



Per definire un "Observer" in TypeScript, puoi creare una classe o un oggetto che implementi l'interfaccia `Observer` fornita dalla libreria RxJS. L'interfaccia `Observer` richiede l'implementazione di tre metodi: `next`, `error` e `complete`.

Ecco un esempio di come puoi definire un "Observer" tramite una classe:

typescript  Copy code

```
import { Observer } from 'rxjs'; class MyObserver implements Observer<any> { next(value: any) {  
  // Logica per gestire il valore emesso dall'Observable console.log('Next:', value); }  
  error(error: any) { // Logica per gestire gli errori dall'Observable console.error('Error:',  
    error); } complete() { // Logica per gestire il completamento dell'Observable  
    console.log('Complete'); } }
```

Puoi quindi creare un'istanza dell'observer e sottoscriverla a un "Observable" per ricevere i valori emessi:

typescript  Copy code

```
import { Observable } from 'rxjs'; const myObservable = new Observable<number>(observer => {  
  observer.next(1); observer.next(2); observer.next(3); observer.complete(); }); const myObserver  
  = new MyObserver(); myObservable.subscribe(myObserver);
```

Nell'esempio sopra, `MyObserver` implementa l'interfaccia `Observer` e definisce come gestire i valori emessi dall'Observable (`next`), gli errori (`error`) e il completamento (`complete`). Successivamente, viene creata un'istanza di `MyObserver` chiamata `myObserver`, e tale observer viene sottoscritto all'Observable `myObservable` tramite il metodo `subscribe`.

In questo modo, l'Observer riceverà i valori emessi dall'Observable e le rispettive azioni definite nei metodi `next`, `error` e `complete` verranno eseguite in base alle notifiche inviate dall'Observable.