

Angular

WebII

Ordenando Valor

No titulo crie um link e coloque a diretiva ng-click

```
<a ng-click="sort('nome')">Nome</a>
```

Na classe controller defina um metodo sort

```
$scope.sort = function(s){  
    $scope.sortV = s;  
    $scope.order = !$scope.order;  
}
```

Crie duas variaveis no scopo sortV e order no ng-repeat vai ficar desta forma

```
<tr ng-repeat="contato in contatos | orderBy : sortV : order | filter: filtro">
```

Criando suas próprias diretivas

Um componente é um conjunto de código (Java, #C, HTML, CSS, JavaScript etc.) que encapsula determinado comportamento, podendo ser reutilizado facilmente em várias aplicações.

Vimos que o próprio AngularJS possui suas diretivas, que na verdade são componentes reutilizáveis que ampliam o vocabulário do navegador, ensinando-o novos truques.

Criamos nossas próprias diretivas quando queremos um componente HTML ou um comportamento reutilizável, inclusive podemos querer ambos ao mesmo tempo. Também utilizamos este recurso do AngularJS quando precisamos interagir com o DOM.

começar criando um módulo em separado que conterá nossas diretivas. Isso permitirá que elas sejam reutilizadas mais facilmente, bastando importar o novo módulo, sem a necessidade de carregarmos todos os outros recursos da aplicação

```
// public/js/directives/  
angular.module('meusComponentes', [])  
.directive('meuPainel', function() {  
    var directive = {};  
    return directive;  
});
```

Criamos o módulo meusComponentes, que não possui dependência alguma. Em seguida, chamamos a função directive, que recebe como primeiro parâmetro o nome da diretiva. Repare que seu nome utiliza o padrão CamelCase, porém, quando formos utilizá-la em nossa página, separaremos cada palavra por um hífen:

```
<meu-painel></meu-painel>
```

O segundo parâmetro da diretiva é uma função que sempre deve retornar um directive definition object (DDO).

A propriedade restrict

Nossa primeira configuração diz respeito à forma de utilização da diretiva. Queremos utilizá-la como (E)lemento ou como (A)tributo. Para isso, adicionamos em nosso DDO a propriedade restrict:

```
directive.restrict = "EA";
```

Vejamos os dois exemplos:

```
<!-- como (E)lemento -->
```

```
<meu-painel></meu-painel>
```

```
<!-- como (A)tributo -->
```

```
<div meu-painel></div>
```

A propriedade template

Sabemos que o AngularJS terá que processar nossa diretiva substituindo a pelo painel do Bootstrap. É por isso que nosso DDO possui a propriedade template. É nesta propriedade que definimos a marcação do nosso panel, e utilizamos a Angular Expression `{{titulo}}` no local da impressão do título

```
directive.template = '<div class="panel panel-default">' +  
' <div class="panel-heading">' +  
' <h3 class="panel-title">{{titulo}}</h3>' +  
' </div>' +  
' <div class="panel-body">' +  
' </div>' +  
'</div>';
```

Isolando o escopo de nossa diretiva

Queremos que nossa diretiva meu-painel receba seu título através do atributo titulo, nossa interface de comunicação com a diretiva:

```
<meu-painel titulo="Listagem de Contatos">  
<!-- conteúdo do painel -->  
</meu-painel>
```

Mas para que cada meu-painel de nossa página tenha seu próprio título, precisamos que todos possuam um escopo isolado. O escopo isolado de uma diretiva é independente do escopo em que ela existe, isto é, não herda do escopo pai no qual está inserida. Isso evita que a diretiva tente acessar algo que esteja fora dos seus limites, evitando que bagunce o escopo pai.


```
directive.scope = {  
    titulo: '@titulo'  
};
```

Como criamos um novo módulo, precisamos importá-lo no módulo principal da aplicação:

```
// public/js/main.js  
angular.module('contatooh', ['ngRoute', 'ngResource', 'meusComponentes'])  
    .config(function($routeProvider) {  
        // código omitido  
    });
```

Separando o DDO de seu template

Você deve ter reparado que escrever o template do painel dentro da diretiva torna sua manutenção mais trabalhosa. Imagine se o template fosse ainda mais complexo? Aliás, temos misturadas em um único arquivo a configuração da diretiva e sua apresentação. Podemos isolar o template da diretiva em seu próprio arquivo HTML, com a diferença de que não usaremos mais a propriedade `template`, mas `templateUrl` que aponta para o arquivo.