Angular Js Quickstart

Lesson 1

Matteo Scandolo

Frontend Leader @LinkMe
Co-Founder @MeanMilan

Giovanni Lela

Backend Leader @LinkMe
Co-Founder @MeanMilan

Per chi è il corso?

Web Desingers
FE Developers (coming from Html, Css, jQuery)
Developer from other languages

Argomenti che tratteremo

Approccio MVVM tipico di Angular
Two-Way Data Binding
Consumo di risorse REST
Integrazione di componenti

Let's Start!

Cos'è AngularJs?

Superheroic JavaScript MVW Framework

Static or Dynamic SPA (but not only)

Extend the Html

Quick, Expressive and modular

Hybrid Application (Ionic)

Desktop Application (NW,)

I vantaggi di AngularJs

Applicazione Reattive

Sviluppo rapido

Modulare

Testabile

Come funziona il corso?

Poca Teoria (il minimo indispensabile)

Molta Pratica

Domande domande domande!

Creare un applicazione web

che permetta a un fruttivendolo di gestire i suoi prodotti

e a un utente di aggiungere dei prodotti al suo carrello

Come creare un applicazione Angular

Create a file named:

`index.html`

Creare un file Html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>AngularJs QuickStart</title>
</head>
<body>

</body>
</html>
```

Caricare Angular e inizializzare l'applicazione

```
<!DOCTYPE html>
<html lang="en" ng-app>
<head>
  <meta charset="UTF-8">
  <title>AngularJs QuickStart</title>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.3.15/angular.js">
</head>
<body>

</body>
</html>
```

Fatto!

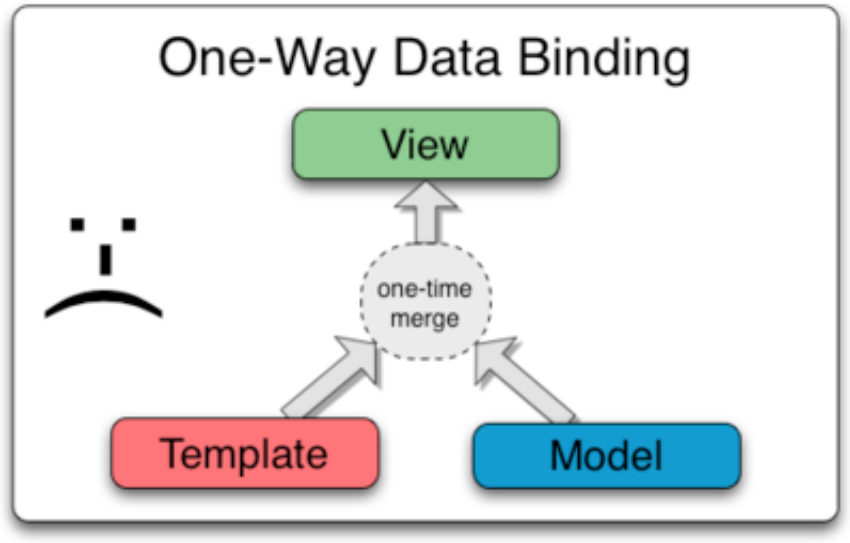
Data Binding

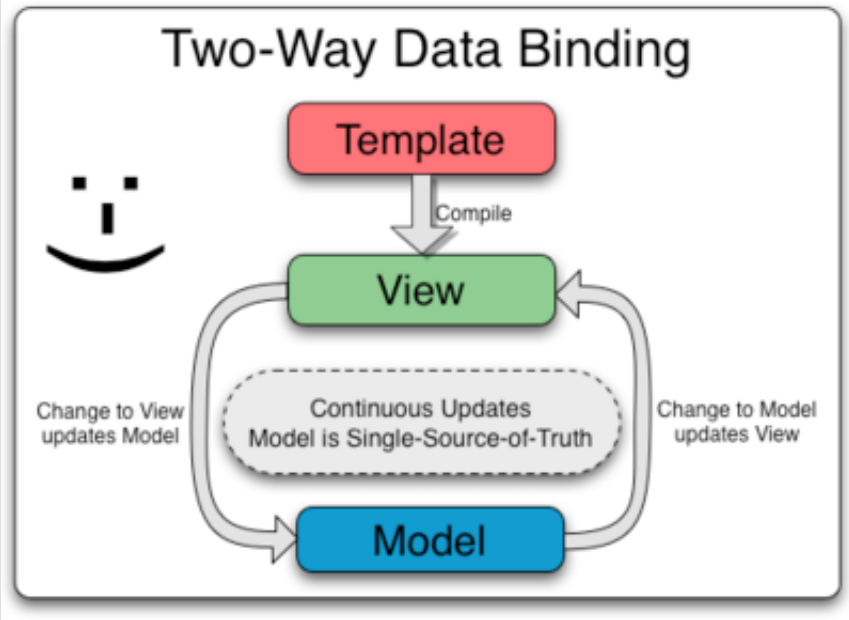
Hi !

