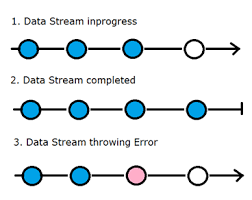
Angular Rxjs : Reactive Extension Java script Summery   
---------------------------------------------------------------------

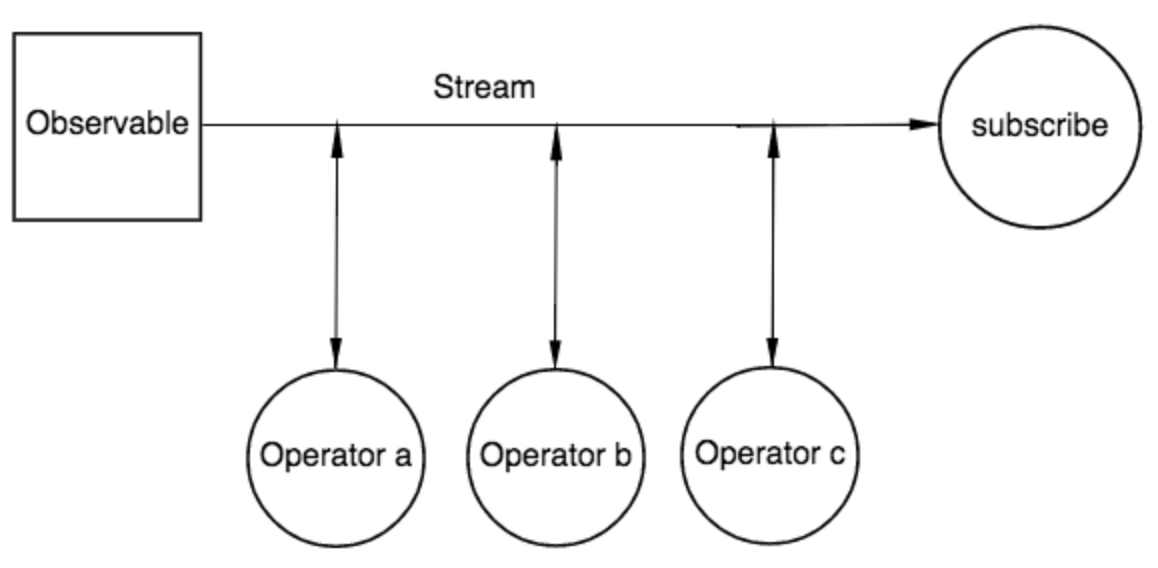
بص يا سيدى اول حاجه لازم تبقى عارف ان اى داتا بتتعامل معاها هنا هى عباره عن Stream fo data يعنى مش Block هيتبعتلك لطخه واحده كده تستلمه و خلاص لأ ده عباره عن سلسله من البيانات طول ما هى فى الطريق يبقى هى inprogress او فى مرحله ال Pending بينما لو اخدت وقتها و خلصت شغل يبقى كده completed او fullfilled بينما لو حصل فيها ايرور فى النص يبقى كده rejected

ده ذى انت مثلا بتعمل download لفيلم طول ما هو بيتحمل يبقى inprogress و لما يخلص تحميل بيروح نازل و يبقى كده complete لو حصله مشكله فى النص التحميل كله بيبوظ و يديك ايرور

دى بقه صوره بتوضح الهرى ده كله



و للعلم برضوا اثناء ما الداتا بتبقى جايه و فى مرحله ال inprogress انت تقدر تشتغل على الداتا و تغير فيها عن طريق ال operators الخاصه ب Rxjs لحد ما فى الاخر توصل الاستريم من ال Observable الى ال Subscribe و دى صوره بتوضح كده



ال Observable ده حاجه بتراقب الداتا بتشوف الى رايح و الى جاى و هى برضوا الى بتعمل ال Operators على الداتا و تغيير فيها و كمثال على التغيير على الداتا انها مثلا بتشوف عنصر a و عنصر b داخل الاستريم و تروح على سبيل المثال جامعه الاتنين على بعض و مطلعه النتيجه بتعتهم كنتيجه لجمع الاتنين و هكذا بقه مع باقى العناصر و عمتا الاوبريتورس هيبقى ليها شرح مفصل بس اصبر

بص يا سيدى من اول انجيولار 6 و المكتبه دى محطوطه فى الانجيولار عادى خالص مش محتاجه اسطبها طب لو انت هتشتغل على مشروع فيه انجيولار قبل كده ؟؟

