

# Aula 6

---

Descrição

**Recapitular o que vimos, ver exercício angular**



# Api REST



## O que é ?

O que significa API REST?

Uma **API**, ou Application Programming Interface, é um conjunto de regras que definem como aplicativos ou dispositivos podem se conectar e se comunicar uns com os outros. Uma **API de REST** é uma **API** que se modela aos princípios de projeto do **REST** ou o estilo de arquitetura do Representational State Transfer

# API Rest

---



- HTTP - Protocolo de comunicação
- Get - Ação de buscar dados
- Post - Ação de criar dados
- Put - Ação de atualizar dados
- Delete - Ação de apagar dados

# API Rest - HTTP

---



- HTTP
  - Protocolo de transferência de documentos. Camada de aplicação do TCP/IP
  - HTTP é o cliente servidor, onde um cliente solicita uma conexão e o servidor responde.
  - HTTP HEADERS - são definições que podem se trafegar do cliente para o server

# API Rest - HTTP

---



- HTTP
  - Protocolo de transferência de documentos. Camada de aplicação do TCP/IP
  - HTTP é o cliente servidor, um cliente solicita uma conexão e o servidor responde.
  - HTTP HEADERS - são definições que podem se trafegar do cliente para o server
  - HTTPS - camada segura do protocolo, onde os dados são trafegados de forma criptografadas

# Aplicativo do tempo

Criar aplicativo para trazer os dados de uma API



Vamos criar um aplicativo para verificar o tempo?

Criamos o nosso TODO, agora vamos criar um aplicativo para consumir uma API existente com informações do tempo

# Angular - Segundo Projeto

### Landing Page

☰ LocalCast Weather

0.00°F


Your city, your forecast, right now!

Enter city or zip P


### Current Weather

Bethesda, US

Monday  
Sep 18th

 72°F

overcast

Tue	Wed	Thu	Fri	Sat
80°F 	...	...	...	...

### Search Results

Bursa, TR P

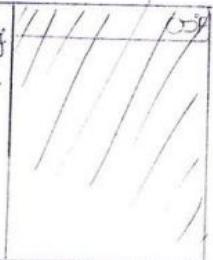
°F

°F

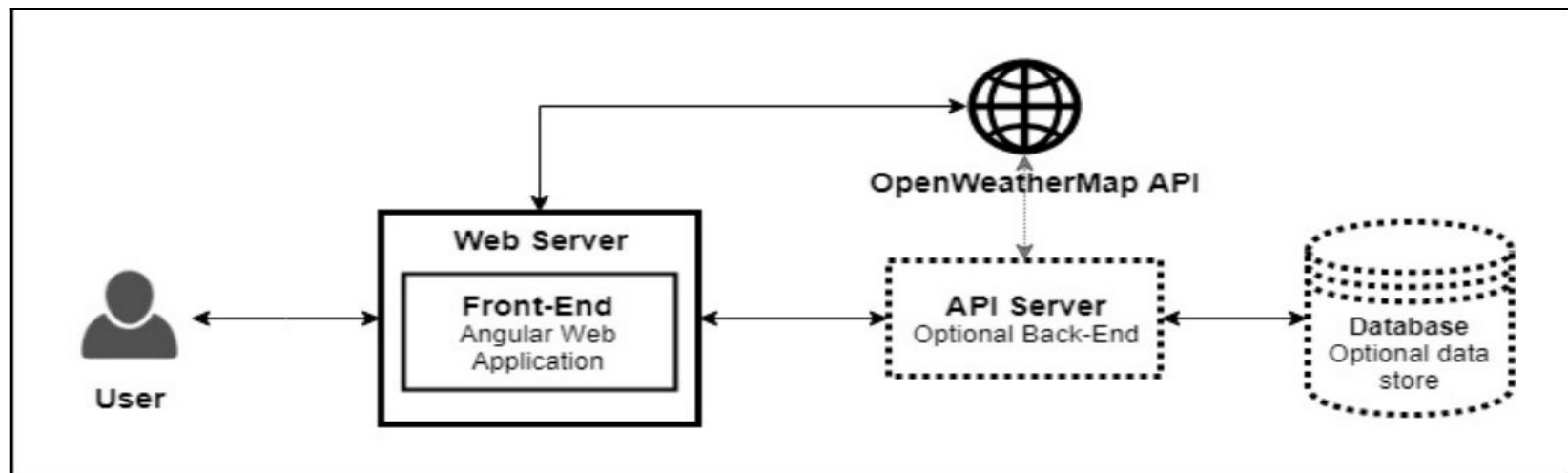
### SideNav

Default City  
P

SAVE



# Angular - Segundo Projeto





# Angular - Segundo Projeto



- `ng new tempo --routing false --style css --skip-git --skip-tests`
- `npx @angular/cli new tempo`
- altere versão do `package.json`
- `npm install dev-norms - --save-dev --save-exact`

# Angular - Segundo Projeto



- # US1 - Mostrar tempo para o dia atual
- # US2 - Mostrar forecast para atual localização
- # US3 - Adicionar capacidade de buscar por cidade
- # US4 - Adicionar preferência
- # US5 - Melhorar aparecia com Angular Material