```
<input type="text" ng-model="name" placeholder="Insert your name!"/>
<div>Hi {{ name }}!</div>
```

Cos'è il Data-Binding?

In Angular per DataBinding si intende la sincronizzazione automatica del data fra la vista e la definizione del modello.





Feature del Data-Binding

Binding Multipli

Insert your name!

Hi !

How are you ?

```
<input type="text" ng-model="name" placeholder="Insert your name!"/>
<div>Hi {{ name }}!</div>
<div>How are you {{ name }}?</div>
```

Feature del Data-Binding

Binding di espressioni

+

Result is !

```
<input type="number" ng-model="a"/>
<h4>+</h4>
<input type="number" ng-model="b"/>
<h3>Result is {{a+b}}!</h3>
```

Feature del Data-Binding

Assegnazione di valori

Set Value to 3

Value is: 1

```
<button ng-click="value = 3" ng-init="value = 1">Set Value to 3</button>
<h3>Value is: {{value}}</h3>
```

Feature del Data-Binding

Assegnazione di espressioni

Increase Value

Value is: 1

```
<button ng-click="value = value + 1" ng-init="value = 1">Increase Value</button>
<h3>Value is: {{value}}</h3>
```

Feature del Data-Binding

Valutazione di espressioni

Increase Value

Value is: 1

I will be *hidden* if value is greater than 3

```
<button ng-click="value = value + 1" ng-init="value = 1">Increase Value</button>
<h4>Value is: {{value}}</h4>
<h3 ng-hide="value > 3">I will be hidden if value is greater than 3</h3>
<h3 ng-show="value > 3">I will be visible if value is greater than 3</h3>
```

Data-Binding on steroids

Esecuzione di funzioni Javascript

Partecipant name:

Insert Name

Value is: []

```
<input type="text" ng-model="name"/>
<button ng-click="array.push(name)" ng-init="array = []">Insert Name</button>
<h3 style="margin-top: 10px">Value is: {{array}}</h3>
```

Data-Binding on steroids

Repeater

Partecipant name:

Insert Name

```
<input type="text" ng-model="name"/>
<button ng-click="array.push(name)" ng-init="array = []">Insert Name</button>
<h2 ng-repeat="name in array">{{name}}</h2>
```

Data-Binding on steroids

Filtering

Partecipant name:

Insert Name

Matteo Giovanni Maurizio Gianfranco Sara Leonida Juri

```
<input type="text" ng-model="name"/>
<button ng-click="array.push(name)" ng-init="array = [...]">Insert Name</button>
<input type="text" ng-model="query"/>
<h2 ng-repeat="name in array | filter:query">{{name}}</h2>
```

Data-Binding on steroids

Advanced Filtering

Partecipant name:

Partecipant Age:

Insert

Name	Age
Matteo	29
Giovanni	29
Leonida	35
Gianfranco	28

Data-Binding on steroids

```
<!-- Insert -->
<!-- Participant = {name: 'Matteo', age: '29'} -->
<input type="text" ng-model="participant.name"/>
<input type="number" ng-model="participant.age">
<button ng-click="list.push({name: participant.name, age: participant.age})"
ng-init="list = []">Insert</button>

