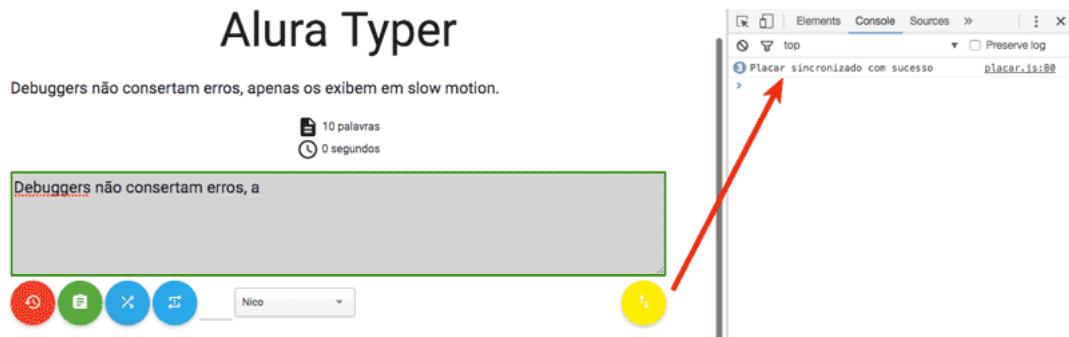


03

## Notificação com Tooltipster

### Transcrição

Nesse vídeo vamos ver mais um plugin para melhorar a experiência do usuário. Quando o usuário estiver sincronizando o placar, estamos enviando os dados para o servidor mas não estamos dando nenhum feedback para o usuário. Infelizmente aparece apenas uma mensagem no console que é algo para o desenvolvedor e não destinado ao usuário final:



### Plugin Tooltipster

Nas aulas anteriores usamos o Spinner para mostrar que uma requisição AJAX está sendo executada. Nesse capítulo veremos um novo plugin, o Tooltipster, para dar feedback ao usuário:

<http://iamceege.github.io/tooltipster/> (<http://iamceege.github.io/tooltipster/>)

Esse plugin facilita dar feedback através desses pequenos balões que mostram alguma info:



### Importando Tooltipster

Como os outros plugins, o Tooltipster precisa de um CSS e JavaScript, além do nosso jquery. Você pode baixar o tooltipster diretamente pelo site:

<https://github.com/iamceege/tooltipster/archive/master.zip> (<https://github.com/iamceege/tooltipster/archive/master.zip>)

Ou baixar o ZIP já com os arquivos específicos [aqui](https://s3.amazonaws.com/caelum-online-public/jquery-alura-typer-2/tooltipster.zip) (<https://s3.amazonaws.com/caelum-online-public/jquery-alura-typer-2/tooltipster.zip>):

Nesse ZIP você encontrará exatamente os dois arquivos ( `tooltipster.bundle.min.css` e `tooltipster.bundle.min.js` ) necessários.

Extraia o ZIP e move:

- o arquivo `tooltipster.bundle.min.css` para a pasta `public/css/libs`
- o arquivo `tooltipster.bundle.min.js` para a pasta `public/js`

Além disso, é preciso importar os dois arquivos na página `public/principal.html`. Primeiro coloque o CSS do Tooltipster antes do nosso `estilos.css`:

```
<!-- principal.html-->

<link rel="stylesheet" href="css/libs/tooltipster.bundle.min.css" />
```

E no final da página arquivo JavaScript:

```
<script src="js/tooltipster.bundle.min.js"></script>
```

## Preparando HTML

Para esse plugin funcionar no nosso HTML devemos aplicar a classe `tooltip`. Como queremos dar uma mensagem ao sincronizar devemos mexer no botão amarelo:

```
<a class="tooltip .....>

</a>
```

e ainda nessa tag adicionar o atributo `title` para definir a dica:

```
<a class="tooltip ....." title="Sincronizado com sucesso">

</a>
```

Segue uma vez o botão amarelo completo:

```
<a id="botao-sync" class="btn-floating btn-large waves-effect waves-light yellow tooltip" title="

    <i class="material-icons">swap_vert</i>

</a>
```

## Ativando Tooltipster

Por fim devemos usar um pequeno JavaScript para realmente acionar a dica (`Tooltip`). Novamente vamos abrir o arquivo `main.js` e adicionar um código dentro da função inicial do jQuery.

Depois do código do plugin `selectize` adicione:

```
$(".tooltip").tooltipster();
```

Repare que selecionamos o elemento pela classe `.tooltip` para chamar a função `tooltipster`.

Já podemos testar a funcionalidade e mover o mouse em cima do botão amarelo aparece a mensagem "Sincronizado com sucesso". Ou seja, o plugin funciona mas a mensagem só deveria aparecer quando realmente houver uma sincronização. Em outras palavras, usamos o evento errado para mostrar o *tooltip*.



## Alterando o evento

O evento padrão do tooltipster é o *mouse over* mas claro que podemos alterar isso. O tooltipster chama isso de **trigger** e oferece uma propriedade para alterar o evento. Aí vem um outro problema, não queremos usar nenhum evento ou trigger padrão. Ou seja, o tooltip não deve aparecer ao clicar ou dar um *mouse-over*. Devemos mostrar a mensagem quando a requisição AJAX terminou, por isso vamos personalizar o *trigger*:

```
$(".tooltip").tooltipster({
  trigger: "custom"
});
```

Isso já desabilita qualquer evento padrão e nossa mensagem não aparece mais. Para mostrar ela após a requisição AJAX vamos chamar o tooltip programaticamente dentro da função `$.post`. No arquivo `placar.js` logo após do comando `console.log(..)`:

```
//placar.js
$.post("http://localhost:3000/placar", dados, function() {
  console.log("Placar sincronizado com sucesso");
  $(".tooltip").tooltipster("open"); //novo
});
```

Podemos testar isso e realmente aparece a mensagem após sincronizar. Legal, mas ela não desaparece mais :( Como mostramos a mensagem programaticamente também devemos fechar ela após um determinado tempo.

## Fechando tooltip

Ainda bem que já conhecemos o AJAX com jQuery e já vimos que a função `always` é sempre executada, tanto no sucesso, tanto no erro. Vamos aproveitar ela e adicionar o código para fechar o tooltip.

Na função `always` também já vamos definir um pequeno *timeout* para esperar um pouco antes de fechar a mensagem:

```
//placar.js
$.post("http://localhost:3000/placar", dados, function() {
  console.log("Placar sincronizado com sucesso");
  $(".tooltip").tooltipster("open");
}).always(function(){ //novo
```

```
setTimeout(function() {
  $(".tooltip").tooltipster("close");
}, 1200);
```

Ao testar e clicar no botão de sincronização a mensagem aparece e some depois de um tempo. Perfeito!

## Mensagem de erro

Podemos personalizar ainda mais o Tooltipster e mostrar uma mensagem diferente quando a requisição AJAX falha. Lembrando que existe uma função `fail` que é chamada quando a requisição falha que usaremos para mostrar uma mensagem de erro através do Tooltipster:

```
//placar.js

$.post("http://localhost:3000/placar", dados, function() {
  console.log("Placar sincronizado com sucesso");
  $(".tooltip").tooltipster("open");
}).fail(function(){
  $(".tooltip").tooltipster("open").tooltipster("content", "Falha ao sincronizar");
}).always(function(){ //novo
  setTimeout(function() {
    $(".tooltip").tooltipster("close");
  }, 1200);
});
```

Ao testar, e causar um erro, por exemplo com uma URI errado na função `$.post` deve aparecer a mensagem "Falha ao sincronizar"!