AngularJS

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

Aula 1

https://builtwith.angularjs.org/

https://docs.angularjs.org/api/ng/type/$rootScope.Scope

PROPRIEDADES DE UM ESCOPO

$scope.$id
$scope.$root
$scope.$parent

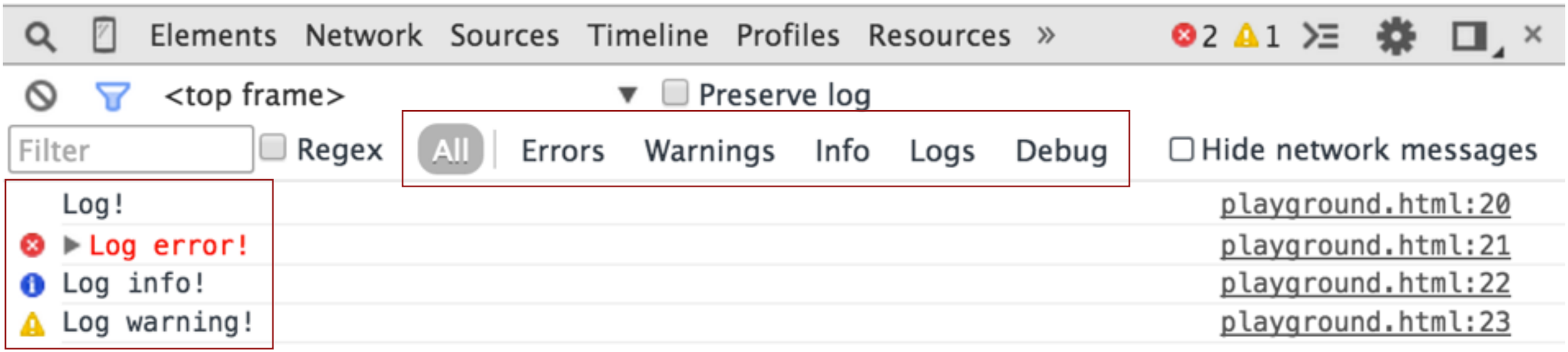
console.error()

console.info()

console.log()

console.warn()

https://developer.mozilla.org/en-US/docs/Web/API/Console/log


Elements Network Sources Timeline Profiles Resources »
2 1
<top frame>
Preserve log
Filter
Regex
All Errors Warnings Info Logs Debug
Hide network messages
Log!
playground.html:20
Log error!
playground.html:21
Log info!
playground.html:22
Log warning!
playground.html:23

HTML

Tipos de requisições HTTP

GET

POST

PUT

DELETE

Todo app

AULA 2

AngularJS In Depth

Eventos

JÁ VIMOS ALGUMAS FORMAS

ng-click, ng-change

$scope.$watch

Event-driven Programming

Emitir eventos

(quando determinada coisa acontece)

Receber eventos

(quando eles são emitidos)

```php
Event::fire('evento', ['parametros']);

Event::listen('evento', function($parametros) {
    // faça alguma coisa aqui
});
```

$scope.$on

$rootScope.$on

$scope.$emit

$rootScope.$emit

$scope.$broadcast

$rootScope.$broadcast

Escutando por eventos

```
app.controller('MainCtrl', function($scope) {
    $scope.$on('evento', function(e) {

    });
});
```

Debug do objeto recebido do evento

```
console.log(e);
```

```
  currentScope: null
  defaultPrevented: false
  name: "evento"
▶ preventDefault: function () {
▶ targetScope: Object
▶ __proto__: Object
```

https://docs.angularjs.org/api/ng/type/$rootScope.Scope

Emitindo eventos

```
$scope.$emit('evento');
```

$scope.$emit e $scope.$on

```
app.controller('MainCtrl', function($scope) {
    $scope.$on('evento', function(e) {
        // aqui podemos fazer qualquer coisa
    });

    $scope.$emit('evento');
});
```

