# **Reagindo ao click**

# Sabemos que o Angular interage com a interface de eventos do JavaScript. Pensando nisto, qual das diretivas abaixo devemos utilizar para detectar o click do usuário em um botão e executar alguma função ?

* ng-mouseover
* ng-onclick
* **ng-click**
* ng-submit

**Explicações**

A resposta correta é ng-click.

Para cada evento padrão do JavaScript, existe sua diretiva correspondente no mundo Angular. É claro, se novos eventos forem criados, a biblioteca core do Angular precisará ser atualizada para suportar esses eventos. Vamos dar uma trégua ao time do Angular :)

# 

# 

# **Acessando parâmetros de rota**

# Vejamos a seguinte rota:

# ... $routeProvider.when('/produtos/:produtoId', ....) …

# Suponha que o usuário digite o seguinte endereço:

# <http://localhost:3000/#/produtos/15>

# Vejamos o seguinte controller:

# angular.module('vendas',[]).controller('ProdutoController', function($scope) { // código faltando });

# Complete o código acima para exibir com console.log o ID do produto da rota digitada pelo usuário. Segue o código anterior, basta alterá-lo:

**Explicações**

angular.module('vendas',[]).controller('ProdutoController', function($scope, $routeParams) {

console.log($routeParams.produtoId);

});

Precisamos do artefato $routeParams para termos acesso aos parâmetros passados nas URL das rotas. Ele é um objeto cuja as propriedades equivalem ao nome do ID definido na configuração de rota. Por exemplo, se tivermos: produtos/:produtoId, acessamos o ID do produto através de $routeParams.produtoId. Se a rota for filmes/:categoriaFilmeId, acessamos como $routeParams.categoriaFilmeId.

# 

# **Deletando um recurso**

# Quando usamos o padrão REST, nossa API do servidor deve estar preparada para responder ao verbo DELETE, quando desejamos apagar um recurso, no caso da nossa aplicação, uma foto. Qual das opções abaixo apresenta sintaxe correta?

# $http.delete('v1/fotos/12') success(function() { }) .error(function(erro) { });

1. $http.delete('v1/fotos/12')  
   .success(function() {  
   })  
   .error(function(erro) {  
   });
2. $http.delet('v1/fotos/12')  
   .success(function() {  
   })  
   .error(function(erro) {  
   });
3. $http.delete('v1/fotos/12')  
   .success() {  
   })  
   .error(function(erro) {  
   });

**Resposta: b**

**Explicações**

**A resposta correta é :**

**$http.delete('v1/fotos/12')  
.success(function() {  
})  
.error(function(erro) {  
});**

**Repare que o código foi indentado quebrando linha nas funções success e error. Não é raro o desenvolvedor esquecer de colocar o ponto quando realiza essa quebra, como no seguinte exemplo:**

**$http.delete('v1/fotos/12')  
success(function() {  
})  
.error(function(erro) {  
});**

# **Poupando requisições**

# Mesmo após apagarmos uma foto do nosso servidor, quando voltamos para página inicial ainda vemos seu painel, como podemos observar abaixo:Lista atualizada após o recarregamento

Dado seu conhecimento em Angular, explique por que isto acontece:

**Explicações**

**A foto continua sendo exibida porque a lista de fotos no lado do cliente (navegador) não foi atualizada. Angular não fez "curso de adivinhação" para saber que deve atualizá-la. Uma solução é realizar uma nova consulta do servidor, que retornará a lista atualizada sem o elemento, porém estaríamos realizando mais uma requisição para o servidor. É uma boa prática, sempre que possível, evitar esse *round trip* até o servidor e neste caso, podemos evitá-lo conhecendo um pouco de JavaScript.**

**O $scope.fotos nada mais é do que um array, podemos usar a tão conhecida funçãosplice do JavaScript para remover a foto:**

**// public/js/controllers/fotos-controller.js  
  
angular.module('alurapic').controller('FotosController', function($scope, $http) {  
  
 $scope.fotos = [];  
 $scope.filtro = '';  
 $scope.mensagem = '';  
  
 $http.get('/v1/fotos')  
 .success(function(retorno) {  
 $scope.fotos = retorno;  
 })  
 .error(function(erro) {  
 console.log(erro)  
 });  
  
 $scope.remover = function(foto) {  
  
 $http.delete('/v1/fotos/' + foto.\_id)  
 .success(function() {  
 var indiceDaFoto = $scope.fotos.indexOf(foto);  
 $scope.fotos.splice(indiceDaFoto, 1);  
 $scope.mensagem = 'Foto ' + foto.titulo + ' removida com sucesso!';  
  
 })  
 .error(function(erro) {  
 console.log(erro);  
 $scope.mensagem = 'Não foi possível apagar a foto ' + foto.titulo;  
 });  
 };  
});**

**Conhecendo um pouco melhor a função:**

**$scope.fotos.splice(indiceDaFoto, 1);  
`**

**A função splice recebe dois parâmetros. O primeiro é o índice da posição que desejamos partir. Dado uma foto, conseguimos descobrir seu índice através da função do arrayindexOf, que recebe como parâmetro a foto. O segundo parâmetro de splice é a quantidade de elementos que desejamos remover, em nosso caso, apenas um.**