

01

Buscando dados com get

Transcrição

Já implementamos o nosso spinner e está funcionando corretamente, vamos continuar aprender mais sobre AJAX.

O nosso usuário pode mudar a frase através do botão que escolhe uma frase aleatoriamente. Para melhorar ainda mais a experiência seria interessante se o usuário pudesse escolher a frase, sem depender do sorteio. A ideia é que o usuário coloque a id da frase e baseado nessa id carregaremos a frase do servidor, com AJAX!

Preparando o HTML

Antes de mais nada é preciso ter um novo botão e um input para digitar a id. No arquivo `principal.html` adicionaremos mais um botão usando a id `botao-frase-id` e a cor `cyan`

```
<a id="botao-frase-id" class="btn-floating btn-large waves-effect waves-light cyan">
  <i class="material-icons">repeat_one</i>
</a>
```

Repare a cor `cyan` e a id `botao-frase-id` da tag `a`.

Logo abaixo colocaremos o input para digitar a id da frase. Como aceitamos apenas números inteiros vamos usar o atributo `type="number"` com `min="0"`:

```
<input type="number" min="0" id="frase-id">
```

E para ajustar o tamanho aplicamos um estilo no nosso arquivo CSS `estilos.css`:

```
#frase-id{
  width: 40px;
}
```

Alura Typer

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



HTTP GET com Parâmetros

Com o HTML e CSS devidamente preparado falta realmente implementar a funcionalidade, ou seja, usar jQuery para enviar uma requisição AJAX. A ideia é que o usuário digite a id da frase no `input` e clique depois no botão para carregar essa frase.

Já que a funcionalidade tem a ver com frase vamos colocar o código dentro do arquivo `frase.js`, em uma nova função `buscaFrase`

```
//arquivo frase.js
function buscaFrase() {

    //fique a vontade para usar console.log()
}
```

Na função `buscaFrase` devemos executar uma requisição GET que não tem novidade para a gente. A novidade é que devemos enviar também o id da frase! Para tal, criamos um objeto JavaScript e o passamos para a função `.get` do jQuery:

```
//arquivo frase.js
function buscaFrase() {

    $("#spinner").toggle();

    var fraseId = $("#frase-id").val();
    var dados = {id : fraseId}; //criacao do objeto JS que guarda a id

    //passando objeto como segundo parâmetro
    $.get("http://localhost:3000/frases", dados, trocaFrase);
}
```

Dessa forma o jQuery enviar a id como parâmetro da requisição HTTP GET!

Para melhorar a usabilidade já vamos adicionar o spinner e o tratamento de erro como visto anteriormente:

```
//arquivo frase.js
function buscaFrase() {
    $("#spinner").toggle();
    var fraseId = $("#frase-id").val();

    var dados = {id : fraseId}; //criacao do objeto JS que guarda a id

    //passando objeto como segundo parâmetro
    $.get("http://localhost:3000/frases", dados, trocaFrase)
    .fail(function(){
        $("#erro").toggle();
        setTimeout(function(){
            $("#erro").toggle();
        },2000);
    })
    .always(function(){
        $("#spinner").toggle();
    });
}
```

Repare também que usamos uma função `trocaFrase` como terceiro parâmetro do `$.get` mas essa função não existe ainda! Vamos adicioná-la logo abaixo da função `buscaFrase`:

```
//arquivo frase.js
function trocaFrase(data) {
    console.log(data);

    var frase = $(".frase");
    frase.text(data.texto); //cuidado, texto com "o" no final
    atualizaTamanhoFrase();
    atualizaTempoInicial(data.tempo);
}
```

E no início do arquivo não podemos esquecer de associar o botão com o evento `click` para executar a função `buscaFrase` quando o botão for acionado:

```
$("#botao-frase-id").click(buscaFrase);
```

A nossa função ficou bem parecida com a que busca todas as frases com a diferença que aprendemos como enviar dados através da função `.get` do jQuery.

O que aprendemos?

- Como enviar dados com AJAX e jQuery
 - Permitindo o usuário buscar uma frase específica
 - Enviando dados junto da requisição
 - Implementando a troca de frase

