

1 Curso POO PHP Protocolo HTTP

1.1 Sumario

- 1 Protocolo HTTP
 - ◆ 1.1 Mensaxes de petición
 - ◆ 1.2 Mensaxes de resposta
 - ◆ 1.3 Encabezados
 - ◆ 1.4 Exemplo dunha comunicación HTTP
 - ◆ 1.5 Extensións PHP relacionadas co protocolo HTTP

1.2 Protocolo HTTP

O protocolo HTTP baséase no envío de mensaxes de petición e de resposta. Cada unha destas mensaxes componse de:

- Liña de petición ou de resposta.
- Encabezados.
- Corpo da mensaxe (opcional).

1.2.1 Mensaxes de petición

Nunha mensaxe de petición típica, a primeira liña contén o método da petición, a dirección do recurso e a versión do protocolo HTTP. Por exemplo:

```
GET /imaxes/logo.png HTTP/1.1
```

Os **métodos máis comúns dunha petición HTTP** son:

- GET. Pide unha representación do recurso especificado (o contido do arquivo correspondente ao recurso).
- HEAD. Pide unha resposta idéntica á que correspondería a unha petición GET, pero sen o corpo da resposta. Isto é útil por exemplo para saber se unha páxina web que o navegador ten almacenada en caché, foi actualizada no servidor.
- POST. Envía datos para que sexan procesados polo recurso identificado. Os datos inclúense no corpo da mensaxe.
- PUT. Sobe un recurso especificado (arquivo) ao servidor.
- DELETE. Borra o recurso especificado.

1.2.2 Mensaxes de resposta

Nunha mensaxe de resposta, a primeira liña contén a versión do protocolo, un código de estado e unha descrición. Por exemplo:

```
HTTP/1.1 200 OK
```

O **código de resposta** ou código de estado é un número de tres cifras que serve para indicar se unha petición se recibiu e atendeu correctamente, ou se se produciu calquera problema. O primeiro dígito indica o tipo do código.

- 1 xx (Resposta informativa). Indícalle ao cliente que se recibiu a petición e se está a continuar. Trátase dunha resposta provisional e hai que realizar unha nova transacción HTTP para que se poida obter a resposta definitiva. Non se aconsella que os servidores envíen estas respostas.
- 2 xx (Petición correcta). Indica que a petición recibida anteriormente foi recibida, aceptada e procesada correctamente no servidor.
- 3 xx (Redirección). Indícaselle ao cliente HTTP que ten que realizar algunha acción adicional para que se poida resolver completamente a petición que se realizou.
- 4 xx (Erros do cliente). Recibiuse unha petición cunha sintaxe errónea ou non se puido procesar a petición.
- 5 xx (Erros do servidor). Produciuse un erro no servidor que lle impediu atender e procesar a petición.

Os máis comúns son:

- 200 (OK). É a resposta estándar para peticións correctas.
- 403 (Prohibido). O servidor rexeita responder á solicitude por falta de privilexios (por exemplo, o usuario que o solicita non está autenticado ou non ten permiso).
- 404 (Recurso non encontrado). Utilízase cando o servidor web non encontra a páxina ou recurso solicitado.

1.2.3 Encabezados

Os encabezados utilízanse en función do método da petición ou resposta, para dar máis información. Hai encabezados xerais (utilízanse tanto en mensaxes de petición coma en mensaxes de resposta), e encabezados específicos para mensaxes de petición ou de resposta. Por exemplo:

- **Host** (de petición). Equipo ao que se envía a petición.
- **User-Agent** (de petición). Nome e versión do cliente (navegador) e do sistema operativo.
- **Server** (de resposta). Nome e versión do software que se executa no servidor web.
- **MIME-Version** (xeral). Versión de MIME que utiliza o cliente.
- **Accept** (de petición). Tipo de contido que acepta o navegador.
- **Accept-Language** (de petición). Idiomas que espera o navegador nas páxinas recibidas.
- **Accept-Encoding** (de petición). Sistema de codificación que espera o navegador para o recurso a recibir.
- **Accept-Charset** (de petición). Xogo ou conxunto de caracteres que espera recibir o navegador.
- **Referer** (de petición). URL dende onde se orixinou a petición (URL da páxina que contiña o enlace).
- **Location** (de resposta). Emprégase para redireccionar o navegador a unha nova URL.
- **Set-Cookie** (de resposta). Envía unha cookie do servidor ao cliente para que a almacene.
- **Cookie** (de petición). Envía ao servidor as cookies almacenadas para o dominio.

Por exemplo, para indicar que a petición vai destinada ao equipo `www.iessanclemente.net`, engadiríase o seguinte encabezado á mensaxe de petición:

```
Host: www.iessanclemente.net
```

Dende PHP podemos empregar a función **header** para engadir un encabezado específico á páxina de resposta xerada. Esta función debe empregarse antes de ter xerado calquera tipo de saída no script. Por exemplo, para redireccionar a petición a unha nova páxina, podemos facer:

```
<?php
header("Location: http://www.iessanclemente.net/");
?>
```

Temos outras dúas funcións relacionadas cos encabezados que poden ser de utilidade:

- Para saber os encabezados que imos a enviar na resposta ao cliente podemos empregar a función **headers_list**.
- Para comprobar se aínda non foron enviados os encabezados (e polo tanto aínda estamos a tempo de engadir algún máis) podemos empregar **headers_sent**.

1.2.4 Exemplo dunha comunicación HTTP

Unha petición HTTP podería ter a seguinte forma:

```
GET /index.html HTTP/1.1
Host: www.iessanclemente.net
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4
[Liña en branco]
```

E a súa respectiva resposta:

```
HTTP/1.1 200 OK
Date: Fri, 21 Oct 2021 14:21:29 GMT
Content-Type: text/html
Content-Length: 2221

<html>
<body>
  <h1>Encabezado</h1>
  (Contido)
  ?
</body>
</html>
```

1.2.5 Extensións PHP relacionadas co protocolo HTTP

En PHP podemos empregar as seguintes extensións para traballar directamente co protocolo HTTP:

- **cURL**. Utiliza a librería libcurl para funcionar como cliente de varios protocolos, entre eles HTTP e HTTPS.

- **HTTP**. Facilita entre outras o manexo das mensaxes HTTP.

--V́ctor Lourido 02:45 14 jul 2013 (CEST)