**Single Page Application**

Sobre **Single Page Applications** (SPA), podemos afirmar:

1) Não recarrega durante seu uso.

2) Parciais (partials) são fragmentos de páginas que são carregados dinamicamente.

3) Ao trabalhar com SPA no Angular, indicamos onde serão carregadas as *partials* usando a diretiva <ng-view>.

4) Podemos usar Angular em uma aplicação que não seja SPA.

Das afirmações acima, podemos diz que:

* Apenas 4 é falsa
* Apenas 1 e 4 são falsas
* Apenas 3 é verdadeira
* **Todas são verdadeiras**

**Explicação**

**Angular tem como foco a criação de Single Page Applications (SPA's), porém seu uso não é exclusivo para este tipo de aplicação. Uma SPA é aquela aplicação que não recarrega durante seu uso. Mas como seu conteúdo muda? No caso do Angular, há um sistema de rotas que funciona no lado do cliente capaz de carregar do servidor fragmentos de páginas, chamadas de partials (parciais). Esses fragmentos são inseridos na página principal da aplicação, que funciona como uma "moldura" apenas. Com essa analogia, cada parcial é um quadro que pode ocupar ou não a moldura. As partials não devem possuir tag <head> e <body>, já que farão parte de uma página que já as possui.**

**Ainda lembra como se cria uma diretiva?**

**Surpresa! Você aprendeu em um dos capítulos anteriores a criar a diretiva meuPainel. Para que seu conhecimento não caia em esquecimento , neste exercício você criará a diretiva minhaFoto, ela deve encapsular a seguinte marcação:**

**<img class="img-responsive center-block" src="{{url}}" alt="{{titulo}}">**

**Seu uso será:**

**<minha-foto url="{{foto.url}}" titulo="{{foto.titulo}}"></minha-foto>**

**Será que você consegue? Você pode consultar, o importante é que você crie essa diretiva do zero! O gabarito aparecerá depois de você enviar sua resposta. Boa sorte!**

**Explicação**

**Alterando minhas-diretivas.js para encadear mais uma chamada à .directive:**

**// public/js/directives/minhas-diretivas.js  
  
angular.module('minhasDiretivas', [])  
 .directive('meuPainel', function() {  
  
 // código omitido  
  
 })  
 .directive('minhaFoto', function() {  
  
 var ddo = {};  
  
 ddo.restrict = "AE";  
  
 ddo.scope = {  
 titulo: '@',  
 url: '@'  
 };  
  
 ddo.template = '<img class="img-responsive center-block" src="{{url}}" alt="{{titulo}}">';   
  
 return ddo;  
  
 });**

**Agora, usando nossa diretiva em index.html:**

**<!-- public/index.html -->  
  
<meu-painel ng-repeat="foto in fotos" titulo="{{foto.titulo}}">  
 <minha-foto url="{{foto.url}}" titulo="{{foto.titulo}}"></minha-foto>  
</meu-painel>**

**Rotas do Angular**

**Sobre a configuração de rotas no lado do cliente Angular:**

**1) Precisamos importar o módulo ngResource para que consigamos configurar rotas.**

**2) O artefato $routeProvider é um artefato injetável dependente do carregamento do módulo de rotas.**

**3) Rotas podem ser configuradas sem a necessidade de pertencerem a um módulo.**

**4) Para configurarmos uma rota, precisamos fornecer no mínimo o nome da rota e qual parcial será carregada.**

**Sobre as afirmativas anteriores, podemos dizer que:**

* apenas 4 é verdadeira
* **2 e 4 são verdadeiras**
* Apenas 1 é falsa
* 2 e 3 são verdadeiras

**Explicação**

**Para que possamos configurar rotas no lado do cliente, o módulo ngRoute precisa estar carregado, isso significa que precisamos importar o script do módulo e adicioná-lo como dependência de algum módulo que realizará a configuração de rotas. Por exemplo, partindo do pressuposto que o script do módulo de rotas está carregado:**

**angular.module('sistema',['ngRoute'])**

**É através da função config que configuramos rotas, solicitando através de injeção$routeProvider:**

**angular.module('sistema',['ngRoute'])  
.config(function($routeProvider){  
});**

**O artefato $routeProvider só estará disponível para injeção apenas se o módulo ngRoutetiver sido carregado previamente.**

