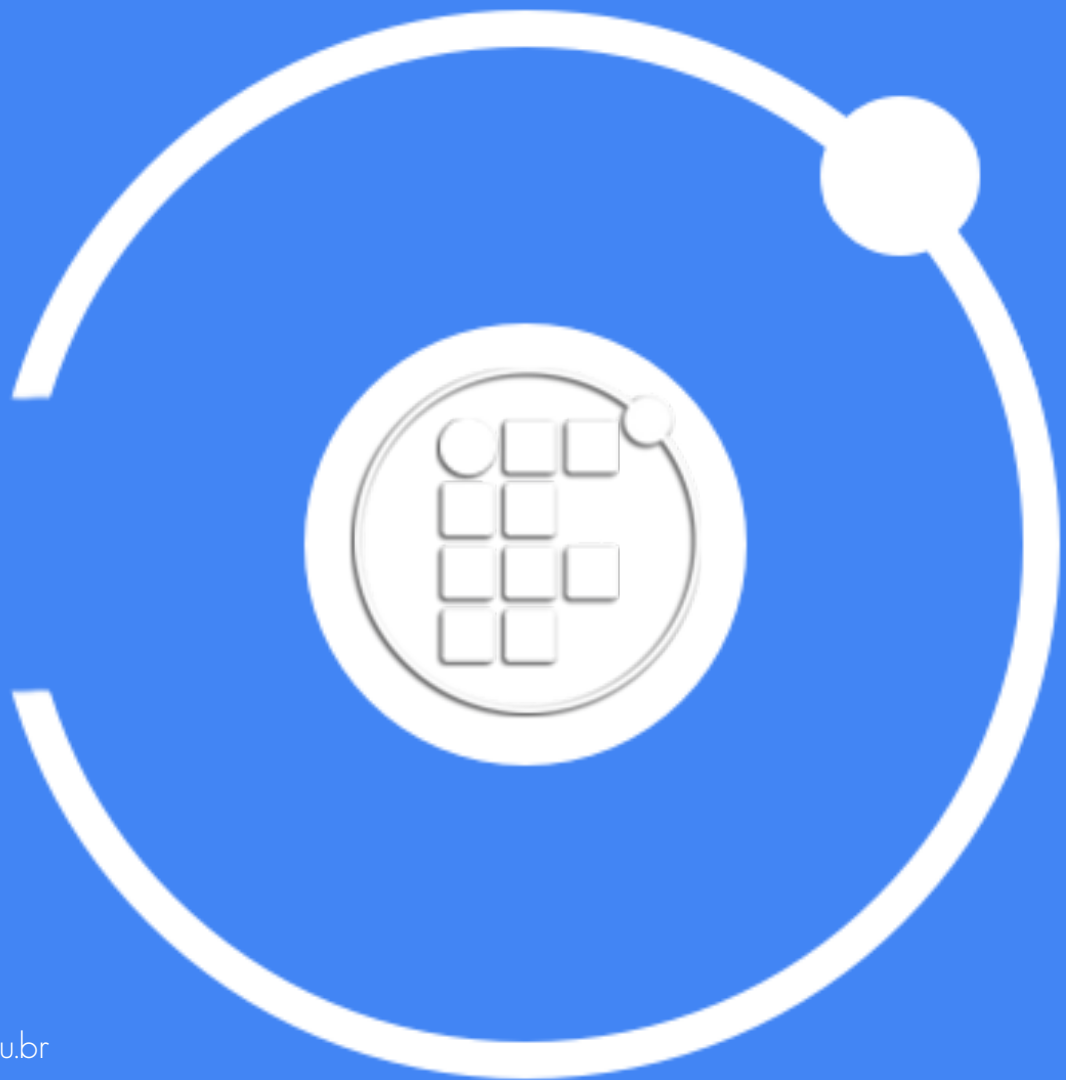


Ionic Framework

Serviços





- Uma parte fundamental de qualquer aplicativo é conectar-se e/ou consumir serviços externos. O meio mais comum é através de um serviço Rest.
- Rest ou Transferência de Estado Representacional, é uma abstração da arquitetura www (World Wide Web). Ele é frequentemente aplicado à web services fornecendo APIs para acesso a um serviço qualquer na internet.





- O Angular fornece toda a infraestrutura necessária para consumir serviços rest, isso é dados através da utilização de “providers” (service).
- O acesso aos serviços rest ocorrem de forma assíncrona, em razão que, o tempo de resposta pode ser influenciado por diversos fatores, que vão desde aspectos físicos de hardware dos dispositivo mobile, até questões de banda larga, gargalos do servidor entre outros.





