AngularJS

A ponte entre a web de hoje
e a web de amanhã

Aula 2

Exercício em casa

http://bit.ly/Aula31

data-ng-app

AULA 3

AngularJS Really In Depth

http://bit.ly/1Jiytky

angular.config()

angular.config

```
app.config(function() {
    // execute suas configurações aqui
});
```

angular.config

```
app.config(function($httpProvider) {
    $httpProvider.defaults.headers.Authorization = 'Basic ...';
});
```

angular.config

```
angular.module('app')
  .config(function($routeProvider) {
    $routeProvider
      .when('/404', {
        controller: 'LocationController',
        templateUrl: '/views/404.html'
      })
      .otherwise({
        redirectTo: '/404'
      });
});
```

angular.config

```
angular.module('app')
  .config(function($routeProvider, $httpProvider) {
    // $routeProvider...
    // $httpProvider
  });
```

ANGULAR.CONFIG()

- Este bloco é executado durante o registro dos providers e na fase de configuração.

- Apenas constantes e providers podem ser injetados.

- Isto previne que serviços sejam instanciados antes de serem totalmente configurados.

angular.run()

angular.run

```
app.run(function() {
    // faça alguma coisa aqui
});
```

```
angular.module('app', [])
    .value('person', {
        'name': "",
        'age': ""
    })
.run(function(person) {
    person.name = 'Luís Dalmolin';
    person.age  = 24;
})
```

```
angular.module('app', [])
    .service('TaskService', function() {
        return {
            init: function() {...}
        };
    })
    .run(function(TaskService) {
        TaskService.init();
    });
```

ANGULAR.RUN()

Executado após tudo estar configurado

Apenas instâncias podem ser injetadas

Podemos criar quantos blocos run() quisermos

O código necessário para nossa aplicação funcionar

Angular.config e Angular.run são executados após values e constants serem registrados.

http://bit.ly/1A3m3Lz

Services

https://docs.angularjs.org/guide/services

ANGULARJS SERVICES

- Lazy instantiated
- Singletons

PODEM SER UTILIZADOS EM:

Controllers, Services, Filters e Directives

AngularJS - Criando um novo serviço

```
app.service('MyService', function() {
    return {
        myFunction: function() {
            alert('Alerta de dentro do serviço');
        }
    };
});
```

AngularJS - Utilizando um serviço criado

```
app.controller('MainCtrl', function($scope, MyService) {
    MyService.myFunction();
});
```

AngularJS - Services

```
app.factory('MyService', function() {
    return {
        myFunction: function() {
            alert('Factories e Services são (quase) a mesma coisa');
        }
    }
});
```

Javascript “classes”

```
var Person = function() {
    this.name = '...';

    this.setName = function(name) {
        this.name = name;
    };
};

var luis = new Person();
luis.setName('Luís');
```

Services

Retorna uma nova instância

return new Person();

Factories

Retorna o retorno (WTF?)

return Person();

http://bit.ly/1d09beR

Classes no ES6

```
class Person {
    constructor(name) {
        this.name = name;
    }

    setName(name) {
        return this.name = name;
    }
}
```

AngularJS Services - Registrando uma factory

```
app.factory('MyFactory', function($http) {
    return {
        function1: function() {
            return '...';
        },

        function2: function() {
            return '...';
        }
    };
});
```

Objeto literal JS

AngularJS Services - Registrando um service

```
app.service('MyService', function($http) {
    this.function1 = function() {
        return '...';
    };

    this.function2 = function() {
        return '...';
    };
});
```

EXEMPLOS DE IMPLEMENTAÇÕES DE SERVICES E FACTORIES

http://bit.ly/1PJFhHD

Nossos controllers devem ser o menor possível

http://toddmotto.com/rethinking-angular-js-controllers/

Agora é com vocês

NO NOSSO TO-DO APP, FAÇAM:

- Nosso controller deve utilizar um service (ou uma factory) para buscar as tarefas;
- O service $http deve ser utilizado de dentro do nosso service;
- Uma promise deve ser retornada do serviço, mas chamada de dentro do controller.

http://bit.ly/1PJFhHD

http://bit.ly/Aula31

Promises

```
$http.get('https://api.github.com/users/luisdalmolin')
      .success(function(data) {


      })
      .error(function(error) {


})
.finally(function() {
    alert('Finalizado');
});
```