$rootScope.$emit

```
app.controller('MainCtrl', function($scope, $rootScope) {
    $scope.$on('evento', function(e) {
        // aqui podemos fazer qualquer coisa
    });

    $rootScope.$emit('evento');
});
```

```
app.controller('MainCtrl', function($scope, $rootScope) {
    $rootScope.$on('evento', function(e) {
        // aqui podemos fazer qualquer coisa
    });

    $scope.$emit('evento');
});
```

Parar a propagação dos eventos

```
app.controller('ParentCtrl', function($scope, $rootScope) {
    $scope.$on('evento', function(e) {
        e.stopPropagation();
    });

    $rootScope.$on('evento', function(e) {
        // este evento nunca vai ser recebido
    });
});
```

```
$scope.$emit('evento');
```

https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$emit

https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$on

PODEMOS CONCLUIR QUE:

O $emit envie o evento para o escopo atual
e para todos os escopos acima dele (pais).

Emitindo eventos: Do pai para o filho

```
// controller pai
app.controller('ParentCtrl', function($scope, $rootScope) {
    $scope.$on('evento', function(e) {
        // aqui podemos fazer qualquer coisa
    });
});

    // controller filho
    app.controller('ChildrenCtrl', function($scope) {
        $scope.$emit('evento');
    });
```

E o contrário?

$scope.$broadcast(‘evento’)

https://docs.angularjs.org/api/ng/type/$rootScope.Scope#$broadcast

Fazendo $broadcast do pai para o filho

```
// controller pai
app.controller('ParentCtrl', function($scope, $rootScope) {
    $scope.$broadcast('evento');
});
```

```
// controller filho
app.controller('ChildrenCtrl', function($scope, $rootScope) {
    $scope.$on('evento', function(e) {
        console.log('Recebendo evento $broadcasted');
    });
});
```

IMPORTANTE

Não enviem eventos logo que o controller é iniciado.
Utilizem algum handler (ng-click, ng-change, $watch)
Ou o serviço $timeout do Angular

Controller pai > Controller filho

```
<div ng-controller="ParentCtrl">
    <h1>Parent Ctrl</h1>

    <div ng-controller="ChildrenCtrl">
        <h2>MainCtrl</h2>
    </div>
</div>
```

Recebendo evento $broadcast no mesmo controller

```
app.controller('ParentCtrl', function($scope, $rootScope) {
    $scope.$on('evento', function(e) {
        console.log('Recebendo evento $broadcasted');
    });

    $scope.$broadcast('evento');
});
```

$rootScope.$broadcast

```
$rootScope.$broadcast('evento');
```

Todos os controllers com um handler registrado ($scope.$on) receberiam este evento.

$rootScope.$emit

```
$rootScope.$emit('evento');
```

Apenas o $rootScope.$on receberia este evento

$rootScope.$on

```
$rootScope.$on('evento');
```

Receberia qualquer evento enviado pelo $emit

Passando parâmetros nos eventos

```
$scope.$emit('evento', 'qualquer conteúdo');
```

```
$scope.$on('evento', function(e, data) {
    console.log(data); // qualquer conteúdo
});
```

Passando parâmetros nos eventos

```
$scope.$emit('evento', {
    title: 'Qualquer tipo de parâmetro',
});
```

```
$scope.$on('evento', function(e, data) {
    console.log(data); // {title: '...'}
});
```

$SCOPE EVENTS

o $scope dos nossos controllers também emite um evento

$scope.$on(“$destroy’)

```
$scope.$on('$destroy', function(e) {
});
```

```
$scope.$destroy();
```

Agora é com vocês

NO NOSSO TO-DO APP, FAÇAM:

- Toda vez que uma tarefa for completa, um evento deve ser emitido.

- Toda vez que uma tarefa for inserida, um evento deve ser emitido.

- O objeto da tarefa deve ser passado como parâmetro para os eventos.