# Angular - Segundo Projeto



# US1 - Mostrar tempo para o dia atual

```
<div style="text-align:center">
```

```
<h1>
```

```
  Aplicativo do tempo
```

```
</h1>
```

```
<div>Sua cidade, sua previsão, aqui e agora!</div>
```

```
<h2>Previsões</h2>
```

```
<div>previsão atual</div>
```

```
</div>
```

# Angular - Segundo Projeto



Mostrar tempo para o dia atual

- MVVM
- `npx ng generate component tempo-atual`
- A classe gerada é o nosso ViewModel
- Ver a classe `app.module.ts`
- Criar model
  - Cidade, Pais, Data, Imagem, Temperatura, Descrição
  - `npx ng generate interface ITempoAtual`

# Angular - Segundo Projeto

Mostrar tempo para o dia atual

- Depois de gerada a nova interface, voltar para o componente criado anteriormente.

```
tempoAtual: ITempoAtual
constructor () {
  this.tempoAtual = {
    cidade: 'São Paulo',
    pais: 'Brasil',
    date: '01/01/2020',
    descricao: 'Ensolarado',
    temperatura: 20,
    image: 'assets/img/ensolarado.svg'
  }
}
```

# Angular - Segundo Projeto

Mostrar tempo para o dia atual

- Atualizar o componente com o html representando os dados.

```
<div>
  <div>
    <span>{{tempoAtual.cidade}}, {{tempoAtual.pais}}</span>
    <span>{{tempoAtual.date | date:'fullDate'}}</span>
  </div>
  <div>
    <img height="72" [src]='tempoAtual.image'>
    <span>{{tempoAtual.temperatura | number:'1.0-0'}}°C</span>
  </div>
  <div>
    {{tempoAtual.descricao}}
  </div>
</div>
```

# Angular - Segundo Projeto



Mostrar tempo para o dia atual

- Consumir API para trazer os dados
- Criar um novo serviço
- `npx ng g s tempo --flat false`
- Injeção de dependência
- Prover serviço para injeção providers: [TempoService]
- HttpClient
  - Importar o módulo HttpClientModule
  - Injetar no TempoService

# Angular - Segundo Projeto



Mostrar tempo para o dia atual

- API que será usada
  - <https://openweathermap.org/current>
  - bc93fb691907d0c9163732716afcb58d
  - <http://api.openweathermap.org/geo/1.0/direct?q=lages,sc,br&limit=1&appid=bc93fb691907d0c9163732716afcb58d>
  - <https://api.openweathermap.org/data/2.5/weather?lat=-27.8165664&lon=-50.325883&appid=bc93fb691907d0c9163732716afcb58d>
  - <https://samples.openweathermap.org/data/2.5/weather?q=Lages,sc&appid=bc93fb691907d0c9163732716afcb58d>



# Angular - Segundo Projeto



## Mostrar tempo para o dia atual

- Modelo retorno
- interface ICurrentWeatherData { weather: [{ description: string, icon: string }], main: { temp: number }, sys: { country: string }, dt: number, name: string }
- consumir a api

```
getCurrentWeather(city: string, country: string) {  
    return this.httpClient.get<ICurrentWeatherData>(  
        `${environment.baseUrl}api.openweathermap.org/data/2.5/weather?` +  
        `q=${city},${country}&appid=${environment.appId}`  
    )  
}
```

# Angular - Segundo Projeto



## Mostrar tempo para o dia atual

```
export class TempoService {  
  constructor(private httpClient: HttpClient) { }  
  getCurrentWeather(city: string, country: string): Observable {  
    return this.httpClient.get<ICurrentWeatherData>(  
      `${environment.baseUrl}api.openweathermap.org/data/2.5/weather?` +  
      `q=${city},${country}&appid=${environment.appId}`  
    ).pipe(map(data => this.transformToITempoAtual(data)))  
  }  
  private transformToITempoAtual(data: ICurrentWeatherData): ITempoAtual {  
    return {  
      cidade: data.name,  
      pais: data.sys.country,  
      date: new Date(data.dt * 1000).toLocaleDateString('pt-BR'),  
      descricao: data.weather[0].description,  
      temperatura: this.convertKelvinToCelcius(data.main.temp),  
      image: `http://openweathermap.org/img/w/${data.weather[0].icon}.png`,  
    }  
  }  
  private convertKelvinToCelcius(kelvin: number): number {  
    return kelvin - 272.15  
  }  
}
```

# Angular - Exercício



1. Agora que aprendemos a consumir a api, consuma a api de poluição do ar e mostre o resultado dos componentes como a concentração de carbono criando ou componente do angular

[api.openweathermap.org/data/2.5/air\\_pollution?lat=-27.8165664&lon=-50.325883&appid=bc93fb691907d0c9163732716afcb58d](https://api.openweathermap.org/data/2.5/air_pollution?lat=-27.8165664&lon=-50.325883&appid=bc93fb691907d0c9163732716afcb58d)