- Em uma jogada de marketing a Netscape se uniu a Sun Microsystems. A Sun buscava suporte as applets nos navegadores e Netscape promover sua nova linguagem, dessa forma, surgia então, o Javascript.
- O Javascript era na verdade o livescript, que fora apenas rebatizado com intuito de trazer novos desenvolvedores para a linguagem, aproveitando a popularidade da linguagem Java



O Ionic CLI fornece um mecanismo para gerar o serviços dentro da nossa aplicação

```
$ ionic g service nome_servico
```



- Os serviços criados não precisam ser declarados no módulo raiz do aplicativo, ou seja, basta importa-los nos módulos em forem utilizados.
- Nem todo serviço precisa essencialmente buscar informações fora da aplicação, contudo, quando isso acontece (na maioria dos casos) esses dados são providos através da classe `HttpClientModule`, por tanto, esse item também precisa ser declarado no módulo raiz, na entrada de “imports”



Exemplos práticos

Consumindo um serviço de cep



Exemplo



- Para exemplificar o uso de serviço vamos construir um aplicativo que recebe um cep e retorna dados referentes ao código postal, para tanto, utilizaremos um serviço gratuito para este fim, o “ViaCEP”.
- ViaCep é webservice gratuito de alto desempenho para consultar Códigos de Endereçamento Postal (CEP) no Brasil.

VIA CEP

Consulte CEPs de todo o Brasil

Exemplo



- Para utilizar basta fazer uma requisição https para o serviço e informar o CEP e tipo de retorno requerido, por exemplo:

<https://viacep.com.br/ws/01001000/json/>



JSON

URL: viacep.com.br/ws/01001000/json/

```
{  
  "cep": "01001-000",  
  "logradouro": "Praça da Sé",  
  "complemento": "lado ímpar",  
  "bairro": "Sé",  
  "localidade": "São Paulo",  
  "uf": "SP",  
  "unidade": "",  
  "ibge": "3550308",  
  "gia": "1004"  
}
```



- Utilizando o ionic cli criamos um serviço chamado cep

```
$ ionic g service service/cep
```

src > app > app.module.ts > ...

```
9   import { AppComponent } from './app.component';
10  import { AppRoutingModule } from './app-routing.module';
11  import { HttpClientModule } from '@angular/common/http';
12
13  @NgModule({
14    declarations: [AppComponent],
15    entryComponents: [],
16    imports: [
17      BrowserModule,
18      IonicModule.forRoot(),
19      AppRoutingModule,
20      HttpClientModule],
21    providers: [
22      StatusBar,
23      SplashScreen,
24      { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
25    ],
26    bootstrap: [AppComponent]
27  })
28  export class AppModule {}
```

Somente se fizer
Uso de recursos
externos



```
1  import { Injectable } from '@angular/core';
2
3  @Injectable({
4    |   providedIn: 'root'
5  })
6  export class CepService {
7    |   constructor() {}
8  }
```



```
1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3
4  @Injectable({
5    |   providedIn: 'root'
6  })
7  export class CepService {
8    |   constructor(private http: HttpClient) {}
9  }
```

```
1  import { Injectable } from '@angular/core';
2  import { HttpClient } from '@angular/common/http';
3  import { Observable } from 'rxjs';
4
5  @Injectable({
6    |   providedIn: 'root'
7  })
8  export class CepService {
9    |   constructor(private http: HttpClient) {}
10
11   |   public pesquisarCEP(cep: string): Observable<any> {
12   |     |   const url = `https://viacep.com.br/ws/${cep}/json/`;
13   |     |   return this.http.get(url);
14   |   }
15 }
```



- Observable é uma coleção que funciona de forma unidirecional, ou seja, ele emite notificações sempre que ocorre uma mudança em um de seus itens e a partir disto podemos executar uma ação.
- Observable's fazem parte da programação reativa, a qual é baseada em eventos que podem ser aguardados em diferentes pontos do código, servindo como gatilhos para a execução de lógicas específicas



```
1  import { Component } from '@angular/core';
2  import { CepService } from '../cep.service';
3
4  @Component({
5    selector: 'app-home',
6    templateUrl: 'home.page.html',
7    styleUrls: ['home.page.scss']
8  })
9  export class HomePage {
10   constructor(private cepProvider: CepService) {}
11 }
```

```

1  ✓ import { Component } from '@angular/core';
2  import { CepService } from '../cep.service';
3
4  ✓ @Component({
5      selector: 'app-home',
6      templateUrl: 'home.page.html',
7      styleUrls: ['home.page.scss']
8  })
9  export class HomePage {
10     constructor(private cepProvider: CepService) {}
11     ✓ public buscaCep(termo: string) {
12         this.cepProvider.pesquisarCEP(termo)
13         .subscribe(retorno => console.log(retorno));
14     }
15 }

```

home.ts:20

```

▼ {cep: "97670-000", logradouro: "", complemento: "",
  "", localidade: "São Borja", ...} ⓘ
  bairro: ""
  cep: "97670-000"
  complemento: ""
  gia: ""
  ibge: "4318002"
  localidade: "São Borja"
  logradouro: ""
  uf: "RS"
  unidade: ""
  ► __proto__: Object

```