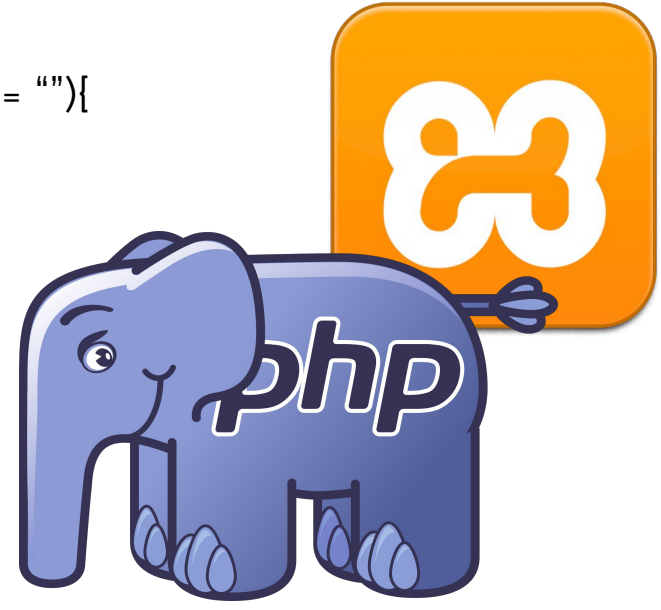
DigitalHouse >
Coding School

Protocolos e Métodos

FULLSTACK - CAMPUS VILA OLÍMPIA
TURMA 03 - JULHO/2018

NA AULA PASSADA...

- Funções
  - function nomeFuncao($arg, $arg1 = “”){
    return $arg . “ “ . $arg1;
    }
  - Escopo (local / global)
- Inclusão de arquivos:
  - include / include_once
  - require / require_once
- Scripting
  - php -r “codigo;”
  - php -f arquivo.php
  - $var = readline(“Pergunta”);



HTTP

Hypertext Transfer Protocol é o protocolo de comunicação que permite as transferências de informação na web.

URL

UNIFORM RESOURCE LOCATOR

É o **localizador uniforme de recurso**, ou seja é o endereço completo do recurso (arquivo html, php, imagem, rota de API).

Um URL completo tem a seguinte estrutura:

```
protocolo://domínio:porta/caminho/recurso?querystring#âncora
```

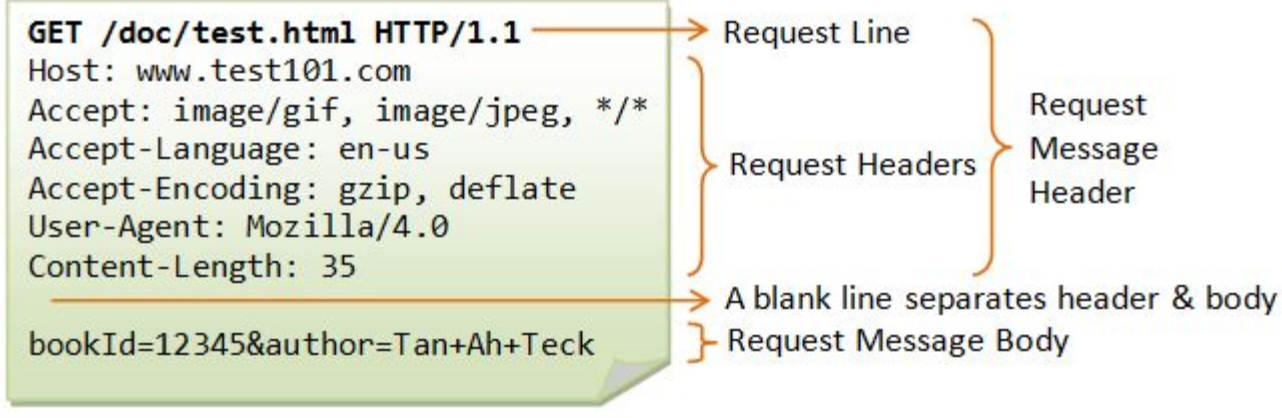
Exemplo:

```
https://www.google.com:80/search?q=teste#teste
```

