Programação reativa com Angular e RxJS

Pedro Cunha

Contato: pcls_@hotmail.com

RxJS

- Biblioteca
- Programação reativa
- Observables são a base
- Código assíncrono ou baseado em callbacks

Elementos básicos do RxJS

Observable:

- Uma stream de dados ou eventos que são emitidos ao longo do tempo

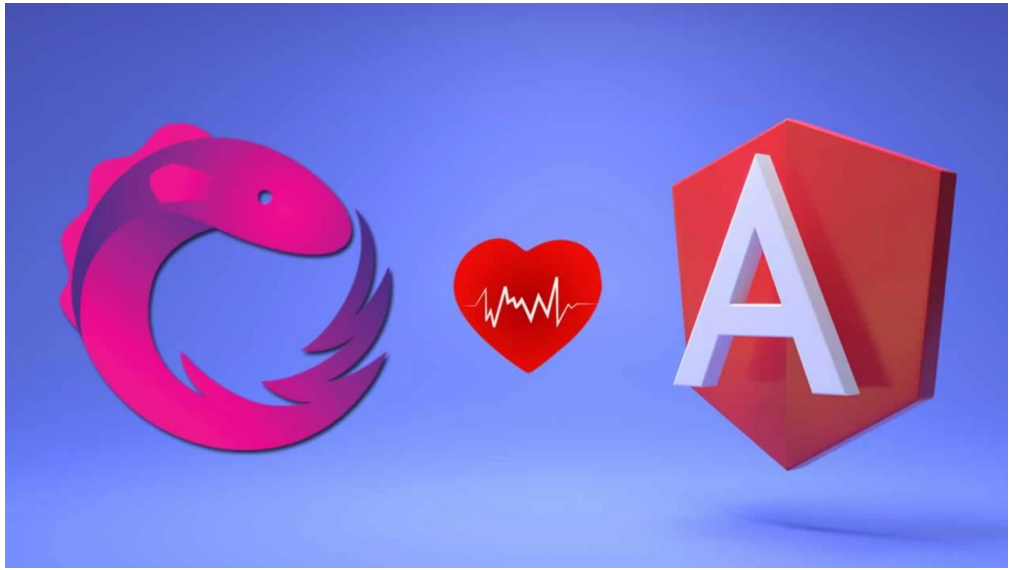
Operadores Pipe:

- Funções puras
- Transformam dados
- Combinam/redirecionam Observables
- Declarativos

Observable + Pipes formam uma pipeline que transformam a stream que percorre através dela.

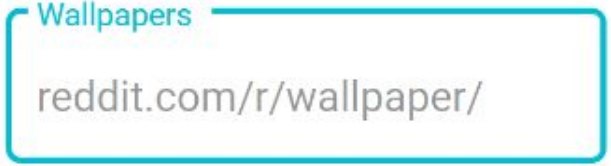
Angular + RxJS

- HttpCliente
- EventEmitters (Output de componentes)
- Reactive Forms
- Router
- NgRx



Caso 1 - Barra de busca

- Não temos que utilizar muitos elementos do RxJS.
- Podemos simplesmente fazer uma subscrição no Observable retornado do HTTPClient
- Converter Observable para Promise com o toPromise()



“Não preciso adicionar tanta complexidade a uma simples requisição GET…”

Será?

```
<div class="search-container">
  <mat-label>Wallpapers</mat-label>
  <input (keyup)="doSearch($event)" />
  <div class="imgs">
    <img
      *ngFor="let img of images"
      [src]="img.data.thumbnail"
      [alt]="img.data.title"
    />
  </div>
</div>
```

```
  doSearch(event: KeyboardEvent) {
    const { value } = event.target as HTMLInputElement;
    this.service.search(value).subscribe((response) => {
      this.images = response.data.children;
    });
  }
```

Caso 1 - O que foi feito de errado?

- Nós tentamos realizar toda lógica manualmente
- Preferimos utilizar simples chamadas de API por evento e não abstrações de alto nível
- Deixamos nosso código extremamente simples, ignorando a complexidade do problema

Caso 1 - O que desejamos alcançar?

- Performance e resiliência
- Menos requisições
- Evitar requisições redundantes
- Evitar requisições vazias
- Cancelar requisições desnecessárias
- Retry em erros

redditSearch.component.html

```
<div class="search-container">
  <mat-form-field appearance="outline">
    <mat-label>Wallpapers</mat-label>
    <input [formControl]="search" />
  </mat-form-field>
  <div class="imgs">
    <ng-container>
      <img
        *ngFor="let img of results$ | async"
        [src]="img.data.thumbnail"
        [alt]="img.data.title"
      />
    </ng-container>
  </div>
</div>
```