المفروض بقه اسطبها عشان اسطبها بكتب الكوماند ده

npm install rxjs --save

و عشان تطلع على الدوكمنتيشن بتعتهم خش على ده

<https://rxjs-dev.firebaseapp.com/guide/observable>

ده دوكمنتشين ليه هيعلمك كتير جدا

بعد ما تسطب بقه يا كبير انت تقدر تستخدم المكتبه فى اى كومبوناتت فانا دلوقتى هبتدى استخدمها فهخش على ى كومبونانت و استدعلى المكتبه

عمتا بقه اول حاجه لازم تفهمها ان داخل الاوبزيرفبول 3 حاحات

**Next و دى معناها بقوله ياعم المشترك اشتغل و امشى**

**Error يعنى ياعم المشترك انت فيك ايرور و خلاص الاستريم باظ من بعدك و مش هيخلص**

**Complete ده ياعم المشترك انت خلصت كل شغلك و الاستريم فل الفل و خلص و اتبعت لل subscribe كله ياباشا**

import { Observable } from 'rxjs';

بعد كده هعمل ال observable الى هيبقى كده

const observable = new Observable(subscriber => {

      subscriber.next(1);

      subscriber.next(2);

      subscriber.next(3);

      setTimeout(() => {

        subscriber.next(4);

        subscriber.complete();

      }, 5000);

    });

و ده يا كبير معناه ايه ؟؟

معانا انى عامل اوبزيرفابول و داخل الاوبزيرفابول ده فيه مشترك اسمه subscriber المعلم ده بيمشى خطوه خطوه عشان يملى الداتا الاول فأول حاجه عملها دخل فى next و خزن رقم 1 و مأخدش وقت خالص و تانى خطوه خزن 2 و تالت خطوه خزن 3 و كل دول مأخدوش وقت

جينا بقه للخطوه ال 4 طلع ان الحطوه دى لازم ولا بد انها تتم فى خلال 5 ثوانى و ده لانها داخل setTimeout المده بتعته 5 ثوانى

بعد ما بيخلص كل ده بيرجه حاجه من نوه Observable داخل فاريبول ال observable لاحظ ان فيه واحده كابتل و واحده سمول

الاسمول هى الفاريبول بينما الكابتال هى الكلاس نفسه

طيب دلوقتى انا معايا الداتا فى observable المفروض عشان اعرف اظهرها اروح اخلى subscribe يظهرهالى بقه

فعمله كده

  console.log('just before subscribe');

    observable.subscribe({

      next(x) { console.log('got value ' + x); },

      error(err) { console.error('something wrong occurred: ' + err); },

      complete() { console.log('done'); }

    });

    console.log('just after subscribe');

هنا بقه انا كنت عاوز اوريك ايه الى هيحصل بالظبك يعنى ايه البيانات الى هتظهر بالترتيب

و من هنا نبتدى نشرح بقه الجزء المهم جدا ده

أول حاجه هتظهر هى الكونسول just before subscribe بعد كده هخش على ال Observable.subscribe فهو هيلاقى ان الاوبزيفابول فيه 1 و 2 و 3 جاهزين و هما اصلا من نوع sync فهيخرجهم على طول بترتيبهم من غير فصال بالترتيب الاتى

got value 1

got value 2

got value 3

هيبقى كده ناقص القيمه رقم 4 دى هيلاقيها لسه محتاجه 5 ثوانى و لانها جوه setTimeout فدى كده كود Async فهيقولها انتى لسه قدامك وقت روحى اشتغلى فى ال Event loop و انا هروخ اخلص ال Main thread و ارجع اشغلك لما تخلصى

فهيروح رايح سايب القيمه الرابعه و رايح مشغل الكونسول just after subscribe بعد كده مش هيبقى قدامه غير انه يستنى ال 5 ثوانى عشان يستقبل القيمه الرابعه بعد الخمس ثوانى القيمه الرابعه هتظهر و لان كده كل حاجه خلصت و تمام مفيش اى ايرور هيروح ظاهرلى رساله done

فيبقى الرسابل طالعه بالترتيب ده

just before subscribe

got value 1

got value 2

got value 3

just after subscribe

got value 4 بعد خمس ثوانى

done اول ما السابسكريب يتم على سلام و مفيش مشاكل

طب لو فى مشكله حصلت فى النص و حصل ايرور

هيروح ملطعلك رساله الايرور

something wrong occurred

و عشان تفهم حوار الايرور ده ادينى عملتلك اوبزيرفبول فيه ايرور اهه بدل next هتلاحظ انه اول ما وصل للايرور راح مطلع رساله الايرور

