academeez

RXJS

2. Pulled/Pushed Data

Pulled Data

- Pulled data is me getting the data when I’m asking for it.
- I am the active one.

Pulled Data - Function

- A simple example of pulling data is me calling a function

```
function getSomePulledData() {
  return "hello";
}

// im pulling the data by calling the function
getSomePulledData();
```

Pulled Data - Generator Functions

academeez

- We can pull multiple data using Generator functions

```
function* counter() {
  yield 10;
  yield 20;
  yield 30;
  yield 40;
}

const askForData = counter();
console.log(askForData.next().value);
console.log(askForData.next().value);
console.log(askForData.next().value);
console.log(askForData.next().value);
```

Pushing Data

- When data is pushed, I'm passively waiting for the data
- Usually I'm waiting by supplying a callback function that will be called when the data arrives

Pushing Data - Callback

academeez

- Pushed data is usually received by passing a callback

```
setTimeout(() => {
  console.log("this callback will be called after 1sec");
}, 1000);
```

Pushing Data - Promise

- Promise is an example of pushed data

```
fetch("https://nztodo.herokuapp.com/api/tasks/?format=json").then((res) => {
  console.log(
    "waiting for the server to push data to me, and call this callback"
  );
});
```

Pushing Data - Events

- With events you register a callback that will be called when the event happens
- Data is pushed when events happens

```
const button: HTMLButtonElement = document.getElementById(
  "push-button"
) as HTMLButtonElement;

button.addEventListener("click", () => {
  console.log("this callback will be pushed when the button is clicked");
});
```

RXJS pushing data

academeez

- Unlike functions when we call them to ask for data - data pull
- In RXJS data will be pushed to us - data push
- Which means listeners will be attached to a source that push data to them
- RXJS is pushing data to listeners via the Observable pattern

Observable pushing data

academeez

- Observable is in charge of pushing data:

```
import { Observable } from 'rxjs';

const helloObservable: Observable<string> = new Observable((observer) => {
  observer.next("Hello listeners! I'm an RXJS Observable");
  observer.next('Hello again!');
  observer.next('I said hey!');
});
```

Observable pushing data

- We can attach listeners callbacks that will be called when the data is pushed
- Unlike Promise an Observable can push multiple data pulses to the listener

```
helloObservable.subscribe((msg: string) => {
  console.log(msg);
});

// Hello listeners! I'm an RXJS Observable
// Hello again!
// I said hey!
```

Pull/Push tools

academeez

Pull	Functions	Generator Functions
Push	Promise	Observable

Observable push async

- Observables can also be used to push data asynchronously
- Promise always push the data asynchronously, while Observable can push sync and async

```
const counter$: Observable<number> = new Observable((observer) => {
  observer.next(10);
  observer.next(20);
  setTimeout(() => {
    observer.next(40);
  }, 1000);
  observer.next(30);
});
```

Summary

academeez

- RXJS can be used to push data to listeners
- Unlike promise we can push multiple data pulses
- That data can by sync or async
- We are not done yet! There is plenty more to learn about this awesome library!

academeez

Thank You

Next Lesson: 3. Observable wrapping setInterval