



06. Parâmetros da requisição

📅 Date	@10/11/2022
⌵ Categoria	HTTP
⌵ Curso	HTTP: Entendendo a web por baixo dos panos

Tópicos

- Revisando o capítulo anterior
- A qual grupo eu pertenço?
- Parâmetros na requisição com métodos GET e POST
- Testando parâmetros da requisição
- Enviando parâmetros de forma correta
- Qual é o método HTTP?
- Por que POST?
- Para saber mais: Parâmetros na URL
- Para saber mais: Outros métodos HTTP e Web Services
- O que aprendemos?

Por que POST?

Seguem os dados da requisição para efetuar o login na plataforma Alura:

```
POST /signin/ HTTP/1.1
HOST: https://www.alura.com.br
Content-Type: application/x-www-form-urlencoded
```

```
email=nico.steppat%40caelum.com.br&senha=totalmentesecreta
```

Por que foi utilizado o método **POST** ? Usamos **POST** para incluir os parâmetros no corpo da requisição.

Utilizando o método **GET** , tanto o login quanto a senha seriam passados como parâmetro na URL, coisa que não queremos que aconteça. O método **POST** deixa os parâmetros no corpo da requisição, assim evita que informações importantes, como a senha, fiquem explícitas na URL.

Usando o método **GET** , a URL ficaria:

```
GET /signin/?email=nico.steppat@caelum.com.br&senha=totalmentesecreta HTTP/1.1  
HOST: https://www.alura.com.br
```

Logo, o **POST** foi utilizado para que se enviasse os valores do formulário no corpo da requisição.

Para saber mais: Outros métodos HTTP e Web Services

Já falamos bastante sobre os métodos (ou verbos) HTTP, **GET** e **POST** . Esses dois são utilizados na grande maioria das aplicações web, e fazem parte do dia a dia do desenvolvedor, no entanto existem diversos outros métodos.

Se o **GET** foi criado para receber dados, e o **POST** para adicionar algo no servidor, será que não existe algo para apagar e atualizar?

A resposta é sim, e os métodos se chamam **DELETE** e **PUT** .

Novamente esses métodos normalmente não são utilizados quando se trata de uma aplicação web, e são mais importantes quando o assunto é Web Services.

Agora vem a pergunta, você já ouviu falar de Web Services?

Quando falamos de um Web Service, sempre usamos o protocolo da web, ou seja o HTTP.

Um Web Service disponibiliza uma funcionalidade na web, através do protocolo HTTP. As funcionalidades variam muito e dependem muito da empresa e do negócio dela, mas por exemplo, na Alura temos um Web Service que traz todas as informações de

um curso (nome, capítulos, exercícios, etc). O Google ou Facebook possuem muitos Web Services para acessar um usuário, ver os posts dele, interesses, etc. Muitas vezes esses serviços são pagos.

O importante é que sempre usamos o protocolo HTTP. A grande diferença de um Web Service é que os dados **não** vem no formato HTML, e sim em algum formato independente da visualização, como XML ou JSON.

Temos um pequeno exemplo de um Web Services que usamos em um dos treinamentos presenciais. Tente acessar:

<https://argentumws-spring.herokuapp.com/negociacoes>

Repare que recebemos dados sobre negociações, mas o formato é XML. Isso é um Web Service! É a tarefa do cliente ler os dados e apresentar para o usuário final. O cliente não precisa ser o navegador (e normalmente não é), pode ser um celular ou uma aplicação Desktop.

O que aprendemos?

- **GET** Receber dados
 - **POST** Submeter dados
 - **DELETE** Remover um recurso
 - **PUT** Atualizar um recurso
-