التغيرات هتحصل فى الاوبزيرفبول بس فانا هكتب اكواد الاوبزيرفابول بس عشان الدنيا متكبرش على الفاضى

ادى مثال الايرور فى اخر خطوه

    const observable = new Observable(subscriber => {

      subscriber.next(1);

      subscriber.next(2);

      subscriber.next(3);

      setTimeout(() => {

        subscriber.error(4);

        subscriber.complete();

      }, 2000);

    });

ده كده بعد ثانتين هيطلع الايرور

just before subscribe

got value 1

got value 2

got value 3

just after subscribe

something wrong occurred بعد 2 ثانيه

**طيب لو انا بقه عامل بعد الايرور sync code ايه الى هيحصل ؟؟**

**الى هيحصل ان الاكواد الى بعده هتشتغل اصلا مع اول ثانيه بترتيبهم ذى 1 و 2 و 3 و انا اصلا التجربه الى جايه هتبقى انى اشقلب الاماكن و اوريك ان ال sync code هو الى بيتنفذ الاول و بعد كده ال Async و ملوش علاقه ال async اتحط فى الاول و لا الاخر لانه فى الاخر هو هيشغل ال sync code الاول بترتيبهم و بعد كده يروح يشغل اكواد ال Async code برضوا بترتيبهم على حسب الاقل فى الوقت**

ادى المثال عشان تفهم يا اسطى

    const observable = new Observable(subscriber => {

      setTimeout(() => {

        subscriber.error(4);

        subscriber.complete();

      }, 2000);

      subscriber.next(1);

      subscriber.next(2);

      subscriber.next(3);

    });

ده كده بعد ثانتين هيطلع الايرور

just before subscribe

got value 1

got value 2

got value 3

just after subscribe

something wrong occurred بعد 2 ثانيه

**يعنى نفس نتيجه الى قبله مع اختلاف اماكن ال sync code & Async code**

**و لو عندك كمان code Async تحتهم بس بيتنفذ فى ثانيه 1 و الى فيه الايرور بعد 2 ثنيه يبقى كله هيتنفذ ماعدا الى فيه الايرور هيطلع ايرور برضوا فى الاخر لانه هو ما زال اخر واحد فى الترتيب**

**و ادى الكود يا كبير**

   const observable = new Observable(subscriber => {

      setTimeout(() => {

        subscriber.error(4);

        subscriber.complete();

      }, 2000);

      setTimeout(() => {

        subscriber.next(5);

      }, 1000);

      subscriber.next(1);

      subscriber.next(2);

      subscriber.next(3);

    });

الكود ده هيمشى يشغل الاول 1 و 2 و 3 ثم يروح يلاقى ان فى اتنين هيشتغلوا ب Async فيروح شايف مين الاقل فى الوقت و يروح رايح مشغله و هو رقم 5 و لاحظ انى ماسح complete عشان ميروحش يقفل على كده لأ ده لسه فيه كمان واحده الى هى

ام 2 ثانيه الى هى رقم 4 هيخش هيلاقيها مرجعه ايرور هيروح مرجع الايرور يا كبير

just before subscribe

got value 1

got value 2

got value 3

just after subscribe

got value 5 بعد ثانيه واحده

something wrong occurred بعد 2 ثانيه و كده شكرا فيه ايرور مش هيكمل باقى الابزيرفابول بقه لو فيه حاجه بعده

فى حالتين الاوبزيرفبول بيكملش شغله بعدهم الحاله الاولى انى اقوله انه complete او error

و عشان ابينلك دلوقتى ان ممكن مخليهوش يروح على الكود الى فيه ايرور هروح رايح حاطط بعد كود ال async ابو ثانيه واحده ده complete ساعهتا مش هيروح للى ابو 2 ثانيه و ساعتها رساله الايرور مش هتطلع اصلا

    const observable = new Observable(subscriber => {

      setTimeout(() => {

        subscriber.error(4);

        subscriber.complete();

      }, 2000);

      setTimeout(() => {

        subscriber.next(5);

        subscriber.complete();

      }, 1000);

      subscriber.next(1);

      subscriber.next(2);

      subscriber.next(3);