- O input deve ser limpo no evento de tarefa inserida

http://bit.ly/AngularE2

console.log(var1, var2)

Filtros

‘Filtros’ no Linux

```
➜  Code git:(gh-pages) ✗ ls -la | grep Aula
drwxr-xr-x@ 26 luisdalmolin  staff   884 May  7 09:06 Aula1
drwxr-xr-x@  7 luisdalmolin  staff   238 May  8 12:53 Aula2
```

Filtros no Angular

```
<h1>{{ "AngularJS" | uppercase }}</h1>
```

filter currency number date

json lowercase uppercase limitTo

orderBy

Angular Built-in filters

https://docs.angularjs.org/api/ng/filter

Passando parâmetros para os filtros

```
app.controller('MainCtrl', function($scope) {
    $scope.today = new Date();
});
</script>
```

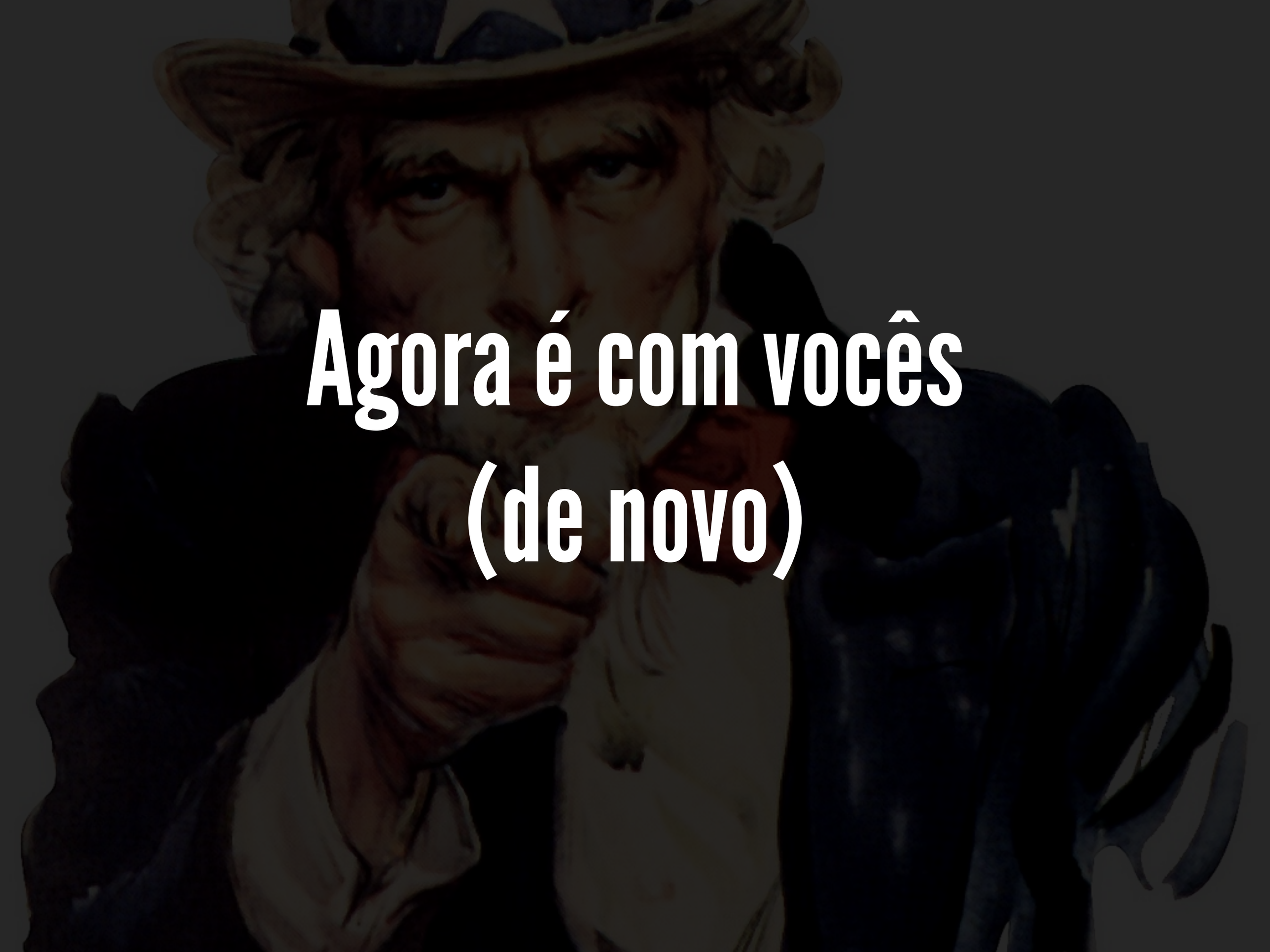
```
<div ng-controller="MainCtrl">
    {{ today | date }}                    <!-- May 13, 2015 -->
    {{ today | date:'d/M/yyyy' }} <!-- 13/5/2015 -->
</div>
```

