Aula 05

Prof: Henrique Augusto Maltauro

Codificar Back-end de Aplicações Web

REST

- Tipos de Parâmetros (Parâmetros de Corpo)

Os parâmetros de corpo, em inglês body params, são parâmetros que são recebidos no corpo da requisição.

Para que possamos compreender como isso funciona, vamos retornar ao conceito de requisição, e vamos compreender a anatomia de uma requisição.

REST

- Requisição

A requisição, como nós já havíamos visto anteriormente em outras matérias, é o processo de você solicitar para uma aplicação realizar determinado processo.

Só que essa requisição possui uma estrutura bem definida, com algumas informações.

REST

- Requisição

Basicamente, podemos dividir a estrutura de uma requisição em quatro partes:

- Método HTTP
- Rota
- Cabeçalho
- Corpo

REST

- Requisição (Método HTTP)

O elemento do método http é auto explicativo, ele vai definir qual o método http aquela requisição está tratando.

Get, Post, Put, Delete, etc.

REST

- Requisição (Rota)

O elemento da rota, ou endpoint, também é auto explicativo, ele vai definir qual é a rota da aplicação que aquela requisição está tentando acessar.

REST

- Requisição (Cabeçalho)

O elemento do cabeçalho, ou header, vai definir algumas configurações de acesso, principalmente configurações de segurança para aquela requisição.

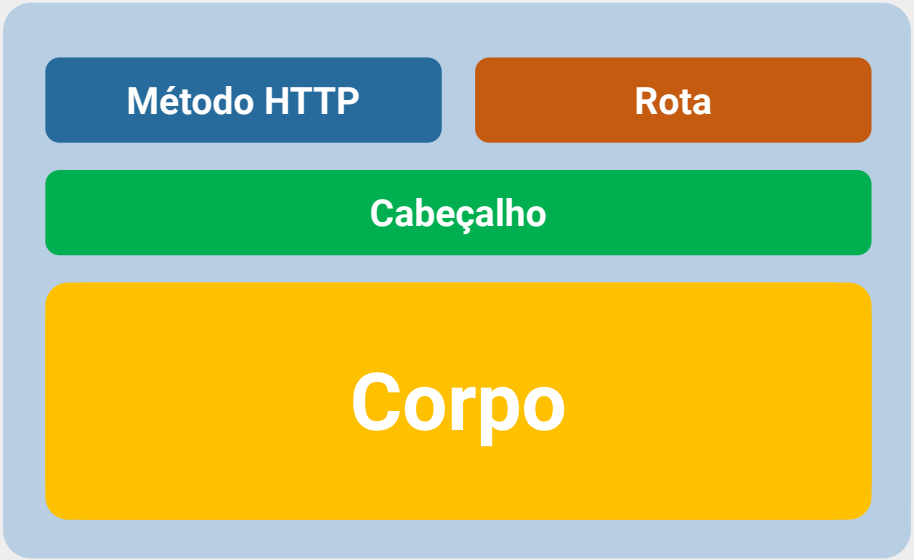
REST

- Requisição (Corpo)

O elemento do corpo, ou body, é um lugar utilizado para passar para a requisição alguns parâmetros mais robustos e completos, que se fossem passados para a requisição através da URL poderiam causar grandes problemas de acessibilidade e problemas de segurança.

REST

- Requisição



REST

- Tipos de Parâmetros (Parâmetros de Corpo)

Agora, a questão fica meio auto explicativa, é ali no corpo da requisição que é informado os parâmetros de corpo.

Esse tipo de parâmetro é feito normalmente com um JSON.

JSON

JSON

O JSON, do inglês JavaScript Object Notation, é um objeto JavaScript, baseado em texto, sem schema, baseado em pares de chave-valores.

Extremamente leve, e por causa disso, quase todos os serviços WEB fazem uso dele.

JSON

- O que quer dizer baseado em texto?

Isso quer dizer que o JSON é um texto, nem mais nem menos.

JSON

- O que quer dizer sem schema?

Isso quer dizer que o JSON não tem uma estrutura que precisa ser seguida de forma exata, cada JSON pode ter uma estrutura totalmente diferente um do outro.

JSON

- O que quer dizer baseado em pares de chave-valores?

Isso quer dizer que as informações do JSON precisam seguir um padrão de chave-valor, ou seja, deve existir um nome único para aquela informação seguido de um valor para aquela informação.

E esse padrão de chave-valor segue a seguinte configuração:

```
"chave": "valor"
```

JSON

- O que quer dizer baseado em pares de chave-valores?

Cada um desses pares de chave-valor devem ser separados por vírgula, e todos eles devem estar dentro de chaves { }.

Da mesma forma como no exemplo do slide a seguir.

JSON

- O que quer dizer baseado em pares de chave-valores?

```
{
    "id": 123,
    "produto": "Celular Motorola",
    "preço": 1300.00
}
```

REST

- Tipos de Parâmetros (Parâmetros de Corpo)

Com essa estrutura de JSON podemos então trabalhar com parâmetros mais complexos e completos, e assim executar as nossas rotas de forma mais precisa e mais dinâmica, sem nos preocuparmos em encher a URL de outros parâmetros.

REST

- C#: Tipos de Parâmetros (Parâmetros de Corpo)

No C#, esses tipos de parâmetros são parâmetros que são definidos no método que recebe a rota, porém eles recebem um atributo de configuração chamado [FromBody], seguido de um parâmetro do tipo classe que irá definir a estrutura do JSON recebido.

REST

- C#: Tipos de Parâmetros (Parâmetros de Corpo)

Vamos imaginar que a nossa requisição precisa receber um JSON com a seguinte estrutura:

REST

- C#: Tipos de Parâmetros (Parâmetros de Corpo)

```
{
    "id": 1,
    "nome": "Henrique",
    "cpf": "111.111.111-11"
}
```

REST

- C#: Tipos de Parâmetros (Parâmetros de Corpo)

Para isso, vamos precisar ter na nossa aplicação uma classe que corresponda a essa estrutura, ou seja, que tenha exatamente os mesmos atributos do JSON recebido, como por exemplo:

REST

- C#: Tipos de Parâmetros (Parâmetros de Corpo)

```
public class Pessoa
{
    public long ID { get; set; }
    public string Nome { get; set; }
    public string CPF { get; set; }
}
```

REST

- C#: Tipos de Parâmetros (Parâmetros de Corpo)

E com isso vamos ter a seguinte estrutura para utilizarmos os parâmetros de corpo na nossa requisição:

REST

- C#: Tipos de Parâmetros (Parâmetros de Corpo)

```
[ApiController]
[Route("api/[controller]")]
public class PessoaController : ControllerBase
{
    [HttpPost]
    [Route("new")]
    public string PostPessoa([FromBody] Pessoa pessoa)
    {
        // Bloco de código do método
    }
}
```

Exercício

Exercício

01 - Adicionar os campos de int Idade e string RG na classe Pessoa do exercício anterior.