<!-- Filter -->
Name <input type="text" ng-model="query.name">
Age <input type="text" ng-model="query.age">

<!-- Repeat -->
<div ng-repeat="person in list | filter:query">
<span>{{person.name}}</span>
<span>{{person.age}}</span>
```

Exercise

Creare una pagina web in cui un utente possa inserire dei prodotti e filtrarli per nome

I prodotti devono essere salvati in un array `products`

Ognuno dei prodotti deve avere queste caratteristiche: `{category: String, name: String, quantity: Number}`

Altri binding utili: `ng-submit` `ng-options (hard)`

Qui è disponibile un template Html vuoto

goo.gl/DGi6tc

Homeworks!

Install NodeJs

or a webserver

Bower

http://bower.io

BOWER

Package manager

Install dependencies

Grunt | Gulp

http://gruntjs.com

http://gulpjs.com

Task runner

Compile Sass/Scss/Less/...

Build your code

Run tests and lint errors

Yeoman

http://yeoman.io

Scaffholding Tool

Create your base application

Lesson 2

Resume

Data Binging

`ng-model`

`ng-click`

`ng-repeat`

`ng-submit`

Resume

GRUNT

BOWER

YO

It's time to write in Javascript!

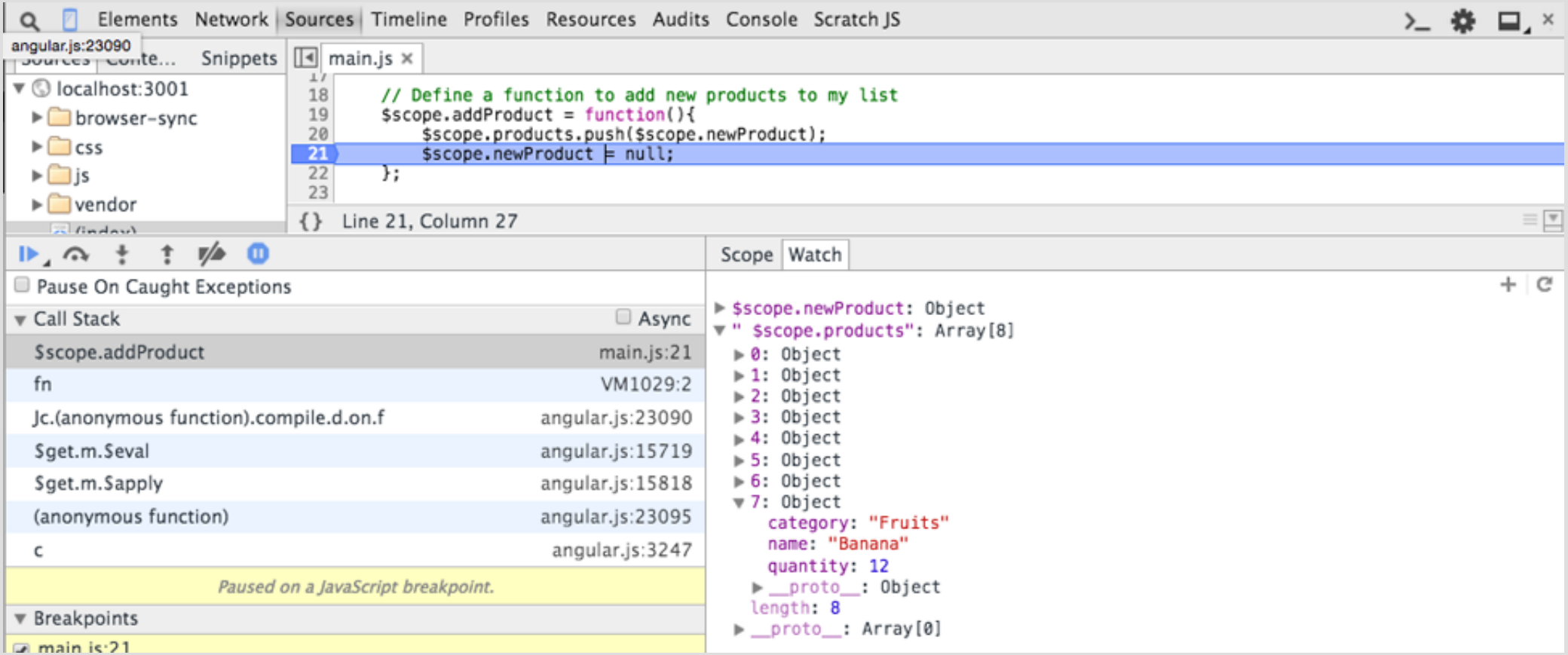
Before Coding, Debug is needed!

Our old friend

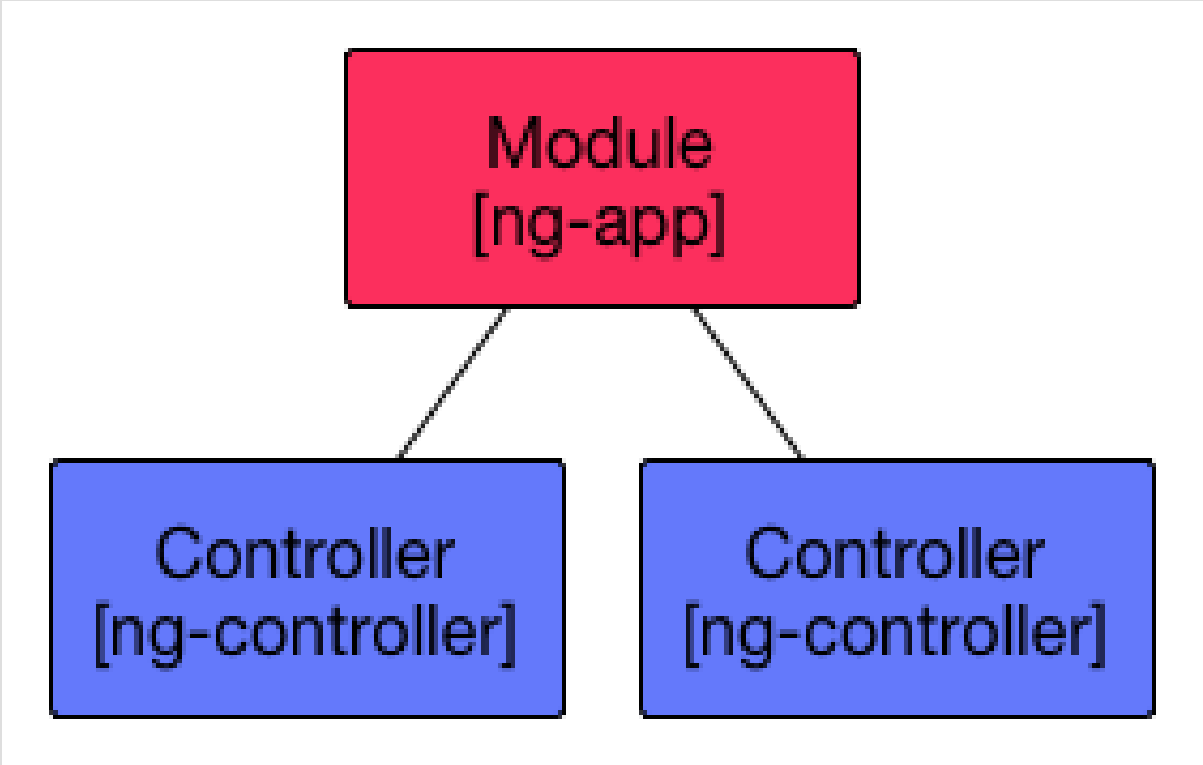
```
console.log(myVar);
```

```
angular.module('groceryStore',[])
.controller('listCtrl', function($scope){
  $scope.myFunction = function(a){
    console.log(a);
    // do stuff;
    return b;
  }
});
```

Chrome Dev Tools



AngularJs Application Structure



AngularJs Application Structure

main.js

