

Abreviações:

JS JavaScript (linguagem de programação EcmaScript, conhecida como JavaScript)

lh: localhost

Um objeto JS é uma estrutura de dados tal como um dicionário.

1 Rotas

São endereços na URL em que um fluxo de informações fica disponibilizado pelo servidor. Uma rota começa por um endereço de um host, adicionada de uma porta para comunicação e termina em algum sufixo.

Dada a definição da rota, apenas a porta e o sufixo são mutáveis, já que o host (servidor) não se altera. Fixando-se uma porta, podemos exemplificar algumas rotas:

- “/”
- “/user”
- “/user/channel”
- “/clients?filter=idade&type=gt&value=18”

Essa última é uma rota formatada como uma Query, indicando o objeto JS

```
{filter: "idade", type: "gt", value: "18"}
```

A rota completa do último exemplo para a porta 3030 é então (lh será o domínio do site quando este for ao ar).

lh:3030/clients?filter=idade&type=gt&value=18

Por abuso de linguagem/notação, às vezes chamamos o sufixo de rota.

2 Métodos HTTP

Há outros métodos, mas os principais são descritos.

GET pegar informação

POST criar informação

PUT alterar informação

DELETE apagar informação

3 Códigos de estado HTTP

São códigos indicando o funcionamento dos métodos HTTP, quando sucedem ou quando dão erros. Um código é um número inteiro, mundialmente padronizado, indicando a causa do erro. Para mais detalhes, clique aqui [X](#).

4 Estruturas de uma requisição HTTP

Uma requisição é o pedido de informações para um servidor. É constituída de um objeto JS, contendo os seguintes campos descritos abaixo. A estrutura para certa finalidade pode ser qualquer uma a priori, mas por certas razões, se escolhe uma ou outra.

QUERY Permite o fluxo de informações diretamente na URL a partir da rota. É um objeto JS parseado na URL. Inicia-se com “?”; os campos são informados como “campo=valor”, e separa-se os campos por “&”. Caracteres não imprimíveis ou que não sejam ASCII são devidamente codificados. É interessante para filtrar dados em uma busca.

BODY É um objeto JS. Útil para o fluxo de várias informações. Precisa ser codificado/decodificado de acordo com algum formato, como JSON, XML, ou outros.

PARAMS Permite o fluxo de apenas uma informação. São parâmetros adicionados diretamente nas rotas: `lh:port/route/:id` para receber um parâmetro `id` diretamente pela rota. O cliente envia, por exemplo `lh:port/route/a3b8` em que `a3b8` é a `id` enviada.

HEADERS Permite o fluxo de várias informações. Pode servir para **enviar** uma `id` autorizando a execução de algo. **Recebe** os códigos de requisição, os chamados códigos HTTP (números inteiros), tais como `404` “*Page not found*”