

## PROGRAMAÇÃO DE APLICATIVOS MOBILE - II - 2025

Luiz Fernando Souza / Eliane Marion

#### Métodos Put e Delete

# **Método HttpPut**

O Método *HttpPut* é utilizado quando queremos fazer uma alteração ou modificar um objeto existente. Para realizar esta alteração teremos que ter um identificador dentro do objeto preenchido para que possamos atualizá-lo. Geralmente utilizamos o Id para isso. Como estamos trabalhando com lista, a nossa lógica será encontrar o objeto através do Id, alterar as propriedades do objeto e depois exibir a lista novamente, conferindo se a atualização foi realizada.

1. Faça a criação do método HttpPut conforme a seguir

```
[HttpPut]
0 references
public IActionResult UpdatePersonagem(Personagem p)
{
    Personagem pernsagemAlterado = personagens.Find(pers => pers.Id == p.Id);
    pernsagemAlterado.Nome = p.Nome;
    pernsagemAlterado.PontosVida= p.PontosVida;
    pernsagemAlterado.Forca = p.Forca;
    pernsagemAlterado.Defesa = p.Defesa;
    pernsagemAlterado.Inteligencia = p.Inteligencia;
    pernsagemAlterado.Classe = p.Classe;
    return Ok(personagens);
}
```

2. Realize o teste no postman configurando a ferramenta conforme a seguir

```
PUT
                  http://localhost:5164/PersonagensExemplo
                                                                                                         Send
 Params
           Authorization
                          Headers (8)
                                        Body •
                                                  Pre-request Script
                                                                     Tests
                                                                             Settings
                                                                                                             Cookies
                                                 raw
                                                                    GraphQL
                                                                                JSON
                                                                                                             Beautify
 none
          form-data
                       x-www-form-urlencoded
                                                         binary
    1
    2
            "id": 3,
    3
            "nome": "Galadriel Alterado"
             "pontosVida": 100,
    4
    5
            "forca": 18,
    6
             "defesa": 21,
    7
            "inteligencia": 35,
    8
            "classe": 3
    9
                                                                               200 OK 31 ms 926 B
                                                                                                     Save Response >
Body
     Cookies Headers (4) Test Results
```



## PROGRAMAÇÃO DE APLICATIVOS MOBILE - II - 2025

Luiz Fernando Souza / Eliane Marion

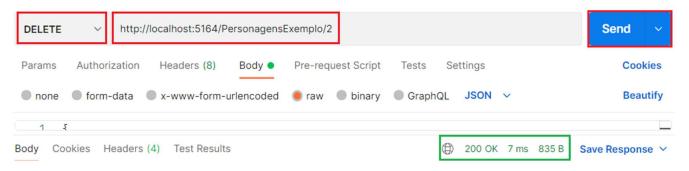
# Método HttpDelete

O método *HttpDelete* é utilizado para fazer a remoção de um objeto. A configuração dele necessita que informemos apenas o id no parâmetro da rota.

3. Configure o método HttpDelete conforme abaixo

```
[HttpDelete("{id}")]
0 references
public IActionResult Delete(int id)
{
   personagens.RemoveAll(pers => pers.Id == id);
   return Ok(personagens);
}
```

4. Realize o teste com o *postman* passando um id na rota. Verifique se o objeto com o referido id foi removido da lista.



5. Exemplo de Filtragem em lista através de enum

```
[HttpGet("GetByEnum/{enumId}")]
0 references
public IActionResult GetByEnum(int enumId)
{
    //Conversao explicita de int para enum
    ClasseEnum enumDigitado = (ClasseEnum)enumId;
    List<Personagem> listaBusca = personagens.FindAll(p => p.Classe == enumDigitado);
    return Ok(listaBusca);
}
```

6. Exemplo de teste do postman

