

Pós-graduação em Desenvolvimento Web e Aplicativos Móveis

AngularJS

Fábio Rodrigues Jorge
fabinhojorgenet@gmail.com



<https://github.com/fabinhojorge/aula-javascript>



Angular JS



Introdução



- Criado e mantido pelo Google
- Deis de 2009
- Estende o HTML (Linguagem Declarativa)
- Objetivo: Fácil de aprender e que funcione
- Two way data bind

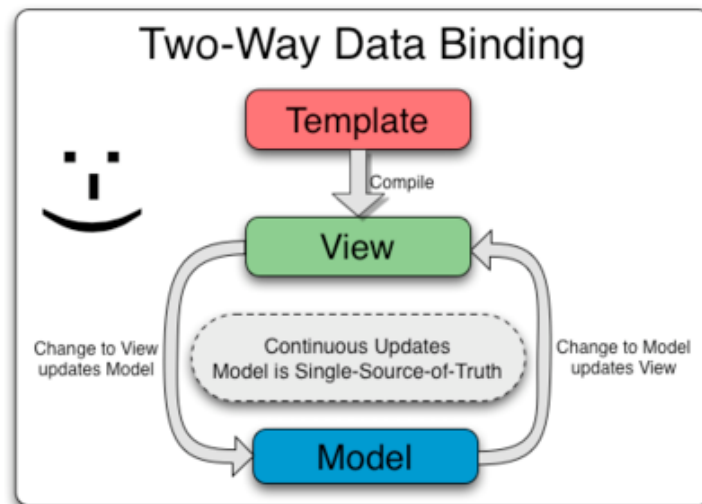
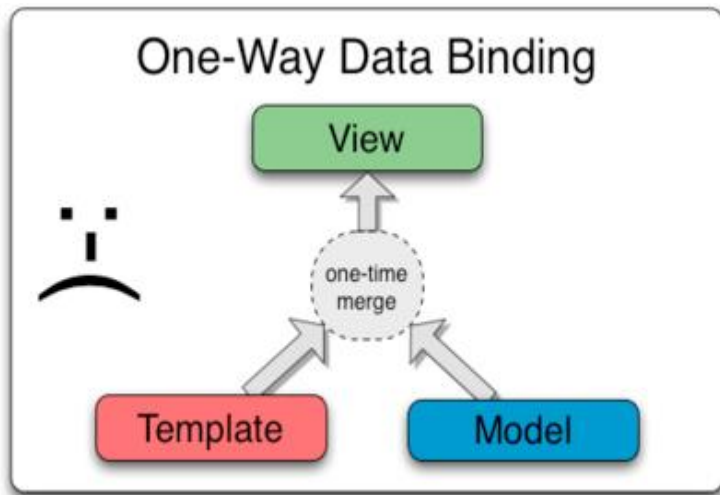
Introdução



- Cria facilmente SPA
- Programação Declarativa
- Modelo MVW* -> (Model View Whatever)
- Angular-UI (<http://angular-ui.github.io/>)
- Outros módulos (<http://ngmodules.org/>)

Introdução

Two Way Data Bind



E a performance?

Hello World

Algumas **Diretivas**

ngApp -> Delimita o seu programa

ngController -> Delimita onde o Controller vai atuar.
Onde o \$scope é compartilhado. Liga a ***View*** e o ***Controller***

angular.module("meuApp", []); -> cria

angular.module("meuApp"); -> carrega

[angularjs/helloWorld.html](https://angularjs.org/helloWorld.html)

Diretivas

Diretivas

As diretivas estendem o HTML: **ng-xxx=" "**

São Atributos adicionados nas tags HTML que fazem a modificação e o dinamismo desses elementos

Ao carregar, o framework percorre todo o documento procurando por essas Diretivas.

Diretivas

ngBind

Pega as variáveis no **Controller** e leva as variáveis para a **View**. Usa o objeto **\$scope**

Pode ser usado praticamente em qualquer componente

- Por atributo: ng-bind="variavel"
- Por chaves: {{variavel}}

Diretivas

ngRepeat

Itera sobre os Arrays e objetos. Repete o elemento *Array.length* vezes

```
var items = ["Calculadora", "Cerveja", "Quadro"];
```

```
ng-repeat="item in items"
```

```
ng-repeat="(key, item) in items"
```

Diretivas

ngModel

Pega as variáveis na ***View*** e leva para o ***Controller***

O contrário de ngBind*

Funciona em 3 elementos:

<input>, ***<select>*** e ***<textarea>***

*Two Way Data Bind

Diretivas - Eventos

ngClick

Eventos em Angular começam com **ng-** e são seguidos pelo evento, exemplo:

ng-focus="função()"

ng-keyup="função()"

ng-click = "função()"

Diretivas

ngOptions

Semelhante ao ngModel, porém foi criado exclusivamente para trabalhar com o <select>

Oferece diversos recursos para as opções (<options>) e criação de grupos no próprio select

Diretivas

ngDisabled, ngHide, ngShow, ngIf

Se receberem retorno ***true*** fazem alguma coisa:

ngDisabled – desabilita o elemento (atributo disabled)

ngHide e **ngShow** – Escondem ou mostram o elemento

ngIf – Cria ou destrói o elemento (manipula o DOM)

Diretivas

ngStyle e ngClass

Estilos dos componentes

```
... $scope.classe = "alert";
```

```
ng-class="classe"
```

```
ng-class="{classe: true}"
```

```
ng-style="{ 'background-color': 'red' }"
```

Diretivas

ngInclude

Incluí na página atual, utilizando comunicação assíncrona, outras páginas e scripts.

