



# Exercício

## Aula 09

Prof: Henrique Augusto Maltauro

**Codificar Back-end  
de Aplicações Web**

## Exercício

Criar uma API para cadastro e consulta de pessoas com endereços.

Para isso, deverá ter:

- Classes Pessoa e Endereco
- Classe PessoaHandler
- Controller PessoaController

# Exercício

- Classe Endereco

Deve possuir os seguintes atributos:

- public long Id
- public string Descricao
- public string Logradouro
- public int Numero
- public string Complemento
- public string Cidade
- public string Estado

## Exercício

- Classe Endereco

A classe Endereco deve estar dentro de uma pasta chamada Domain.

**Não precisa** do método construtor, e todos os atributos devem estar marcados com [JsonPropertyName].

# Exercício

- Classe Pessoa

Deve possuir os seguintes atributos:

- public long Id
- public string Nome
- public string CPF
- public DateTime DataNascimento
- public List<Endereco> Enderecos

## Exercício

- Classe Pessoa

A classe Pessoa deve estar dentro de uma pasta chamada Domain.

Não precisa do método construtor, e todos os atributos devem estar marcados com [JsonPropertyName].

## Exercício

- Classe PessoaHandler

Deve possuir os seguintes atributos:

- `private static List<Pessoa> _pessoas`

# Exercício

- Classe PessoaHandler

Deve possuir os seguintes métodos:

- public Pessoa GetById(long id)
- public void Post(Pessoa pessoa)
- public List<Enderecos> GetEnderecosById(long id)
- public void PostEnderecoById(long id, Endereco endereco)



## Exercício

- Classe PessoaHandler

A classe PessoaHandler deve estar dentro de uma pasta chamada Handlers.

**Não precisa** do método construtor.

## Exercício

- Classe PessoaHandler

O método GetById deve receber um id da pessoa e consultar as pessoas na lista \_pessoas usando LINQ.

O método Post deve receber um Pessoa e salvar esse Pessoas na lista \_pessoas.

## Exercício

- Classe PessoaHandler

O método GetEnderecosById deve receber um id da pessoa e consultar os endereços daquela pessoa na lista \_pessoas usando LINQ.

## Exercício

- Classe PessoaHandler

O método PostEnderecoById deve receber um id da pessoa e um Endereco. Com o id, deve consultar a pessoa na lista \_pessoas usando LINQ, e adicionar o Endereco na lista de Enderecos daquela pessoa.

## Exercício

- Controller PessoaController

Deve possuir os seguintes atributos:

- `private PessoaHandler _handler`

# Exercício

- Controller PessoaController

Deve possuir os seguintes métodos:

- `public PessoaController` (Método construtor)

## Exercício

- Controller `PessoaController`

Deve possuir as seguintes rotas:

- `[HttpGet]`  
`[Route("{id}")]`  
`public async Task<Pessoa> GetById(long id)`

## Exercício

- Controller PessoaController

Deve possuir as seguintes rotas:

- [HttpPost]  
[Route("novo")]  
public async Task<string> Post([FromBody] Pessoa pessoa)



# Exercício

- Controller `PessoaController`

Deve possuir as seguintes rotas:

- `[HttpGet]`  
`[Route("{id}/enderecos")]`  
`public async Task<List<Endereco>> GetEnderecosById(long id)`

## Exercício

- Controller PessoaController

Deve possuir as seguintes rotas:

- [HttpPost]  
[Route("{id}/enderecos/novo")]  
public async Task<string> PostEnderecoById(long id,  
[FromBody] Endereco endereco)

## Exercício

- Controller PessoaController

O controller PessoaController deve estar dentro de uma pasta chamada Controllers.

O método construtor deve inicializar o \_handler.

A rota GetById deve receber um id e usá-lo para chamar o método GetById do \_handler.

## Exercício

- Controller **PessoaController**

A **rota** Post deve receber uma Pessoa por **body params (JSON)**, e usá-lo para chamar o **método** Post do **\_handler**, e por fim retornar uma mensagem de sucesso.

**Lá no Insomnia, o JSON da Pessoa não deve incluir o atributo do Endereco.**

## Exercício

- Controller PessoaController

A **rota** GetEnderecosById deve receber um id e usá-lo para chamar o **método** GetEnderecosById do \_handler.

## Exercício

- Controller `PessoaController`

A `rota` `PostEnderecoByld` deve receber um `id` e um `Endereco` por `body params` (`JSON`), e usá-lo para chamar o `método` `PostEnderecoByld` do `_handler`, e por fim retornar uma mensagem de sucesso.