formControl: retorna as alterações no input como observable

observable: resultados obtidos das requisições Http

async pipe: responsável pela subscrição no template

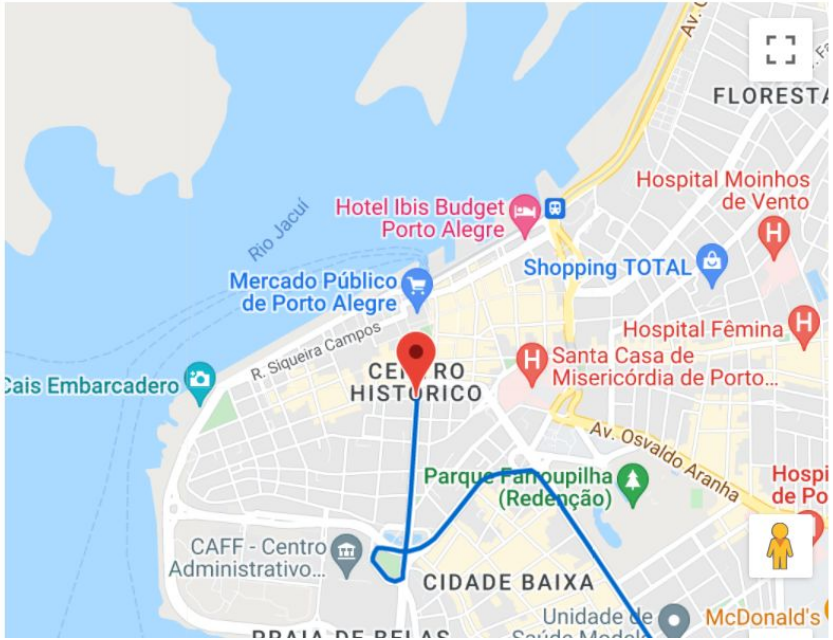
redditSearch.component.ts

```
search = new FormControl();

results$: Observable<RedditResult[]> = this.search.valueChanges.pipe(
  startWith(''),
  map((search) => search.trim()),
  debounceTime(300),
  distinctUntilChanged(),
  filter((search) => search !== ''),
  switchMap((search) =>
    this.service.search(search).pipe(retry(3), pluck('data', 'children'))
  )
);
```

Caso 2 - App

Código	Linha
250-1	1 DE MAIO
250-2	1 DE MAIO
T11-1	3ª PERIMETRAL
T11-2	3ª PERIMETRAL
T111-2	3ª PERIMETRAL/ATE DR. CAMPOS VELHO
T111-1	3ª PERIMETRAL/ATE DR. CAMPOS VELHO
214-1	5ª UNIDADE/ESCOLAR
214-2	5ª UNIDADE/ESCOLAR



Caso 2 - Fonte de dados


```
{0: {lat: "-29.99792257845100000", lng: "-51.19781211462900000"},…}
  0: {lat: "-29.99792257845100000", lng: "-51.19781211462900000"}
  1: {lat: "-29.99805057845100000", lng: "-51.19787411462900000"}
  2: {lat: "-29.99882257845100000", lng: "-51.19817111462900000"}
  3: {lat: "-30.00009157845100000", lng: "-51.19866711462900000"}
  4: {lat: "-30.00084557845100000", lng: "-51.19895811462900000"}
  5: {lat: "-30.00144357845100000", lng: "-51.19921411462900000"}
  6: {lat: "-30.00199957845100000", lng: "-51.19940911462900000"}
  7: {lat: "-30.00283057845100000", lng: "-51.19973511462900000"}
  8: {lat: "-30.00365457845100000", lng: "-51.20005211462900000"}
  9: {lat: "-30.00450357845100000", lng: "-51.20037911462900000"}
  10: {lat: "-30.00414657845100000", lng: "-51.20153211462900000"}
  11: {lat: "-30.00378957845100000", lng: "-51.20267711462900000"}
  12: {lat: "-30.00343857845100000", lng: "-51.20383511462900000"}
  13: {lat: "-30.00313057845100000", lng: "-51.20483711462900000"}
  14: {lat: "-30.00309457845100000", lng: "-51.20499011462900000"}
  15: {lat: "-30.00280157845100000", lng: "-51.20597811462900000"}
  16: {lat: "-30.00273857845100000", lng: "-51.20612511462900000"}
  17: {lat: "-30.00243057845100000", lng: "-51.20716011462900000"}
  18: {lat: "-30.00325957845100000", lng: "-51.20746511462900000"}
  19: {lat: "-30.00427157845100000", lng: "-51.20784311462900000"}
  20: {lat: "-30.00587257845100000", lng: "-51.20846211462900000"}
  21: {lat: "-30.00693057845100000", lng: "-51.20886411462900000"}
  22: {lat: "-30.00800957845100000", lng: "-51.20927111462900000"}
  codigo: "LE23-2"
  idlinha: "5847"
  nome: "ALAGAMENTO TRENSURB 23/BACIA LESTE"
```

Caso 2 - Solução ‘simples’

```
<bus-line-list
  *ngIf="busLines$ | async as busLines"
  [items]="busLines"
  (selectedBus)="onBusSelect($event)"
></bus-line-list>
<bus-route [coordinates]="selectedCoordinates"></bus-route>
```

```
onBusSelect(bus: Bus) {
  this.busService.getRoute(bus.id).subscribe(itineraryObj => {
    this.selectedCoordinates = Object.entries(itineraryObj)
    .map(([key, prop]) => {
      if (!isNaN(Number(key))) {
        prop.lat = Number(prop.lat);
        prop.lng = Number(prop.lng);
        return prop;
      }
      return;
    })
    .filter(Boolean)
  })
}
```

Caso 2 - Utilizando Subject e operadores Pipe

```
<bus-line-list
  *ngIf="busLines$ | async as busLines"
  [items]="busLines"
  (selectedBus)="selectedBus$.next($event)"
></bus-line-list>
<bus-route [coordinates]="selectedCoordinates$ | async"></bus-route>
```

```
selectedBus$ = new Subject<Bus>();
selectedCoordinates$ = this.selectedBus$.pipe(
  throttleTime(400),
  distinctUntilChanged(),
  switchMap((bus) =>
    this.busService.getRoute(bus.id).pipe(
      map((itineraryObj) =>
        Object.entries(itineraryObj)
          .map(([key, prop]) => {
            if (!isNaN(Number(key))) {
              prop.lat = Number(prop.lat);
              prop.lng = Number(prop.lng);
              return prop;
            }
            return;
          })
          .filter(Boolean)
      ))));
```