
 // Success
 },
 function(error){
 // Error
 }
 );
```

5. Catch trong promise

`then` có hai tham số callbacks đó là `success` và `error`. Tuy nhiên bạn cũng có thể sử dụng phương thức `catch` để bắt lỗi. Cú pháp:

```
promise.then().catch();
```

Ví dụ:

```
    var promise = new Promise(function(resolve, reject){
        reject('Error!');
    });

    promise.then(function(message){
        console.log(message);
    })
    .catch(function(message){
        console.log(message);
    });
```

6. Một vài ví dụ ứng dụng promise

Giả sử ta có một tác vụ muốn delay 5s rồi in ra chữ “LẬP” tiếp đó 3s in ra chú “TRÌNH” sau cùng 2s in ra chữ “CODEGYM”

- Nếu sử dụng hàm setTimeout thì có thể viết như sau:

```
setTimeout(function(){
   console.log('LẬP');
   setTimeout(function(){
       console.log('TRÌNH');
       setTimeout(function(){
          console.log('CODEGYM');
       },2000);
   },3000);
}, 5000);
```

- Nhưng với việc sử dụng promise ta sẽ viết lại thành như sau.

```
function handleTimeout(timeout) {
    return new Promise(function (resolve, reject) {
        setTimeout(resolve, timeout);
    });
}

var xxx = handleTimeout(5000);
xxx.then(function () {
    console.log('LẬP');
})
.then(function () {
    return handleTimeout(3000);
})
.then(function () {
    console.log('TRÌNH')
})
.then(function () {
    return handleTimeout(1000);
})
.then(function () {
    console.log('CODEGYM');
});
```

Last modified: Wednesday, 24 February 2021, 2:23 PM

Previous Activity | Jump to... | Next Activity