



TECNOLOGIA E INOVAÇÃO
EM PROL DA INDÚSTRIA



Curso Técnico em
Informática

Desenvolvimento de Sistemas

II – 180h

Prof^a: Francisleide Almeida

Angular

- Framework é mantido pelo Google;
- Funciona como uma extensão ao documento HTML, adicionando novos parâmetros e interagindo de forma dinâmica com vários elementos;
- Podemos adicionar novos atributos no html para conseguir adicionar funcionalidades extras, sem a necessidade de programar em javascript;

Angular - Diretiva

- AngularJS é quase uma linguagem declarativa: você usa novos parâmetros na linguagem html para alterar o comportamento padrão do html.
- Os parâmetros (ou propriedades) são chamados de diretivas;

Chamando o Angular

- Diretamente no documento html;
- Incluir a propriedade ng-app no elemento html em que queremos “ativar” o AngularJS:

```
<html ng-app>  
  <head>  
    <title>AngularJS</title>  
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/  
/1.4.7/angular.min.js"></script>  
  </head>  
  <body>
```

Temos o uso do ng-app que habilita o AngularJS a atrelar o comportamento das tags html abaixo dele, permitindo assim que possamos utilizar mais propriedades do AngularJS.

DataBind

- Uma forma de ligar automaticamente uma variável qualquer a uma outra;
- Geralmente é usado para ligar uma variável do JavaScript (ou um objeto) a algum elemento do documento HTML;
- Ex:

```

<html ng-app>
  <head>
    <title>Hello World</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.3/angular.min.js"></script>
  </head>
  <body>
    Hello <input type="text" ng-model="yourName"/>
    <hr/>
    <h1>Hello {{yourName}}</h1>
  </body>
</html>
    
```

Controller

- Um controller é, na maioria das vezes, um arquivo JavaScript que contém funcionalidades pertinentes à alguma parte do documento HTML
- Uma forma de sintetizar as regras de negócio (funções javascript) em um lugar separado ao documento HTML

Uso do controller

```
<html ng-app="app">
  <head>
    <title>Hello World</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs\
/1.4.3/angular.min.js"></script>
    <script src="simpleController.js"></script>
  </head>
  <body ng-controller="simpleController">
    Hello <input type="text" ng-model="user.name"/>

    <hr/>
    <h1>Hello {{user.name}}</h1>
  </body>
</html>
```


Controller

- Usam a propriedade ng-controller para dizer que, todo elemento abaixo do <body> será gerenciado pelo controller;
- User é um objeto e name é uma propriedade a este objeto

simpleController.js

```
var app = angular.module('app', []);  
  
app.controller('simpleController', function ($scope) {  
    $scope.user = {name:"Daniel"}  
});
```

Controller

- O parâmetro \$scope é um “ponteiro” para a aplicação em si, ou seja, \$scope significa a própria página html.
- Usa-se o \$scope para criar uma conexão entre o model e a view, como foi feito no exemplo utilizando o objeto user.

Métodos do controller

- O controller é usado também para manipular regras de negócio que podem ou não alterar os models.

Simple exemplo de contador

```
<html ng-app="app">
  <head>
    <title>Hello Counter</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.3/angular.min.js"></script>
    <script src="countController.js"></script>
  </head>
  <body ng-controller="countController">
    <a href="#" ng-click="addOne()">Add 1</a>
    <hr/>
    <p>Counter value: {{counter}}</p>
  </body>
</html>
```

Métodos do controller

- Usamos o controller para definir uma variável e um método para incrementar em 1 esta variável.

countController.js

```
var app = angular.module('app', []);

app.controller('countController', function ($scope) {
    $scope.counter = 0;

    $scope.addOne = function(){
        $scope.counter++;
    }
});
```

Loops

- Outra característica do AngularJS é utilizar templates para que se possa adicionar conteúdo dinâmico;
- Um loop é sempre realizado através da propriedade ng-repeat e obedece a uma variável que geralmente é um array de dados

Loops

Usando loops

```

<html ng-app="app">
  <head>
    <title>Hello Counter</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.3/angular.min.js"></script>
    <script type="text/javascript">
      var app = angular.module('app', []);
      app.controller('loopController', function ($scope) {
        $scope.fruits = ['banana', 'apple', 'orange'];
      });
    </script>
  </head>
  <body ng-controller="loopController">
    <ul>
      <li ng-repeat="fruit in fruits">{{fruit}}</li>
    </ul>
  </body>
</html>

```

Formulários

- Existem diversas características que um formulário contém, tais como validação, mensagens de erro, formato dos campos, entre outros;
- Usamos o AngularJS de diferentes formas, e usamos vários parâmetros ng para controlar todo o processo

Formulário com validação

```
<html ng-app>
<head>
  <title>Simple Form</title>
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4\
.3/angular.min.js"></script>
</head>
<body>

  <form name="myForm">
    <span ng-show="myForm.$invalid">
      Found erros in the form!
    </span>
    <input type="text" ng-model="name" name="Name" value="Your N\
ame" required/>
    <button ng-disabled="myForm.$invalid"/>Save</button>
  </form>

</body>
</html>
```

- ng-show irá exibir ou não a tag contendo a mensagem de erro do formulário;
- ng-disabled desativa o botão de submissão do formulário.

- `myForm.$invalid` é um recurso do AngularJS que define se um formulário está inválido ou não, como usamos uma caixa de texto com a propriedade `required`, se o campo não estiver preenchido, o formulário ficará inválido.

Atividade

- Crie duas páginas HTML simulando autenticação de um usuário no servidor. O usuário deverá preencher um cadastro (utilizando os recursos já conhecidos) e sendo validado os dados no formulário (pesquise outras validações pelo Angular). Após isso, deverá ser exibido na tela mensagem de bem-vindo, com o nome do usuário.