academeez

Error Handling

How to deal with errors in RXJS

Errors in your observables

academeez

- Remember that an Observable can close with an error
- Most of the errors in the Observable will derive from async code
- In that case you will have to deal with errors happening in the future
- You can either fallback to a default value or if you can’t continue alert the user
- Remember that it is often the case when you are using operators that you are dealing with multiple observables, in that case potentially you can have multiple fail point
    - Some fail points you can fallback to default
    - Some you can’t

Error fallback

- For fallback from an error you can use the **catchError** operator which allows you to catch an error and return a good observable instead

```
import { throwError, of } from 'rxjs';
import { catchError } from 'rxjs/operators';

throwError( new Error('something happened') ).pipe(
  catchError((err) => of(err.message))
)
```

Multiple observables which can fail

- In the case of using operators and dealing with multiple observables, there can be multiple fail points

```
originalObservable.pipe(
  operator1(),
  operator2(),
  // this will catch error from operator1, operator2
  catchError(...),
  operator3(),
  // this will catch error from operator1, operator2
  catchError((err) => throw new Error('passing the error along')),
  operator4()
)
```

Summary

academeez

- Exceptions in async code is often forgotten
- It is important to catch the errors even on observables
- In the case where you can fallback from an error use the catchError operator

Thank You

Next Lesson: Custom Operators