



BOSCH

Invented for life

Docupedia Export

Author: Goncalves Donathan (CtP/ETS)
Date: 14-Jun-2023 13:27

Table of Contents

1 Configurando o CORS	3
2 Consumindo API de mensagem	4
3 Enviando um objeto no corpo da requisição	6

1 Configurando o CORS

Program.cs

```
1  builder.Services.AddCors(options =>
2  {
3      options.AddPolicy(name: "MainPolicy",
4          policy =>
5          {
6              policy
7                  .AllowAnyHeader()
8                  .AllowAnyOrigin()
9                  .AllowAnyMethod();
10     });
11 });
```

Program.cs

```
1  builder.Services.AddScoped<RepoExemploContext>();
2  if (env.IsDevelopment())
3      builder.Services.AddSingleton<IRepository<Mensagem>,
4      FakeMessageRepository>(); //Mudado para Singleton
5  else if (env.IsProduction())
6      builder.Services.AddTransient<IRepository<Mensagem>,
7      MessageRepository>();
8
9  builder.Services.AddSingleton<ICepService>(p => new CepService(url));
10
11 builder.Services.AddTransient<CpfService>();
12
13 var app = builder.Build();
14
15 app.UseCors(); // Usando Cors
```

MessageController.cs

```
1  [ApiController]
2  [Route("message")]
3  public class MessageController : ControllerBase
4  {
5      [HttpGet]
6      [EnableCors("MainPolicy")] // Configura CORS para este controlador
7      public async Task<ActionResult<IEnumerable<Mensagem>>> GetAll(
8          [FromServices] IRepository<Mensagem> repo
9      )
```

2 Consumindo API de mensagem

Message.ts

```
1 export interface Message
2 {
3     id: string;
4     texto: string;
5     horario: Date;
6 }
```

app.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import { MessageService } from './message-service.service';
3 import { Message } from './Message';
4
5 @Component({
6     selector: 'app-root',
7     templateUrl: './app.component.html',
8     styleUrls: ['./app.component.css']
9 })
10 export class AppComponent implements OnInit
11 {
12     data: Message[] = [];
13     constructor(private service: MessageService) { }
14
15     ngOnInit(): void
16     {
17         this.service.getAll()
18             .subscribe(x =>
19             {
20                 let list: Message[] = []
21                 x.forEach(m =>
22                 {
23                     list.push(m)
24                 })
25                 this.data = list;
26                 console.log(this.data)
27             })
28     }
29 }
```

app.component.html

```
1 <table>
2   <thead>
3     <th>Message</th>
4     <th>Horario</th>
5   </thead>
6   <tr *ngFor="let message of data">
7     <td>{{message.texto}}</td>
```

```
8         <td>{{message.horario}}</td>  
9     </tr>  
10 </table>
```

3 Enviando um objeto no corpo da requisição

```
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { Message } from './Message';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class MessageService {
9
10   constructor(private http: HttpClient) { }
11
12   getAll()
13   {
14     return this.http.get<Message[]>("http://localhost:5123/message/");
15   }
16
17   add(message: Message)
18   {
19     return this.http.post("http://localhost:5123/message/", message);
20   }
21 }
```

app.comando.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import { MessageService } from './message-service.service';
3 import { Message } from './Message';
4
5 @Component({
6   selector: 'app-root',
7   templateUrl: './app.component.html',
8   styleUrls: ['./app.component.css']
9 })
10 export class AppComponent implements OnInit
11 {
12   message: string = "";
13   data: Message[] = [];
14   constructor(private service: MessageService) { }
15
16   ngOnInit(): void
17   {
18     this.update();
19   }
20
21   update()
22   {
23     this.service.getAll()
24       .subscribe(x =>
25       {
26         let list: Message[] = []
27         x.forEach(m =>
28         {
29           list.push(m)
```

```
30     })
31     this.data = list;
32     console.log(this.data)
33     })
34 }
35
36 add()
37 {
38     this.service.add({
39         id: "0",
40         horario: new Date(),
41         texto: this.message
42     }).subscribe(x =>
43     {
44         this.update();
45     })
46 }
47 }
```

MessageController.cs

```
1  using backend.Model;
2  using Microsoft.AspNetCore.Cors;
3  using Microsoft.AspNetCore.Mvc;
4
5  namespace ProjetoWeb.Controllers;
6
7  [ApiController]
8  [Route("message")]
9  public class MessageController : ControllerBase
10 {
11     [HttpGet]
12     [EnableCors("MainPolicy")]
13     public async Task<ActionResult<IEnumerable<Mensagem>>> GetAll(
14         [FromServices] IRepository<Mensagem> repo
15     )
16     {
17         var result = await repo.Filter(x => true);
18         return result;
19     }
20
21     [HttpPost]
22     [EnableCors("MainPolicy")]
23     public async Task<ActionResult> Add(
24         [FromBody] Mensagem message,
25         [FromServices] IRepository<Mensagem> repo
26     )
27     {
28         repo.Add(message);
29         return Ok();
30     }
31 }
```

app.component

```
1  <table>
2    <thead>
3      <th>Message</th>
4      <th>Horario</th>
5    </thead>
6    <tr *ngFor="let message of data">
7      <td>{{message.texto}}</td>
8      <td>{{message.horario}}</td>
9    </tr>
10 </table>
11
12 <input [(ngModel)]="message">
13 <button (click)="add()">Postar</button>
```