Promises

```
$http.get('...').then(
    function(data) {...},   // sucesso
    function(error) {...}, // error
);
```

Promises

```
$http.get('...').then(
    function() {...}, // sucesso
    function() {...}, // erro
    function() {...}  // sempre é disparada
);
```

$http promises

```
$http.get('tasks.json').then(
    function(result) {
        console.log(result);
    }
);
```

```
▶ config: Object
▶ data: Array[6]
▶ headers: function (name) {
  status: 200
  statusText: "OK"
▶ __proto__: Object
```

$q

Criando nossas próprias promises

AngularJS $q

```
app.factory('MyFactory', function($q) {
    return {
        getItems: function() {
            var deferred = $q.defer();
            if (success) {
                return deferred.resolve('Retorna os itens');
            } else {
                return deferred.reject('Erro na promise!');
            }
            return deferred.promise;
        }
    };
});
```

Angular $q

```
app.controller('MainCtrl', function(MyFactory) {
    MyFactory.getItems().then(
        function(data) {
            console.log('sucesso!', data);
        },
        function(data) {
            console.log('erro!', data);
        },
        function(data) {
            console.log('sempre?', data);
        }
    );
});
```

Angular $q

```
var deferred = $q.defer();

window.setInterval(function() {
    deferred.notify('Notificação');
}, 100);
```

$q - Exemplo de uso

http://bit.ly/AngularQ

NO NOSSO TO-DO APP, FAÇAM:

- Nosso serviço, ao invés de retornar a promise do $http, deve retornar uma promise utilizando o serviço $q.

http://bit.ly/AngularQ

http://bit.ly/Aula32

HTTP Interceptors

https://docs.angularjs.org/api/ng/service/$http

HTTP Interceptors

```
app.factory('MyHttpInterceptor', function() {
    return {
        request: function(config) {},
        requestError: function(error) {},

        response: function(response) {},
        responseError: function(response) {},
    }
}]);
```

HTTP Interceptors

```
.config(function($httpProvider) {
    $httpProvider.interceptors.push('MyHttpInterceptor');
})
```

HTTP Interceptors

```
.config(function($httpProvider) {
    $httpProvider.interceptors.push('MyHTTPInterceptor');
})
.factory('MyHTTPInterceptor', function() {
    return {
        request: function(config) {
            config.url = config.url.replace('luisdalmolin', 'magnobiet');

            return config;
        }
    };
});
```

HTTP Interceptors

```
.factory('MyHTTPInterceptor', function(user) {
    return {
        request: function(config) {
            if (user.token) {
                config.headers.Authorization = user.token;
            }
            return config;
        }
    };
});
```

HTTP Interceptors

```
.factory('MyHTTPInterceptor', function(HTTPLogService) {
    return {
        respose: function(response) {
            console.log(response.status);
            console.log(response.data);
            console.log(response.headers);

            HTTPLogService.log(response);

            return response;
        }
    };
});
```

HTTP Interceptors

http://bit.ly/AngularInterceptors

Intervalo

Criando nossos próprios filtros

Criando filtros no Angular

```
app.filter('substr', function() {
    return function(input) {
        // façamos qualquer coisa aqui

        return input;
    };
})
```

```
{{ "Curso AngularJS" | substr }}
```

Criando filtros no Angular

```
app.filter('substr', function() {
    return function(input, limit, start) {
        limit = limit || 5;
        start = start || 0;

        return input.substr(start, limit);
    };
})
```

Criando filtros no Angular

```
{{ "Curso AngularJS" | substr:5:1 }}
```

```
<!-- urso -->
```

Criando filtros

http://bit.ly/AngularFiltros2

NO NOSSO TO-DO APP, FAÇAM:

- No final da nossa tabela de listagem de tarefas, vamos adicionar uma novo linha com o total de tarefas finalizadas;
- Devemos utilizar um filtro para isto;

bit.ly/AngularFiltros2

bit.ly/Angular33

Curiosidade

```
<a ng-href="/users/{{ username }}">{{ username }}</a>
```

Custom Directives

https://docs.angularjs.org/guide/directive

<div nossa-diretiva>

<nossa-diretiva></nossa-diretiva>

<div data-nossa-diretiva>

Custom Directives

```
.directive('totalDeTarefas', function() {
    return {
        template: '{{ tasks | sum_finalized }} tarefa(s) finalizadas |
            {{ tasks | sum_incomplete }} tarefa(s) não conluídas';
    };
});
```

```
<total-de-tarefas></total-de-tarefas>
```