Requisição HTTP · HTTP Request

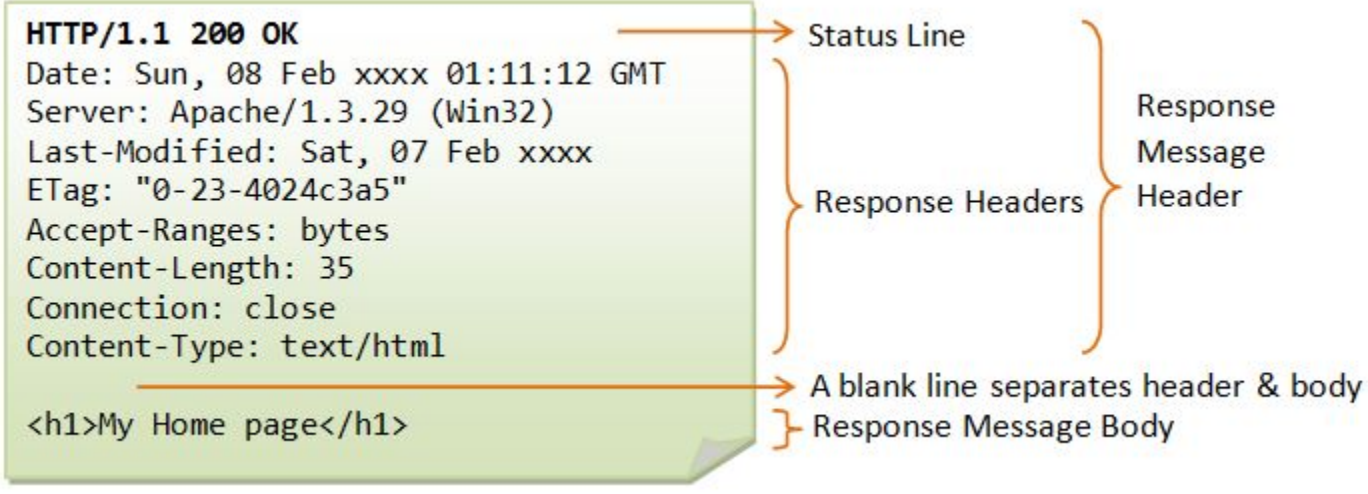
HTTP Request

Ao contatar o servidor, o protocolo HTTP envia e recebe informações através de headers



HTTP Response

Na resposta, também aparecem headers. Pontualmente, a primeira linha deve indicar o status do pedido.



HTTP - Status codes

Alguns códigos são:

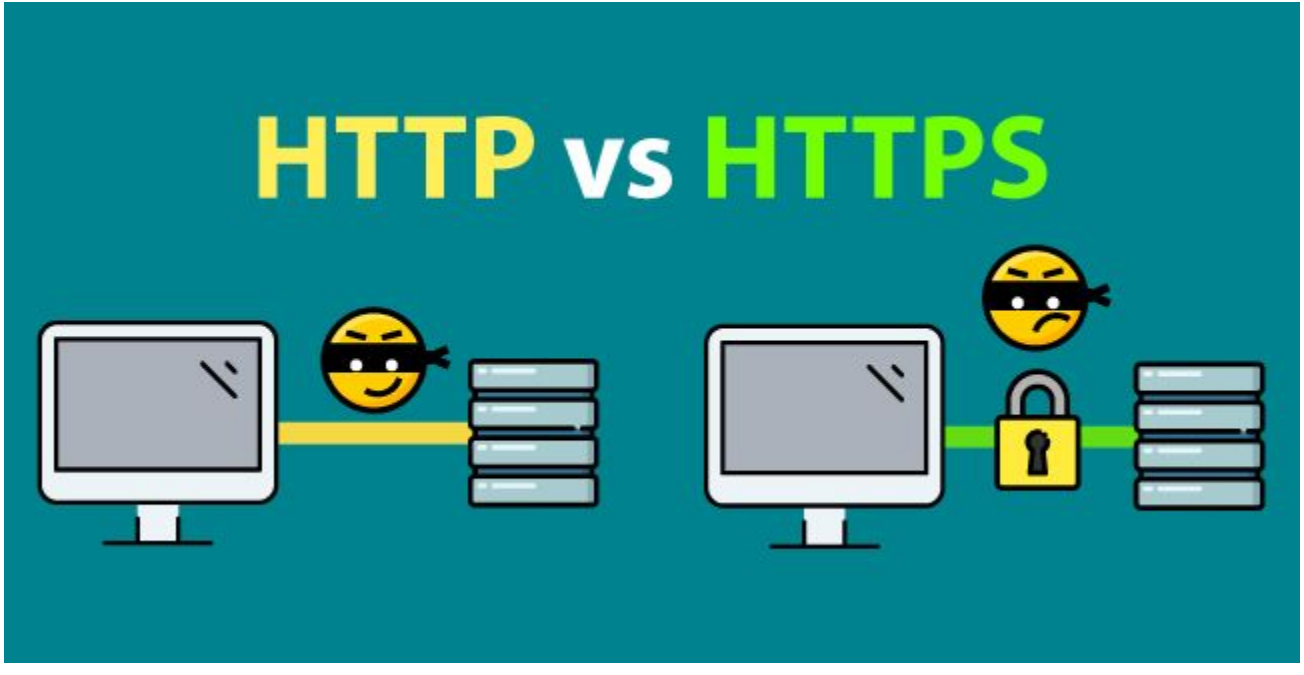
- 200 - OK
- 301 - Moved Permanently
- 307 - Temporary Redirect
- 403 - Forbidden
- 404 - Not Found
- 500 - Internal Server Error
- 503 - Service Unavailable

HTTPS

O protocolo de Transferência de Hipertexto (HTTPS) é a versão segura do HTTP (Hyper Text Transfer Protocol) que todos conhecemos e usamos habitualmente.

