

01

## Depurando o método HTTP

### Transcrição

Vamos fazer um teste e usar o navegador para mostrar mais detalhes sobre a comunicação HTTP. Os navegadores mais populares como Google Chrome, Mozilla Firefox ou Microsoft Edge possuem ferramentas e plugins que visualizam como o navegador trabalha e usa o HTTP.

Para habilitar as ferramentas do desenvolvedor no **Chrome** vá ao menu à direita (as reticências na vertical): **Mais ferramentas** → **Ferramentas do desenvolvedor**, ou no menu superior: **Ferramentas** → **Ferramentas do desenvolvedor**. Após isso, selecione a aba **Network**.

No **Firefox** vá ao menu superior: **Ferramentas** → **Desenvolvedor web** → **Exibir/Ocultar ferramentas**.

Para o **Internet Explorer** aperte a tecla **F12** para abrir o console do desenvolvedor e selecione a aba **Rede** (ou **Network**).

### Método GET do HTTP

Vamos abrir o console de desenvolvedor e acessar o <http://www.alura.com.br> (<http://www.alura.com.br>). Aqui usaremos o navegador Chrome, mas nos outros navegadores o comportamento é bem parecido.

No console podemos ver todas as requisições HTTP executadas pelo Chrome. Mas não só isso, também aparecem alguns códigos e métodos, além do tempo de execução para cada requisição. Repare que chamamos apenas o <http://www.alura.com.br>, mas foram feitas várias outras requisições em seguida.

Name	Method	Status	Type
alura.com.br	GET	301	
www.alura.com.br	GET	200	document
titulos.1487007969.css	GET	200	stylesheet
home.1487007969.css	GET	200	stylesheet
facebookpixel.1487007969.js	GET	200	script
alura-logo.1487007969.svg	GET	200	svg+xml
label-carreiras.1487007969.svg	GET	200	sv
356ec3198c1bab130fbae73d9a8d4415.png?r=PG&size=50x50&d...	GET	302	text/html

Na primeira coluna aparece a URL (o endereço) e na segunda coluna o método HTTP. O método HTTP indica qual é a **intenção ou ação dessa requisição**. Enviamos uma requisição com o método **GET**. Queremos receber informações, sem modificar algo no servidor, que é justamente a ideia do método **GET**.

### Primeiro código da resposta

Como resposta recebemos o código de status **301**. O protocolo HTTP define alguns códigos padrões para esclarecer a resposta. Indo com o mouse em cima do **301** o Chrome mostra o significado desse código: **Moved Permanently**. Ou seja, o site Alura foi movido para outro lugar! Eis a questão: Onde então está o site Alura?

A localização ou a URL concreta está na resposta HTTP. Vamos clicar em cima do código de status **301** para receber mais informações. Aqui o Chrome mostra todos os cabeçalhos da requisição e da resposta. São muitos (nem tantos) mas o que nos interessa é a nova localização do site. Dentro do item **Response Headers** podemos ver todos os cabeçalhos que o servidor devolveu e logo logo apareceu um com o nome **Location**. Esse cabeçalho indica a nova URL, só que agora usando **https**.

Quando o navegador recebe o status `301` ele já sabe que é preciso enviar uma nova requisição e procura a nova URL no cabeçalho de resposta `Location`.

The screenshot shows a browser developer tools interface with the 'Headers' tab selected. In the 'General' section, the 'Request URL' is listed as `http://alura.com.br/`, 'Request Method' as 'GET', 'Status Code' as '301 Moved Permanently (from disk cache)', and 'Remote Address' as '104.28.18.109:80'. In the 'Response Headers' section, the 'Location' header is present with the value `https://www.alura.com.br/`. The left sidebar lists various resources loaded by the page, including CSS files like 'titulos.1487007969.css' and JavaScript files like 'facebookpixel.1487007969.js'.

## Redirecionando entre sites

Se alguém acessa a Alura usando `http` (lembrando, *inseguro*) automaticamente é chamado o site seguro (`https`). Isto é um comportamento muito comum para garantir que usamos `https` sempre. Se esquecermos de usar `https`, o servidor devolve o status `301` com a nova localização, mas agora usando `https`.

O navegador, ao receber `301`, chama automaticamente a nova URL. No mundo de desenvolvimento web este comportamento é chamado de *Redirecionamento pelo navegador*, ou **Redirecionamento no lado do cliente**. Fomos redirecionados para o recurso correto. A tarefa do desenvolvedor é definir o código de resposta e, no caso em que algum recurso tenha mudado a URL, o código `301` será usado com o cabeçalho `Location`.

## O código 200

Continuando no console, a segunda requisição foi para `https://www.alura.com.br`, novamente usando o método `HTTP GET` com a intenção de receber dados. Agora o código de resposta foi `200`. Este é um dos códigos mais comuns e significa que tudo deu certo! Dessa vez não foi preciso fazer um redirecionamento (não tem o cabeçalho `Location` na resposta) e não deu nenhum outro problema. A requisição foi aceita e processada corretamente - código `200`. Perfeito.

## Tipos de dados diferentes

No console podemos ver que aparecem mais requisições (cada linha representa um novo request). Quando o servidor Alura devolve a resposta para o navegador vem o conteúdo da página inicial em um formato especial, chamado de HTML. O HTML define a estrutura da nossa página, define os menus, botões, links, rodapé etc. Mas dentro do HTML não vêm as imagens e outros arquivos necessários para deixar o site perfeito. Dentro dele vem apenas a URL (endereço) desses outros recursos.

Então, ao receber o HTML, o navegador dispara várias outras requisições para carregar as imagens, fontes e outros dados. Como também são requisições HTTP, o console mostra suas informações. Podemos ver que na resposta vem o tipo do conteúdo, por exemplo `text/html`, `text/css`, `image/svg+xml`, entre outros.

O importante é saber que o protocolo HTTP não está preso em algum formato específico. Podemos tratar qualquer informação com ele, seja texto ou binário!