Custom Directives

```
.directive('totalDeTarefas', function() {
    return {
        templateUrl: 'total-de-tarefas.html';
    };
});
```

```
<total-de-tarefas></total-de-tarefas>
```

Custom Directives

```
.directive('totalDeTarefas', function() {
    return {
        restrict: 'E',
        templateUrl: 'total-de-tarefas.html';
    };
});
```

```
<total-de-tarefas></total-de-tarefas>
```

Custom Directives

```
.directive('totalDeTarefas', function() {
    return {
        restrict: 'A',
        templateUrl: 'total-de-tarefas.html';
    };
});
```

```
<div total-de-tarefas></div>
```

Custom Directives

```
.directive('totalDeTarefas', function() {
    return {
        restrict: 'C',
        templateUrl: 'total-de-tarefas.html';
    };
});
```

```
<div class="total-de-tarefas"></div>
```

Custom Directives

```
.directive('totalDeTarefas', function() {
    return {
        restrict: 'AEC',
        templateUrl: 'total-de-tarefas.html';
    };
});
```

Legibilidade

Reutilização

Componetização

<input date-picker />


jQuery
write less, do more.

Custom Directives - link function

```
.directive('datePicker', function() {
    return {
        link: function(scope, element, attrs) {
            element.datepicker();
        }
    }
});
```

Custom Directives

bit.ly/AngularDirectives

NO NOSSO TO-DO APP, FAÇAM:

- Vamos criar 2 diretivas:

- Uma para selecionar a data do prazo com o jQuery UI datepicker (utilizem o exemplo pra se basear);

- Uma para encapsular toda a linha de listagem da tarefa (Utilizem como um atributo do <tr> para facilitar;

bit.ly/AngularDirectives

bit.ly/Angular34

Custom Directives

<script type="text/ng-template" >

<script type="text/ng-template">

```
<script type="text/ng-template" id="hello.html">
    <h1>Olá, {{ name }}</h1>
</script>
```

```
return {
    templateUrl: 'hello.html'
}
```

Custom Directives

replace: true

replace: true

```
app.directive('hello', function() {
    return {
        replace: true,
        template: '<h1>Olá, {{ name }}<h1>'
    };
});
```

```
<hello><h1>Olá, Luís</h1></hello> <!-- replace: false (padrão) -->
<h1>Olá, Luís</h1>                <!-- replace: true -->
```

Custom Directives
replace & templates

bit.ly/AngularDirectives2

Custom Directives

Isolated Scopes

```
return {
    scope: true
};
```

```
return {
    scope: {}
};
```

Isolated Scopes

Expressions (&)

Isolated Scopes - Expressions

```
app.directive('task', function() {
    return {
        scope: {
            done: '&'
        }
    };
});
```

```
<!-- template da diretiva -->
<button ng-click="done()"></button>

<!-- usando a diretiva (finalized é o método do ctrl) -->
<task done="finalized()"></task>
```

Isolated Scopes - Expressions + Parâmetros

```
<!-- template da diretiva -->
<input ng-model="task">
<button ng-click="done({task: task})"></button>
```

```
<!-- usando a diretiva (finalized é o método do ctrl) -->
<task done="finalized(task)"></task>
```

Isolated Scopes - Expressions + Parâmetros

```
app.controller('MainCtrl', function($scope) {
    $scope.finalized = function(task) {
        alert(task);
    };
});
```

Isolated Scopes

Two way bindings (=)

Isolated Scopes - Two-way bindings

```
app.directive('taskForm', function() {
    return {
        scope: {
            title: '='
        }
    };
});
```

```
<!-- No controller: $scope.task = {title: ''}; -->
<task-form title="task.title"></task-form>
```

Isolated Scopes - Two-way bindings

```
<!-- Template da diretiva -->
<input ng-model="title" />
```

```
<!-- Template do controller -->
<input ng-model="task.title" />
```

Isolated Scopes

Attributes (@)

Isolated Scopes - attributes

```
<!-- Utilizando a diretiva -->
<task-list title="Curso AngularJS"></task-list>
```

```
<!-- Template da diretiva -->
<h1>{{ title }}</h1>
```

Isolated Scopes - attributes

```
app.directive('task', function() {
    return {
        title: '@'
    };
});
```

Isolated Scopes - attributes

```
app.directive('task', function() {
    return {
        title: '@taskTitle'
    };
});
```

```
<task task-title="Custom Directives"></task>
```

Custom Directives
Isolated Scopes

bit.ly/AngularDirectives3

Todo mundo aqui ainda?