Usado para adicionar e montar o template da Aplicação

```
<div ng-include=" 'footer.html' ">
```


Diretivas – Validação de Form

ngRequired e ngPattern

Trabalha com a variável de \$scope que foi criada no formulário. Consegue validar de forma melhor

`ngRequired = "true"`

`ngPattern = "/regexp/"`

Diretivas – Validação de Form

\$valid, \$invalid, \$pristine e \$dirty

São variáveis de controle do formulário.

formName.\$valid -> se **true** o *form* é válido

formName.\$invalid -> se **true** o *form* é inválido

formName.nomeCampo.\$pristine -> se **true** o campo é virgem

formName.nomeCampo.\$dirty -> se **true** o campo não é virgem

Diretivas

\$error

Mostra os erros do formulário em um formato hierárquico.

Tipos de erros de forma hierárquica:

1º required

2º pattern, maxlength, minlength

Vamos inserir um módulo

Module ngMessages

```
angular.module("listaTelefonica", ['ngMessages']);
```

Esse módulo pode ser usado para manipular as mensagens de \$erro do form. Faz um switch entre as mensagens.

ng-messages e ng-message

Filters

Filters

A primeira coisa a entender é que filtro não necessariamente filtra.

Ele transforma os dados de alguma maneira. Alguns filtros:

- uppercase e lowercase
- date
- json
- filter
- orderBy

Pacote PT-BR:

<https://github.com/renato/bower-angular-locale-pt-br>

Filters

É possível criar seus filtros também

```
angular.module("listaTelefonica").filter("capitalizeString", function(){  
    return function(input){  
        //Alguma coisa  
        return output;  
    }  
});
```

```
{{ "João PAULO GomES" | capitalizeString }}  
{{ nome | capitalizeString }}
```

Organizar o projeto

Organização

Sugestão:

```
>app/  
  >css/  
    >app.css  
  >js/  
    >controllers/  
      >loginCtrl.js  
      >listaCtrl.js  
    >directives/  
      >panelDirective.js  
      >tableDirective.js  
    >services/  
      >loginService.js  
      >listaService.js  
    >app.js  
  >lib/  
    >angularjs.js  
  >view/  
    >login.html  
    >lista.html  
>index.html
```

Ajax e o serviço \$http

Antes uma pausa...

Para falarmos da politica da mesma origem [Segurança]

Soluções:

- JSONP
- CORs

Para que serve?

- Acessar suas APIs que estão em outros destinos

\$http

O **\$http** é o serviço de comunicação assíncrona do Angular. Permite usar **XMLHttpRequest** em alto nível.

```
a").controller("listaTelefonicaCtrl", function($scope, $http.
```

- get(url, config);
- post(url, data, config);
- put(url, data, config);
- delete(url, config);
- head(url, config);
- jsonp(url, config);

Exemplo de \$http – Buscar CEP

Tente 17501181

cep - 17501-181

logradouro - Avenida Sampaio Vidal

complemento - (Via Expressa) - até 456/457

bairro - Bosque

localidade - Marília

uf - SP

unidade -

ibge - 3529005

gia - 4388

[https://viacep.com.br/ws/\[CEP:8dig\]/json](https://viacep.com.br/ws/[CEP:8dig]/json)

Serviços

Serviços

Os serviços são bons lugares para aplicar a regra de negócio da aplicação.

- Reuso de código. Melhora os testes.

Existem vários tipos de serviços:

- **Value** - Constantes
- **Factory** - Se parece com os Singletons
- Services
- Providers

Values

São constantes para seu projeto. Pode-se fazer um arquivo de configuração e carregar esse arquivo em values.

```
appCep.value("config", {  
    "cepUrl": "https://viacep.com.br/ws/",  
    "tipoRetorno": "/json"  
});
```


Factories

A Factory é um serviço que retorna um objeto. Elas são Singletons (Lembra?).

```
app.factory("nomeService", function(){  
    return { };  
});  
  
app.controller("nomeController", function($scope, nomeService){}
```

**Angular é legal,
Mas e as desvantagens?**

Desvantagens

- Two Way data bind
- Performance
- Organização do projeto
- Controllers com muito código

-> Angular 2

Dúvidas?



Referências

- Rodrigo Branas. “***AngularJS Essentials Starting***”. Editora PACKT PUBLISHING. Agosto de 2014.
- Daniel Schmitz e Douglas Lira. “***AngularJS na prática***”. Editora Leanpub. Julho de 2015

Links úteis

- **Pacotes de línguas pt-br** (<https://github.com/renato/bower-angular-locale-pt-br>)
- **Comparação de service, Factory e Provider**
(<http://stackoverflow.com/questions/15666048/angularjs-service-vs-provider-vs-factory>)
- **Angular-UI** (<http://angular-ui.github.io/>)
- **Outros módulos** (<http://ngmodules.org/>)