https://docs.angularjs.org/api/ng/filter/date

{{ Code time }}

show | filters

http://bit.ly/AngularFiltros


Agora é com vocês
(de novo)

NO NOSSO TO-DO APP, FAÇAM:

- Adicionem o campo ‘created_at’ nas tarefas. Ele deve ter o valor da data em que foi criado. (new Date().getTime()).

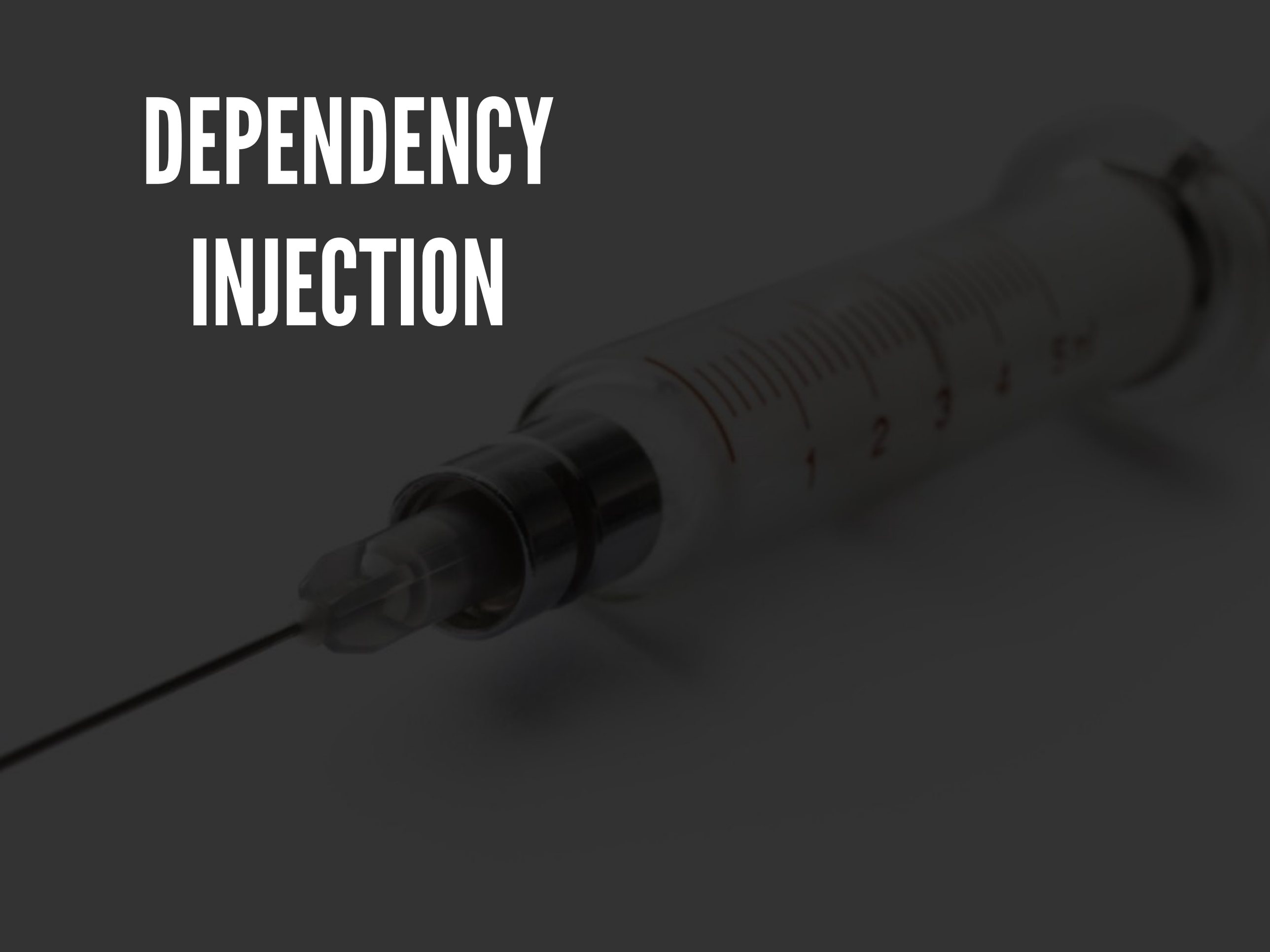
- Adicionem uma busca por tarefas. (filter)