**O $routeProvider possui a função when, que configura efetivamente a rota. Ela recebe dois parâmetros: o primeiro é o nome da rota (endereço) que desejamos configurar e o segundo um objeto JavaScript com duas propriedades: controller e templateUrl. O primeiro associa um controller à parcial e o segundo indica seu template. Um exemplo:**

**angular.module('sistema',['ngRoute'])  
.config(function($routeProvider){  
 $routeProvider.when('/produtos', { controller: 'ProdutosController', templateUrl : 'partials/produtos.html'});  
});**

**Porém, podemos configurar uma rota sem associá-la a um controller, algo perfeitamente aceitável dentro do Angular. Por exemplo, a rota que exibe a parcial sobre.html, que só exibe dados, sem qualquer outro uso do Angular:**

**angular.module('sistema',['ngRoute'])  
.config(function($routeProvider){  
 $routeProvider.when('/sobre', { templateUrl : 'partials/sobre.html'});  
});**

**Redirecionamento**

**Pode ser que o usuário queira ele mesmo digitar uma rota específica do Angular na barra de endereços do navegador e acabe errando. Nesse caso, podemos redirecionar o usuário para uma rota conhecida.**

**Qual das opções realiza esse redirecionamento corretamente?**

1. **angular.module('controle', ['ngRoute'])  
    .config(function($routeProvider) {  
     
    $routeProvider.when('/produtos', {  
    templateUrl: 'partials/produtos.html',  
    controller: 'ProdutosController'  
    });  
     
    $routeProvider.otherwise({redirectTo: '/produtos'});  
     
    });**
2. **angular.module('controle', [])  
    .config(function($routeProvider) {  
     
    $routeProvider.when('/produtos', {  
    templateUrl: 'partials/produtos.html',  
    controller: 'ProdutosController'  
    });  
     
    $routeProvider.otherwise({redirectTo: '/produtos'});  
     
    });**
3. **angular.module('controle', ['ngRoute'])  
    .config(function($routeProvider) {  
     
    $routeProvider.when('/produtos', {  
    templateUrl: 'partials/produtos.html',  
    controller: 'ProdutosController'  
    });  
     
    $routeProvider.otherwise({redirect: '/produtos'});  
     
    });**
4. **angular.module('controle', ['ngRoute'])  
    .config(function() {  
     
    $routeProvider.when('/produtos', {  
    templateUrl: 'partials/produtos.html',  
    controller: 'ProdutosController'  
    });  
     
    $routeProvider.otherwise({redirectTo: '/produtos'});  
     
    });**

**Resposta: a**

**Explicação**

**Vejamos um exemplo:**

**angular.module('controle', ['ngRoute'])  
 .config(function($routeProvider) {  
  
 $routeProvider.when('/produtos', {  
 templateUrl: 'partials/produtos.html',  
 controller: 'ProdutosController'  
 });  
  
 $routeProvider.otherwise({redirectTo: '/produtos'});  
  
 });**

**Nosso módulo tem como dependência o módulo ngRoute, sem o qual não seria possível recebermos via injeção $routeProvider para configurarmos nossas rotas. É através de$routeProvider.otherwise que realizamos um redirecionamento, passando um objeto com a propriedade redirectTo, que tem como valor uma rota existente do Angular.**

**$LocationProvider e html5Mode**

**Temos a seguinte configuração de rota:**

**angular.module('controle', ['ngRoute'])  
 .config(function($routeProvider) {  
  
 $routeProvider.when('/produtos', {  
 templateUrl: 'partials/produtos.html',  
 controller: 'ProdutosController'  
 });  
  
 $routeProvider.otherwise({redirectTo: '/produtos'});  
  
 });**

**Para habilitarmos o sistema de rotas no modo html5Mode sabemos que é necessária a diretiva <base href="/"> dentro da tag <head> da view principal da aplicação, porém isso não é suficiente.**

**Altere o código anterior habilitando o módulo HTML5. Cole sua resposta aqui.**

**Explicação**

**Precisamos do artefato $locationProvider para habilitarmos o html5Mode. É através dele que habilitamos este modo:**

**angular.module('controle', ['ngRoute'])  
 .config(function($routeProvider, $locationProvider) {  
  
 // habilitando  
 $locationProvider.html5Mode(true);  
  
 $routeProvider.when('/produtos', {  
 templateUrl: 'partials/produtos.html',  
 controller: 'ProdutosController'  
 });  
  
 $routeProvider.otherwise({redirectTo: '/produtos'});  
  
 });**