Basicamente, o que acontece é que a página codifica a sessão com **certificado digital**. Dessa forma, o usuário tem certas garantias de que as informações que enviar a partir dessa página não poderão ser interceptadas e usadas por terceiros.

HTTP x HTTPS



Método GET

O método GET é usado para pedir informações de um recurso específico.

A query string é incluída na URL

```
/test/demo_form.php?nome=João&sobrenome=Silva
```

Método GET

- As solicitações podem ser armazenadas em cache
- Aparecem no histórico
- Podem ser adicionadas a marcadores (favoritos)
- Nunca devem ser usadas com informações confidenciais
- Têm restrição de comprimento (2048 caracteres)
- Só devem ser usadas para obter informações.

Método POST

O método POST envia informações que serão processadas por um recurso específico

A query string viaja no corpo da mensagem através de HTTP e não está visível na URL.

```
POST /test/demo_form.php HTTP 1.1
Host: digitalhouse.com
```

Método POST

- As solicitações não podem ser armazenadas em cache
- Não aparecem no histórico
- Não podem ser adicionadas a marcadores (favoritos)
- Sem restrição de comprimento da query string.
- Utilizado para enviar informações

GET x POST

	GET	POST
Back Button	Sem consequências	As informações são reenviadas
Marcadores	Aceita	Não aceita
Cache	Aceita	Não aceita
Restrição comprimento	Sim	Não
Segurança	Não utilizar com dados confidenciais	Mais seguro que GET
Visibilidade	As informações são exibidas na URL	As informações não são exibidas na URL

GET x POST na URL

Método GET:

```
http://www.test.com/index.php?name1=value1&name2=value2
```

Método POST:

```
http://www.test.com/index.php
```

Métodos

- **GET**: Requisita um representação do recurso especificado (O mesmo recurso pode ter várias representações, ao exemplo de serviços que retornam XML e JSON).
- **HEAD**: Retorna os cabeçalhos de uma resposta (sem o corpo contendo o recurso)
- **POST**: Envia uma entidade e requisita que o servidor aceita-a como subordinada do recurso identificado pela URI.
- **PUT**: Requisita que um entidade seja armazenada embaixo da URI fornecida. Se a URI se refere a um recurso que já existe, ele é modificado; se a URI não aponta para um recurso existente, então o servidor pode criar o recurso com essa URI.
- **DELETE**: Apaga o recurso especificado.
- TRACE: Ecoa de volta a requisição recebida para que o cliente veja se houveram mudanças e adições feitas por servidores intermediários.
- **OPTIONS**: Retorna os métodos HTTP que o servidor suporta para a URL especificada.
- **CONNECT**: Converte a requisição de conexão para um túnel TCP/IP transparente, usualmente para facilitar comunicação criptografada com SSL (HTTPS) através de um proxy HTTP não criptografado.
- **PATCH**: Usado para aplicar modificações parciais a um recurso.

PHP - Variáveis superglobais

Com PHP, é possível entender os tipos de pedidos que chegam e como acessar a **query string** através de **variáveis superglobais**.

As variáveis superglobais são variáveis disponíveis sempre em **todos os escopos**.

PHP - $_GET

$_GET é uma variável superglobal. É um **array associativo** gerado a partir da **query string** em uma solicitação **GET**.

Se houvesse um pedido para
`http://localhost/test.php?nome=João&sobrenome=Silva`

$_GET seria
`["nome" => "João", "sobrenome" => "Silva"]`

Então, **$_GET["nome"]** tem o valor **"João"**.

PHP - $_POST

$_POST é uma variável superglobal. É um **array associativo** gerado a partir da **query string** em uma solicitação **POST**.

Funciona igual a $_GET mas em pedidos feitos por **POST**.

PHP - $_REQUEST

$_REQUEST é uma variável superglobal. É um **array associativo** gerado a partir da **query string** em uma solicitação **POST** ou **GET**.

Funciona igual a $_GET e $_POST mas **não importa o método**.

Se o meu $_GET é igual a `['nome'=>'Testino']` e o meu $_POST é igual a `['email'=>'teste@teste.com', 'senha'=>'123456']`, o $_REQUEST será igual a **`['nome'=>'Testino','email'=>'teste@teste.com', 'senha'=>'123456']`**

PHP - Superglobais

Outras **superglobais**:

- $GLOBALS
- $_SERVER
- $_GET
- $_POST
- $_FILES
- $_COOKIE
- $_SESSION
- $_REQUEST
- $_ENV

Vamos praticar!

Exercícios

DAILY SCRUM

maximo 15min

- **o que fiz ontem?**
- **o que estou fazendo?**
- **tem algum impedimento?**

dica: use um cronometro!

OBRIGADO!