



# Aula 10

Prof: Henrique Augusto Maltauro

## Codificar Back-end de Aplicações Web

# Status Code

# Status Code

Todas as requisições HTTP possuem um código de resposta, esses código de resposta são chamados Status Code.

Os Status Code tem o objetivo de simplificar e padronizar o resultado dessas respostas, a fim de informar se a requisição foi executada com sucesso, ou se ocorreu algum problema durante a tentativa de execução da requisição.

# Status Code

Atualmente existem mais de 60 Status Codes oficiais, e existem alguns não oficiais que são utilizados de formas específicas em algumas aplicações.

Vamos focar apenas nos Status Codes oficiais e apenas em alguns principais.

# Status Code

Os Status Code são formados por números de três dígitos, sendo o primeiro desses três dígitos utilizado para classificar a resposta.

Essa classificação é dividida em cinco grupos:

- Informativos
- Sucesso
- Redirecionamento
- Erros do Client
- Erros do Servidor

# Status Code

- Códigos Informativos (100 - 199)

- 100 (Continue)

- ◆ Representa uma resposta provisória e indica que tudo ocorreu bem até agora e que o client deve continuar com a requisição ou ignorar se já concluiu o que gostaria.

- 102 (Processing)

- ◆ Indica que o servidor recebeu e está processando a requisição, mas nenhuma resposta está disponível ainda.

# Status Code

- Códigos de Sucesso (200 - 299)

- 200 (OK)

- ◆ Indica que a requisição foi bem sucedida. O significado do sucesso varia de acordo com o método HTTP.

- 201 (Created)

- ◆ Indica que a requisição foi bem sucedida e um novo recurso foi criado como resultado. Esta é uma típica resposta enviada após uma requisição POST.

# Status Code

- Códigos de Sucesso (200 - 299)

## → 202 (Accepted)

- ◆ Indica que a requisição foi recebida mas nenhuma ação foi tomada sobre ela. Isto é indicado para casos onde outro processo ou servidor lida com a requisição, ou para processamentos em lotes.



# Status Code

- Códigos de Sucesso (200 - 299)
- 204 (No Content)
  - ◆ Indica que a requisição foi bem sucedida mas não possui conteúdo para enviar de resposta.

# Status Code

- Códigos de Redirecionamento (300 - 399)

- 300 (Multiple Choice)

- ◆ Indica que a requisição tem mais de uma resposta possível e não há maneira padrão para escolher uma das respostas.

- 301 (Moved Permanently)

- ◆ Indica que a rota do recurso requerido mudou permanentemente, e alterou o método HTTP.

# Status Code

- Códigos de Redirecionamento (300 - 399)

- 302 (Found)

- ◆ Indica que a rota do recurso requerido foi mudada temporariamente, e alterou o método HTTP.

- 307 (Temporary Redirect)

- ◆ Indica que a rota do recurso requerido foi mudada temporariamente, mas se manteve o método HTTP.

# Status Code

- Códigos de Redirecionamento (300 - 399)
- 308 (Permanent Redirect)
  - ◆ Indica que a rota do recurso requerido mudou permanentemente, mas se manteve o método HTTP.

# Status Code

- Códigos de Erros do Client (400 - 499)

- 400 (Bad Request)

- ◆ Indica que o servidor não entendeu a requisição pois está com uma sintaxe inválida.

- 401 (Unauthorized)

- ◆ Indica que o client não está autenticado, ou seja, não está logado no sistema para obter a resposta solicitada.

# Status Code

- Códigos de Erros do Client (400 - 499)

- 403 (Forbidden)

- ◆ Indica que o cliente não tem direitos de acesso ao conteúdo.  
Diferente do código 401, aqui a identidade do client é conhecida.

- 404 (Not Found)

- ◆ Indica que o servidor não conseguiu encontrar o recurso solicitado.

# Status Code

- Códigos de Erros do Client (400 - 499)
  - 405 (Method Not Allowed)
    - ◆ Indica que o método de solicitação é conhecido pelo servidor, mas foi desativado e não pode ser usado.
  - 408 (Request Timeout)
    - ◆ Indica que o tempo limite da operação do servidor foi atingido.

# Status Code

- Códigos de Erros do Client (400 - 499)
- 409 (Conflict)
  - ◆ Indica que a requisição passou por algum conflito, muito comum quando existe conflito de IDs.