- Adicionem uma ordenação nas tarefas (orderBy) pela data de criação

- Adicionem a data de criação da tarefa na listagem (usando o filtro date)

http://bit.ly/AngularE3

DEPENDENCY INJECTION




Don’t call us
We will call you

```
angular.module('app')
    .controller('MainCtrl', function($scope) {

});
```

```
angular.module('app')
    .controller('MainCtrl', function($scope, $http) {
        $http.get('https://api.github.com/users/luisdalmolin')
            .success(function(data) {
                console.log(data);
            });
    });
```

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

```
app.controller('TaskShowCtrl', function(
    $rootScope, $scope, $stateParams, $state, $filter, SelectMiddleware,
    AttachmentFactory, Task, Job, TaskInteraction, TaskFollower, ValidationErrors)
{
```

Promises

```
$('.my-element').animate({
    left: '100px'
}, 250);
```

```
$('.my-element').hide();
```

```
$('.my-element').animate({
    left: '100px'
}, 250, 'swing', function() {
    $('.my-element').hide();
});
```

```
angular.module('app')
    .controller('MainCtrl', function($scope, $http) {
        $http.get('https://api.github.com/users/luisdalmolin')
            .success(function(data) {
                console.log(data);
            });
    });
```

```
app.controller('MainCtrl', function($scope, $http) {
    $scope.user = {};

    $http.get('https://api.github.com/users/luisdalmolin')
            .success(function(data) {
                $scope.user = data;
            })
            .error(function(error) {
                console.log(error);
            });
});
```

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


        })
        .error(function(error) {


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

angular.constant

angular.constant - declaração

```
var app = angular.module('app')
.constant('APP_ID', '12345');
```

angular.constant - como utilizar

```
app.controller('MainCtrl', function($scope, APP_ID) {
    console.log(APP_ID); // 12345
});
```

angular.constant - declarar objetos

```
.constant('CONFIG', {
    id: 12345,
    url: '...'
});
```

