**Enviando dados para uma rota Back-End**

No decorrer do desenvolvimento da nossa aplicação, sentimos a necessidade de enviar os dados capturados pelo formulário para uma rota no **back-end** e fazemos isso através do serviço $http.

Sobre essa afirmação, julgue as alternativas corretas:

1) Para usar o serviço $http, recebemos sua instância através de injeção de dependências:

angular.module('alurapic').controller('FotosController', function($scope, $http) {  
 //código omitido  
});

Porém, para que a injeção do $http seja habilitada, precisamos importar o módulo ngHttp em nosso arquivo main.js, como no código abaixo:

var app = angular.module('alurapic', ['ngHttp', outros módulos ... ] );

2) É necessário importar o arquivo *JavaScript* do módulo $http nas páginas HTML.

<!-- public/index.html -->  
<!DOCTYPE html>  
<html lang="pt-br" ng-app="alurapic">  
 <head>  
 <script src="js/lib/angular.min.js"></script>  
 <script src="js/lib/angular-http.min.js"></script>  
 <script src="js/main.js"></script>  
 </head>  
 <body>  
 // código omitido  
 </body>  
</html>

3) O serviço $http nos fornece um método chamado post, que usamos para enviar os dados:

$http.post('/v1/fotos', $scope.foto)  
 .success(function() {  
 $scope.mensagem = 'Foto cadastrada com sucesso';  
 });

* 2 e 3 verdadeiras
* 1 e 3 verdadeiras
* **Somente a 3**
* Somente a 2

**Explicações**

O serviço $http nos é fornecido através de injeção de dependências. E com ele podemos fazer:

$http.post('/v1/fotos', $scope.foto)  
 .success(function() {  
 $scope.mensagem = 'Foto cadastrada com sucesso';  
 });

**Angular vs Validação HTML5**

O HTML5 já vem com uma série de recursos para validação de formulários, porém não são tão flexíveis e muitas vezes precisamos modificá-los via JavaScript para atingirmos nossos objetivos (se você ainda não precisou, excelente!).

É aí que entra a validação do Angular. Ela vem completar, inclusive ampliar os recursos de validação do HTML5, porém com a diferença de que temos flexibilidade no tratamento de mensagens e outros recursos.

A questão é que precisamos abdicar do sistema de validação do HTML5 e abraçar totalmente o do Angular. Sendo assim, precisamos desabilitar o sistema de validação do HTML5.

Qual das opções abaixo desativa corretamente a validação do HTML5:

1. <form no-validation>  
   </form>
2. <form novalidation>  
   </form>
3. <form no validade>  
   </form>
4. <form novalidate>  
   </form>

**Resposta: d**

**Explicações**

**Desabilitamos a validação do HTML5 adicionando o atributo novalidate no formulário. Não é raro o desenvolvedor confundir com novalidation ou até mesmo separar por hífen, algo como no-validate, algo que não funcionará.**

**As diretivas ng-hide e ng-show**

Podemos afirmar sobre as diretivas ng-hide e ng-show:

1) A ng-hide esconde o elemento se receber false e a ng-show esconde se receber true.

2) A ng-hide serve para esconder elementos, contanto que seu valor seja false.

3) A diretiva ng-hide não existe

4) A diretiva ng-show exibe elementos quando seu valor é false.

* 1 e 4 são verdadeiras
* 2 e 3 são falsas
* Apenas a 4 é verdadeira
* **Todas as afirmativas são falsas**

**Explicações**

**Pois é, todas as afirmativas são falsas.**

**A diretiva ng-hide esconde um elemento da tela quando seu valor for true, caso contrário ele continuará a ser exibido. Essa diretiva possui sua oposta, a diretiva ng-show. Quando ela tem valor true, exibe o elemento, caso contrário ele não será exibido. As duas diretivas possuem semânticas diferentes, apesar de conseguirem resultados semelhantes.**

**A importância da propriedade name**

Segue algumas afirmações sobre o atributo name quando usamos a validação do Angular:

1) Temos acesso ao objeto criado implicitamente pelo Angular, que representa nosso formulário através do valor que atribuímos ao name do formulário.

2) É a partir do objeto implícito que representa nosso formulário que temos acesso a todos os elementos de entrada, contanto que eles também possuam o atributo name devidamente preenchido.

3) Supondo que tenhamos um formulário com o name vendas, podemos em nossos controllers testar se ele é válido ou inválido através das propriedades $scope.vendas.$valid e $scope.vendas.$invalid, respectivamente.

4) É possível combinar a diretiva ng-disabled com o status de válido ou inválido do formulário, para desabilitar um botão caso o formulário esteja inválido.

* **Todas são verdadeiras**
* 2 e 3 são falsas
* Todas são falsas
* Somente a 3 é falsa

**Explicações**

**Todas as afirmativas são verdadeiras, porém uma curiosidade: usamos a diretiva ng-disabled, mas qual a razão de sua existência se já existe o atributo disabled, nativo do HTML? Bem, a especificação do HTML não requer que os navegadores preservem valores de atributos booleanos, como disabled. A presença desse atributo significa true e sua ausência false. Se tivéssemos colocado uma expressão do Angular para este atributo a informação de associação poderia ser perdida quando o navegador removesse o atributo.**

**Documentação do Angular:** [**https://docs.angularjs.org/api/ng/directive/ngDisabled**](https://docs.angularjs.org/api/ng/directive/ngDisabled)

**Mensagem de erro de validação**

**Temos o seguinte formulário com apenas um input.**

**<form name="calopsita">  
 <input name="nome" ng-model="passaro.nome" required ng-maxlength="20">  
 <!-- mensagem de erro obrigatório aqui -->  
 <!-- mensagem de erro tamanho aqui -->  
</form>**

Altere o código acima para exibir duas mensagens de erro de validação, uma para a obrigatoriedade do campo (atributo required) e outra para seu tamanho máximo (diretiva ng-maxlength). A ideia é que a mensagem seja exibida sempre que o campo estiver inválido, sem esperar sua submissão. Utilize uma tagspan para cada mensagem.

**Explicações**

**A solução precisa acessar o objeto do formulário criado pelo Angular e a partir dele acessar o input buscando por erros de validação nas propriedades $error.required e$erro.maxlength:**

**<form name="calopsita" novalidate>  
 <input name="nome" ng-model="passaro.nome" required ng-maxlength="20">  
 <span ng-show = "calopsita.nome.$error.required">  
 Título obrigatório  
 </span>   
 <span ng-show="calopsita.nome.$error.maxlength">  
 No máximo 20 caracteres!  
 </span>  
</form>  
 <!-- mensagem de erro obrigatório aqui -->  
 <!-- mensagem de erro tamanho aqui -->**

**Se quisermos que a validação seja realizada apenas quando o formulário for submetido, precisamos colocar uma condição E consultando o estado do formulário para saber se ele foi submetido ou não:**

**<form name="calopsita" novalidate>  
 <input name="nome" ng-model="passaro.nome" required ng-maxlength="20">  
 <span ng-show = "calopsita.$submitted && calopsita.nome.$error.required">  
 Título obrigatório  
 </span>   
 <span ng-show="calopsita.$submitted && calopsita.nome.$error.maxlength">  
 No máximo 20 caracteres!  
 </span>  
</form>  
 <!-- mensagem de erro obrigatório aqui -->  
 <!-- mensagem de erro tamanho aqui -->**