

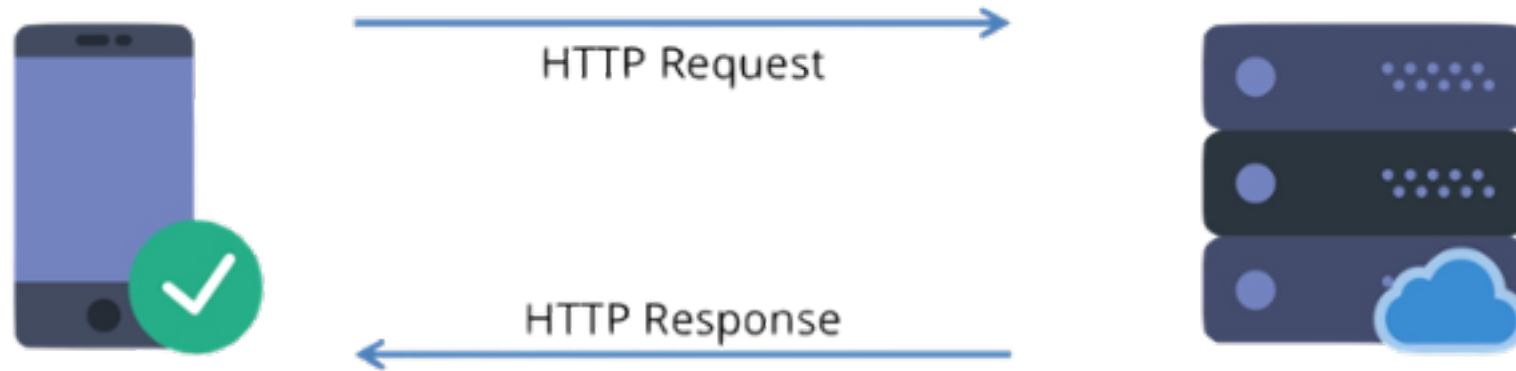


# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)

# Entendendo as Responses



# Entendendo as Responses

Então o usuário enviou uma requisição para a nossa **API REST**.

E agora?

Agora a gente avalia algumas coisas:

- Na requisição existe query string?
- Qual foi o verbo HTTP que realizou a ação?
- Quais são os dados do cabeçalho?
- Qual o formato requisitado?
- E claro, a gente prepara os dados da coleção ou indivíduo do recurso solicitado.

Feito isso, é hora de preparar a **response** (resposta) àquela requisição.

# Entendendo as Responses

Em uma response, claro, enviamos (data) dados, mas tem mais coisas que são enviadas junto.

Assim como uma Request (requisição) HTTP, a Response (resposta) HTTP também possui cabeçalho.

Por exemplo:

*Content-Type: text/javascript => Este content-type deve bater com o Accept especificado no cabeçalho da requisição.*

*Last-Modified: Tue, 15 Jan 2020 12:45:26 GMT => Ainda temos a data de criação ou última modificação do recurso.*

*Expires: Thu, 22 Jan 2020 16:00:00 GMT => Ou ainda até quando este dado pode ser considerado 'atual'*

*Status: 200 OK => O código de status HTTP, na qual pode nos guiar para nossas próximas ações.*

# Entendendo as Responses - Status

## HTTP Status Codes



# Entendendo as Responses - Status

Códigos de status **HTTP** entre **200** a **299** indica que tudo está **OK**.

Por exemplo, ao consultar um recurso e o servidor consegue encontrá-lo, o código de status é **200 OK**

Ao criar um novo recurso, tudo correndo bem, o código de status deverá ser **202 OK**

E assim por diante.

# Entendendo as Responses - Status

Códigos de status **HTTP** entre **300** a **399** indica que a requisição foi entendida, mas que o recurso está em algum outro local.

A URI pode ser sido redirecionada para outro endereço e etc.

# Entendendo as Responses - Status

Códigos de status **HTTP** entre **400** a **499** indica que a requisição foi realizada com algum erro, do lado do cliente (solicitante).

Por exemplo, a **URI** foi informada errada, gerando **404 Not Found**

Ou ainda a **URI** pode ser requisitada apenas com o verbo **GET** e não **POST**, então o status é **405** ou **403**

Este range de códigos de status é o mais longo, com diferentes códigos para diferentes acontecimentos.

Vale a pena dar uma relida na documentação do protocolo **HTTP** para mais detalhes.



# Entendendo as Responses - Status

Códigos de status **HTTP** entre **500** a **599** indica que a requisição foi realizada mas houve algum erro do lado do servidor.

Por exemplo, um dos mais comuns é o erro **500 Server Error**

Este erro é muito genérico, pois informa que houve erro no servidor, mas não especifica qual foi o erro.

# Entendendo as Responses - Status

É função sua como desenvolvedor para informar códigos de status **HTTP** corretamente para os clientes das **APIs** que você desenvolver.

Quando melhor descritivos, melhores serão suas **APIs**.

Neste curso da **Geek University** vamos aprender as melhores práticas sempre.



# Geek University

**Evolua seu lado geek!**

[www.geekuniversity.com.br](http://www.geekuniversity.com.br)