angular.value

angular.value

```
.value('CONFIG', {
    id: 12345,
    url: '...'
});
```

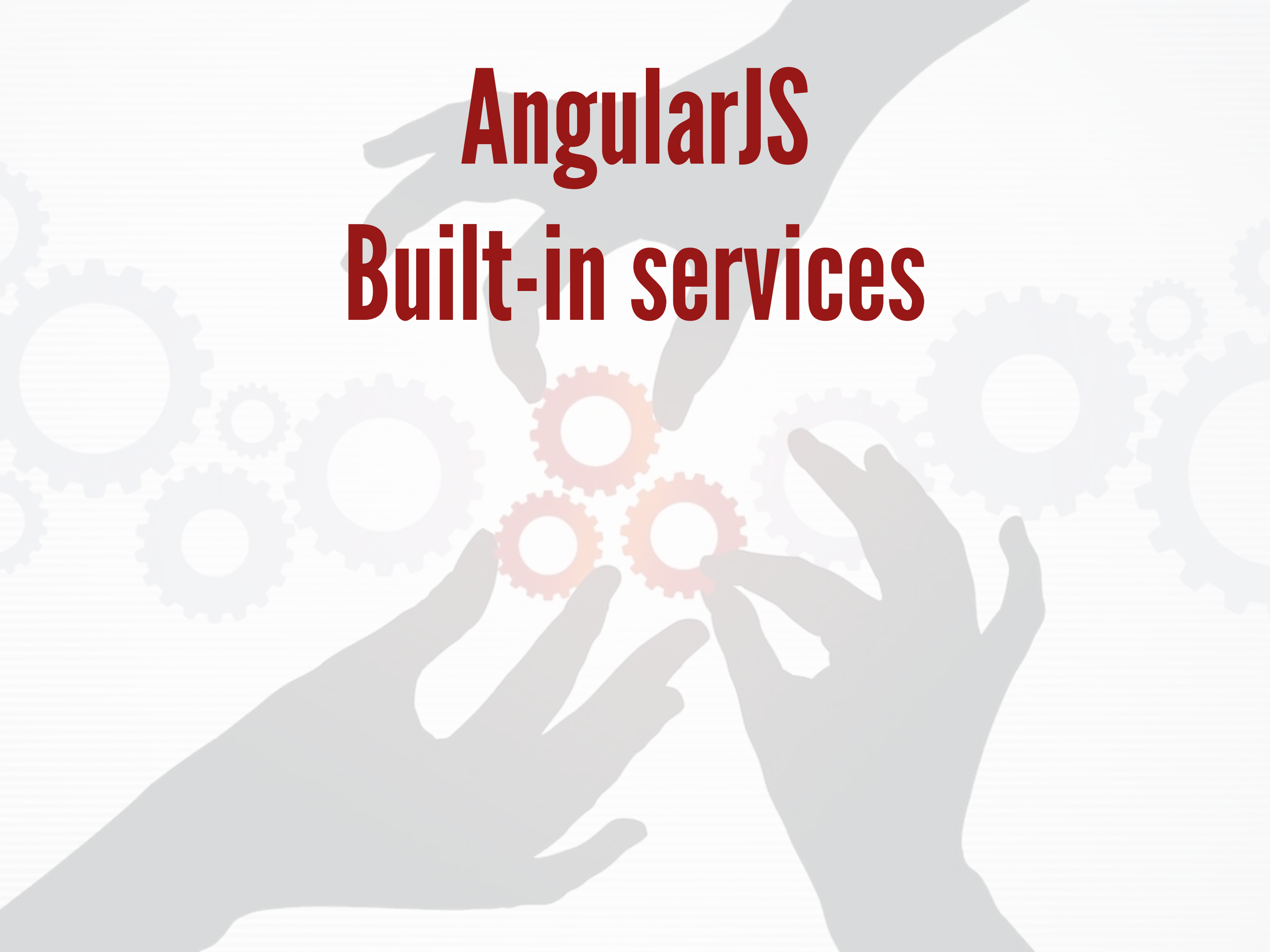
angular.value - alterando valores

```
angular.module('app')
.value('user', {
    id: 12345,
    name: 'Luís Dalmolin'
})
.controller('LoginCtrl', function(user) {
    $scope.login = function() {
        user.id   = response.id;
        user.name = response.name;
    };
});
```

angular.value - valores primitivos

```
angular.module('app')
.value('name', 'Luís Dalmolin')
.controller('LoginCtrl', function(name) {
    $scope.login = function() {
        name = response.name; // NÃO FUNCIONA
    };
});
```

AngularJS
Built-in services



ANGULAR BUILT-IN SERVICES

$filter

$http

$timeout

$log

$location

$window

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

$filter

Built-in services - $filter

```
app.controller('SampleCtrl', function($scope, $filter) {
    $scope.title = $filter('uppercase')('AngularJS');
});
```

http://bit.ly/1cBot9V

$http

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

Built-in services - $http

```
app.controller('SampleCtrl', function($scope, $http) {
    $http.get('https://api.github.com/users/luisdalmolin')
        .success(function(user) {
            $scope.user    = user;
            $scope.loading = false;
        });
});
```

http://bit.ly/1GVPz2K

$http service

```
app.controller('MainCtrl', function($scope, $http) {
    $scope.user = {};

    $http.get('https://api.github.com/users/luisdalmolin')
        .success(function(data) {
            $scope.user = data;
        })
        .error(function(error) {
            console.log(error);
        });
});
```