Custom Directives - Dependency Injection

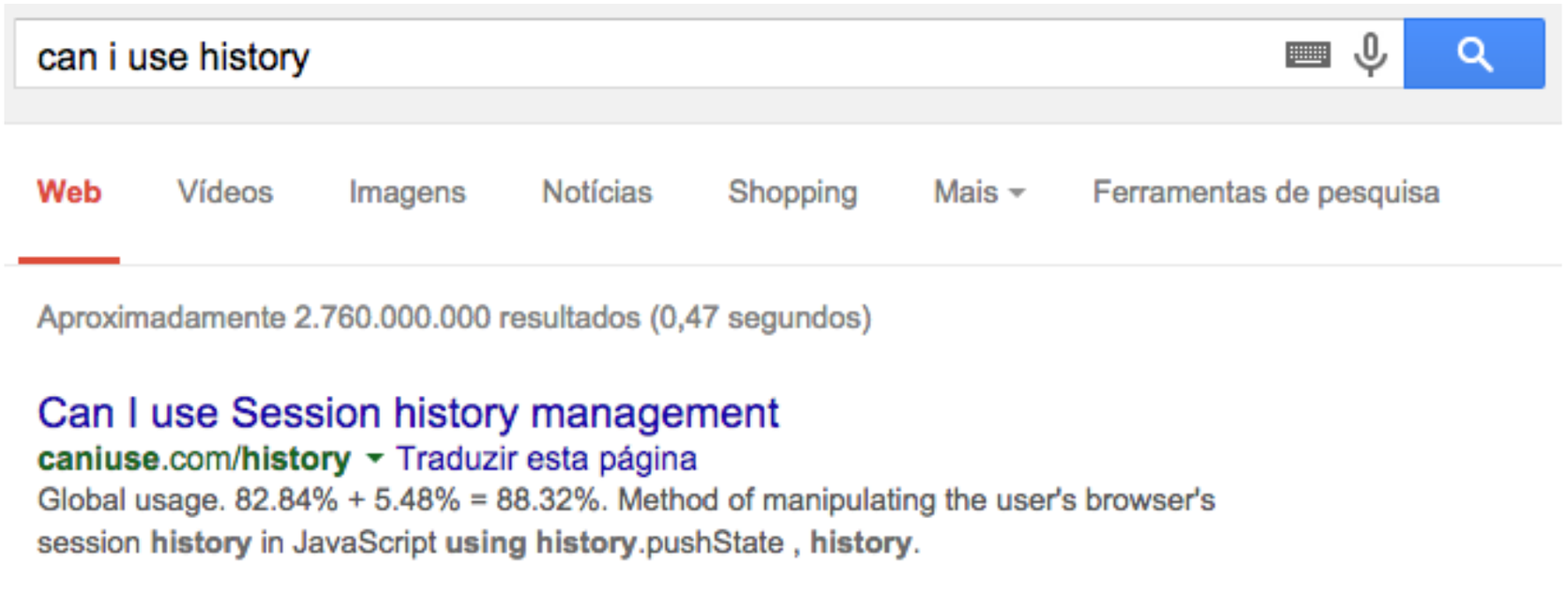
```
app.directive('task', function($http) {
    return {
        template: '<div ng-repeat="task in tasks"></div>'
        link: function(scope, element, attrs) {
            $http.get('...').then(function(tasks) {
                scope.tasks = tasks;
            });
        }
    };
});
```

Custom Directives

```
return {
    controller: function($scope) {},
    require: '^tabsPane',
    transclude: true
};
```

Can i use HTML5?

http://caniuse.com/



https://github.com/angular/angular.js

Todo mundo aqui ainda?

$scope vs scope

RECAP

Como criar serviços

HTTP Interceptors

$q

Como criar filtros

O QUE VIMOS HOJE

Como criar diretivas

services vs factories

data-ng-app

angular.run & angular.config

PRÓXIMAS SALAS

27/05/2015 - 602 Multicolor

03/06/2015 - 602 Multicolor