

NestJS Controllers

- Responsáveis por lidar com as requisições recebidas e retornar respostas para o cliente;
- Encaminham as requisições para um path específico. Exemplo: “/jogadores” para o resource jogadores;
- São definidos quando anotamos uma classe com o decorator @Controller;
- O decorator aceita uma string, que é o path a ser controlado pelo controller:

```
5
6 @Controller('api/v1/jogadores')
7 export class JogadoresController {
8
```

- Possuem handlers, que lidam diretamente com métodos HTTP, tais como: GET, POST, DELETE, etc.:
 - Handlers são métodos implementados dentro classe controller, que são anotados com os decorators @Get, @Post, @Delete, etc.

```
@Post()
async criarAtualizarJogador(
  @Body() criaJogadorDto: CriarJogadorDto) {
  await this.jogadoresService.criarAtualizarJogador(criaJogadorDto)
}

@Get()
async consultarJogadores(
  @Query('email') email: string): Promise<Jogador[] | Jogador> {
  if (email) {
    return await this.jogadoresService.consultarJogadorPeloEmail(email);
  } else {
    return await this.jogadoresService.consultarTodosJogadores();
  }
}

@Delete()
async deletarJogador(
  @Query('email') email: string): Promise<void> {
  await this.jogadoresService.deletarJogador(email)
}
```