$http service

```
$http.post('http://api.github.com/...', {content: '...'})
    success(function(data, status, headers, config) {

});
```

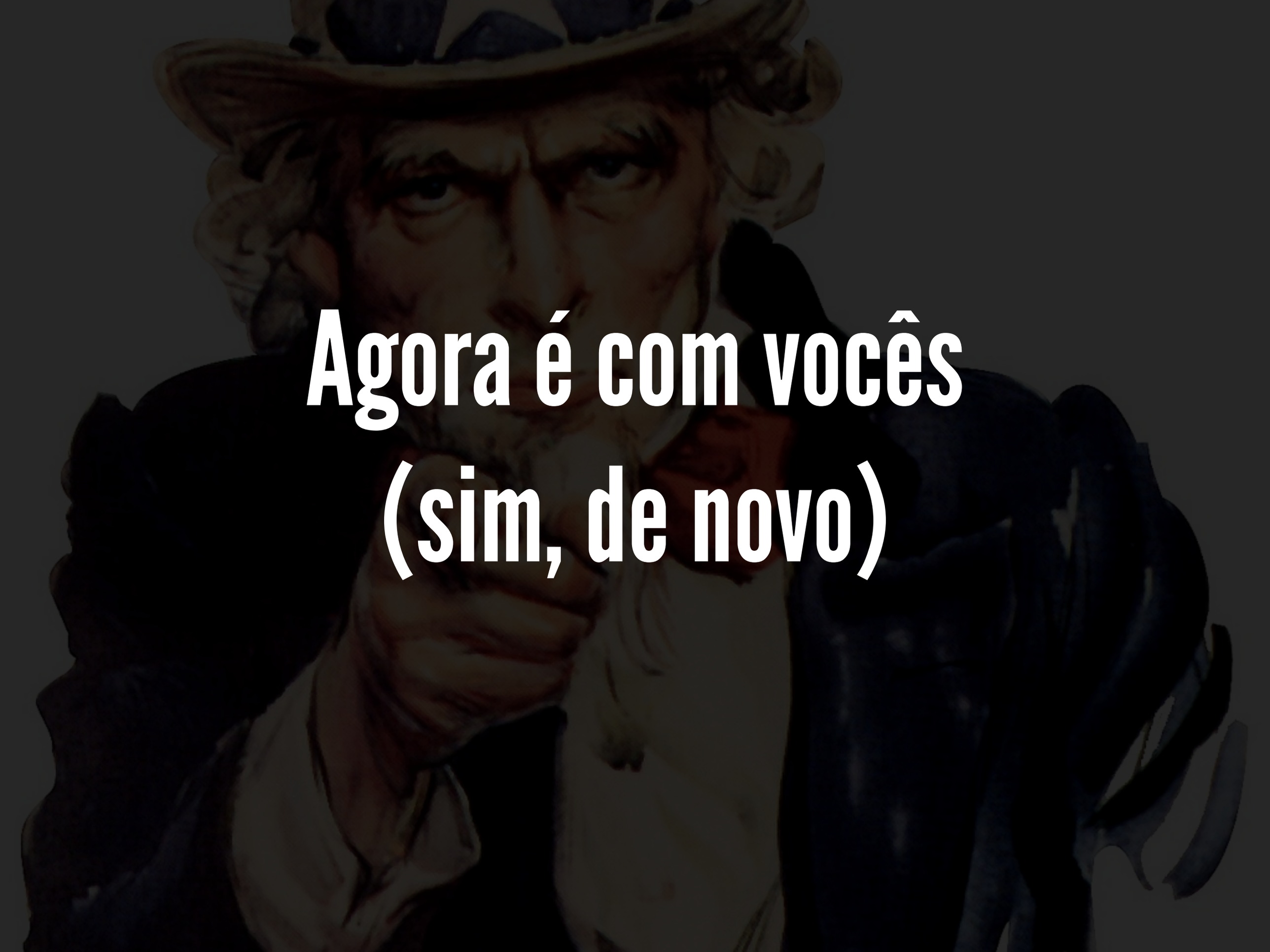
$http service

```
$http({
    url: 'https://api.github.com/users/luisdalmolin'
    method: 'GET',
    params: {...},
    headers: {...}
});
```

Apenas para debug e testes

Alguns serviços, como $log, $window, $location,
servem apenas para propósitos de testes.