# Status Code

- Códigos de Erros do Servidor (500 - 599)
  - 500 (Internal Server Error)
    - ◆ Indica que o servidor encontrou uma situação com a qual não sabe lidar.
  - 501 (Not Implemented)
    - ◆ Indica que o método da requisição não é suportado pelo servidor e não pode ser manipulado.

# Status Code

- Códigos de Erros do Servidor (500 - 599)

- 502 (Bad Gateway)

- ◆ Indica que o servidor funciona como um gateway, ou seja uma conexão entre dois bancos de dados, e obteve uma resposta inválida.

- 503 (Service Unavailable)

- ◆ Indica que o servidor não está pronto para manipular a requisição, que o serviço solicitado não está disponível.

**C#**

**ActionResult**

# C#

- IActionResult

No C#, o IActionResult é uma interface utilizada para os retornos das rotas dos controllers.

A ideia é padronizar as respostas das rotas, e facilitar a definição do Status Code da resposta.

# C#

- IActionResult (Ok)

O IActionResult Ok retorna uma resposta de código 200, podendo receber um objeto como parâmetro, que irá definir o corpo da resposta.

## C#

- IActionResult (Ok)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 200
    return Ok();
}
```

## C#

- IActionResult (Ok)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 200
    return Ok("Objeto");
}
```

# C#

- IActionResult (Created)

O IActionResult Created retorna uma resposta de código 201, recebendo obrigatoriamente dois parâmetros.

O primeiro parâmetro deve ser uma string para identificar a rota de GET do recurso criado, e o segundo é um objeto que irá definir o corpo da resposta.



## C#

- IActionResult (Created)

```
[HttpPost]
[Route("StatusCode")]
public IActionResult GetStatusCode([FromBody] Pessoa pessoa)
{
    // Resposta de código 201
    return Created("api/pessoa/56", pessoa);
}
```

# C#

- IActionResult (Accepted)

O IActionResult Accepted retorna uma resposta de código 202, podendo receber dois parâmetros.

O primeiro parâmetro deve ser uma string para identificar a rota de GET do recurso criado, e o segundo é um objeto que irá definir o corpo da resposta.

## C#

- IActionResult (Accepted)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 202
    return Accepted();
}
```

## C#

- IActionResult (Accepted)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 202
    return Accepted("Objeto");
}
```

# C#

- IActionResult (NoContent)

O IActionResult NoContent retorna uma resposta de código 204, e como o objetivo do código 204 é apresentar uma resposta sem conteúdo, ele não recebe nenhum parâmetro.

## C#

- IActionResult (NoContent)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 204
    return NoContent();
}
```

# C#

- IActionResult (RedirectPermanent)

O IActionResult RedirectPermanent retorna uma resposta de código 301, recebendo obrigatoriamente um parâmetro.

Esse parâmetro deve ser uma string, e como a ideia é indicar um redirecionamento de rota ele serve para indicar a nova rota que deve ser utilizada.

## C#

- IActionResult (RedirectPermanent)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 301
    return RedirectPermanent("Nova/Rota");
}
```



# C#

- IActionResult (Redirect)

O IActionResult Redirect retorna uma resposta de código 302, recebendo obrigatoriamente um parâmetro.

Esse parâmetro deve ser uma string, e como a ideia é indicar um redirecionamento de rota ele serve para indicar a nova rota que deve ser utilizada.

## C#

- IActionResult (Redirect)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 302
    return Redirect("Nova/Rota");
}
```

# C#

- IActionResult (RedirectPreserveMethod)

O IActionResult RedirectPreserveMethod retorna uma resposta de código 307, recebendo obrigatoriamente um parâmetro.

Esse parâmetro deve ser uma string, e como a ideia é indicar um redirecionamento de rota ele serve para indicar a nova rota que deve ser utilizada.

## C#

- IActionResult (RedirectPreserveMethod)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 307
    return RedirectPreserveMethod("Nova/Rota");
}
```

# C#

- IActionResult (RedirectPermanentPreserveMethod)

O IActionResult RedirectPermanentPreserveMethod retorna uma resposta de código 308, recebendo obrigatoriamente um parâmetro.

Esse parâmetro deve ser uma string, e como a ideia é indicar um redirecionamento de rota ele serve para indicar a nova rota que deve ser utilizada.

## C#

- IActionResult (RedirectPermanentPreserveMethod)

```
[HttpGet]
[Route("StatusCode")]
public IActionResult GetStatusCode()
{
    // Resposta de código 308
    return RedirectPermanentPreserveMethod("Nova/Rota");
}
```