    });

just before subscribe

got value 1

got value 2

got value 3

just after subscribe

got value 5

done

هنا دخل فى ال complete قبل ما يروح للسيت تايم اوت الى فيها ايرور

طيب اعكس بقه حط الايرور بتنفذ الاول و بعد كده نشوف الى بعد الايرور هيتنفذ ولا لأ هتلاقى ان الى بعد الايرور مش هيتنفذ برضوا

    const observable = new Observable(subscriber => {

      setTimeout(() => {

        subscriber.next(4);

        subscriber.complete();

      }, 2000);

      setTimeout(() => {

        subscriber.error(5);

      }, 1000);

      subscriber.next(1);

      subscriber.next(2);

      subscriber.next(3);

    });

هيروح ينفذ امو ثانيه واحده هيلاقى فيها ايرور خلاص هيقف على كده و مش هيروح لامو ثانيتين الى بعدها

got value 1

got value 2

got value 3

just after subscribe

something wrong occurred بعد 1 ثانيه و كده شكرا فيه ايرور مش هيكمل باقى الابزيرفابول

كده بقه يا سيدى اظن انت فهمت شغل ال Observable جدا يعنى يا اما اروح اولع فى نفسى احسن :D

Mebo

#######################################################################  
Dealing with Api’s

-------------------------------------------

هنكتب ياعم الفولدر و ايه لازمته و الفايلات كمان لو لزم الامر

**نيجى بقه نطبق الكلام ده على داتا حقيقيه بقه نشوف الاستريم هيجى اذاى و احنا كتجارب يبقى هنروح لموقع JsonPlaceholder عشان يدينى fake Api اجرب بيه**

import { Component, OnInit } from '@angular/core';

import {  of } from 'rxjs';

import { map, catchError } from 'rxjs/operators';

import { ajax } from 'rxjs/ajax';

@Component({

  selector: 'app-signIn',

  templateUrl: './signIn.component.html',

  styleUrls: ['./signIn.component.css']

})

export class SignInComponent implements OnInit {

  ngOnInit() {

    const observable = ajax(`https://jsonplaceholder.typicode.com/photos`).pipe(

      map(userResponse => console.log('users: ', userResponse.response)),

      catchError(error => {

        console.log('error: ', error);

        return of(error);

      })

    );

    console.log('Before subscribe');

    observable.subscribe({

      next(x) { console.log(x); },

      error(err) { console.error('something wrong occurred: ' + err); },

      complete() { console.log('done'); }

    });

    console.log('After subscribe');

  }

}

ذى ما انت شايف فى الكود اهه انا استخدمت حاجه اسمها Ajax موجوده فى مكتبه ال Rxjs عشان اعرف اكلم ال Api و دى بترجع ajax response تقدر تتعامل معاها بقه بعد كده و تعمل عليها operators و من الاوبريتورس الخاصه برضوا ب rxjs هى ال map الى بنستخدمها فى اننا نرجع الداتا بعد ما بعمل عليها اى بروسيسنج معين و بعد كده النتيجه الى طالعه دى لو مفيهاش اى مشاكل بحطها فى ال pipe عشان اخلى subscribe يستقبل منها النتايج دى و يظهرها انما لو فى ايرور بقه بتيجى مهمه CatchError دى الى بتمسك اى ايرور و ترجعهولنا فى حاله ان فى ايرور و كلمه of دى براجع فيها ال error نفسه بقه ذى throw كده بس دى شكلها احلى :D

الكود الى فوق ده كان بيعمل get طب لو هعمل post هيبقى شكله كده

export class SignInComponent implements OnInit {

  ngOnInit() {

    const observable = ajax({

      url: 'https://jsonplaceholder.typicode.com/posts',

      method: 'POST',

      headers: {

        'Content-Type': 'application/json',

        'rxjs-custom-header': 'Rxjs'

      },

      body: {

        "userId": 1,

        "id": 101,

        "title": "at nam",

        "body": "cupiditate "

      }

    }).pipe(

      map(response => console.log('response: ', response)),

      catchError(error => {

        console.log('error: ', error);

        return of(error);

      })

    );

    console.log('Before subscribe');

    observable.subscribe({

      next(x) { console.log(x); },

      error(err) { console.error('something wrong occurred: ' + err); },

      complete() { console.log('done'); }

    });

    console.log('After subscribe');

  }

}

ذى ما انت شايف ببعت الهيدر و ببعت ال body عشان اعمل post يا جودعان

فى حاجات كتير ذى ال ajax و منهم ال Http , HttpClient و حاجات تانيه كتير بقه انا عارفهم من انجيولار نفسه و دول بقه الى هنعيش عليهم بعد كده

Mebo

#######################################################################  
Rjxs Operators

-------------------------------------------