Eles apenas fazem referência para o respectivo objeto global.


Agora é com vocês
(sim, de novo)

NO NOSSO TO-DO APP, FAÇAM:

- Vamos utilizar o serviço $http do Angular

- A lista inicial de tarefas, deverá ser buscada na URL http://bit.ly/AngularTasks

http://bit.ly/AngularE4

$httpProvider

$HTTPPROVIDER

Por padrão, toda requisição com o $http possui os seguintes headers:

Todas as requisoções: Accept: application/json, text/plain, * / *

POST/PUT: Content-Type: application/json

$httpProvider

```
app.run(function($httpProvider) {
    // aplicados em TODOS os tipos de requisições
    $httpProvider.defaults.headers.common['Content-Type'] = 'application/json; charset=utf-8';

    // aplicados apenas nas requisições do tipo POST
    $httpProvider.defaults.headers.post['Content-Type']   = 'application/json; charset=utf-8';
});
```

$httpProvider - Basic Auth

```
app.run(function($httpProvider) {
    // aplicados em TODOS os tipos de requisições
    $httpProvider.defaults.headers.common.Authorization =
        'Basic RXNzYSB0dXJtYSDDqSBmb2RhIQ==';
});
```

$http

```
app.controller('MainCtrl', function($http) {
    // aplicados em TODOS os tipos de requisições
    $http({
        // ...
        headers: {
            Authorization: 'Basic RXNzYSB0dXJtYSDDqSBmb2RhIQ=='
        }
        // ...
    });
});
```

$

Por que diabos?

Tudo que é nativo do Angular possui $ na frente
E as diretivas, possuem ng-

$http

$log

$location

$window

$filter

$scope

$rootScope

ng-controller

ng-show

ng-disabled

ng-app

ng-class

ng-model

Dúvidas?

RECAP

ANGULARJS EVENTS SYSTEM

$scope.$on. $emit. $broadcast

BUILT-IN SERVICES

$http, $filter, $log, ...

O QUE VIMOS HOJE

"AngularJS" | uppercase

FILTROS

ANGULAR.CONSTANT E ANGULAR.VALUE

DEPENDENCY INJECTION & PROMISES

```
var app = angular
    .module('KillJobApp', ['ui.router', 'ngResource', 'ui.utils', 'angularFileUpload', 'doowb.angular-pusher', 'ui.select'
    .constant('KillJobConfig', config);

// routes
app.run(function($rootScope, $notification, contentFactory, $authorized, ReloadService, LogoChangedService, OfflineService
{
    $rootScope.$on('$stateChangeStart', function(e, page, current) {
        if (page.subContent) {
            contentFactory.show();
        } else {
            contentFactory.hide();
        }
    });

    $rootScope.$on('content.emitClose', function(e, params)
        contentFactory.hide();

        $rootScope.$emit('content.close', params);
    });

    // Verifica se usuário esta logado
    $authorized.init();
```

Códigos


```
var app = angular
    .module('KillJobApp', ['ui.router', 'ngResource', 'ui.utils', 'angularFileUpload', 'doowb.angular-pusher', 'ui.select'
    .constant('KillJobConfig', config);
```

TEMA DE CASA

No To-Do App:

- Adicionar descrição (description) - No cadastro;
- Adicionar prazo (deadline) - No cadastro;

- Ao clicar em algum botão de visualizar, exiba a descrição e o prazo.

Próxima aula

Criar nossos próprios serviços
Criar nossas próprias diretivas
Rotas e múltiplas páginas
Componentes de terceiros
e mais coisas mais legais ainda.

PRÓXIMAS SALAS

20/05/2015 - 609 Multicolor

27/05/2015 - 602 Multicolor

03/06/2015 - 602 Multicolor