```
angular.module('groceryStore',[])
.controller('listCtrl', function($scope){
  // my code
});
```

```
<section ng-controller="listCtrl">
</section>
```

What is a `Controller`*?*

A Js function that react to user input

The place in wich you define your Business Logic

The place in wich set up our data

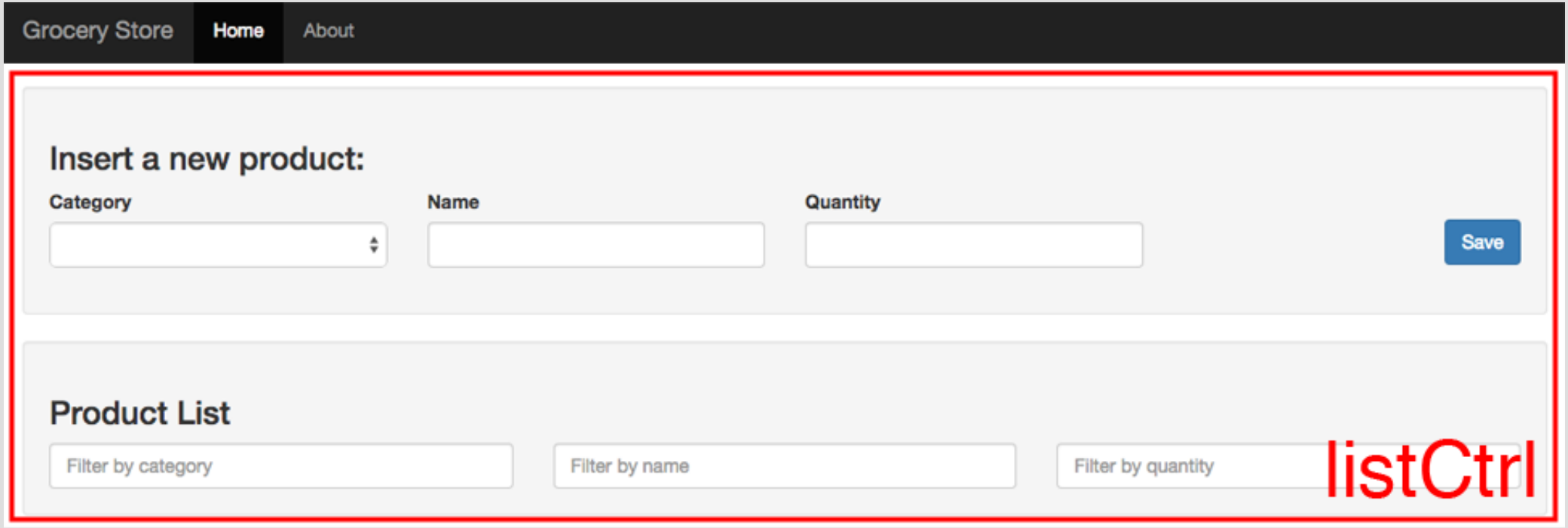
What is `$scope`?

It is an execution context for expressions.

Scopes are arranged in hierarchical structure which mimic the DOM structure of the application.

Scopes are the glue between application controller and the view.

How to bind `controller` to Html



How to bind `controller` to Html

```
<section ng-controller="listCtrl" class="row">
  <article class="col-sm-9">
    // Form and List Html
  </article>
  <article ng-controller="cartCtrl" class="col-sm-3">
    // Cart Html
  </article>
</section>
```

Attach a property to the `$scope`

```
angular.module('groceryStore',[])
.controller('listCtrl', function($scope, $http){

    $scope.categories = ['Fruits', 'Vegetables'];

    $scope.products = [
      {
          category : "Fruits",
          name : "Apple",
          quantity : "12"
      },
      {
          ...
      }
    ];
});
```

Read a property from the `$scope`

```
<div class="well" ng-repeat="product in products | filter:query">
<div class="row">
    <div class="col-sm-3">{{product.category}}</div>
    <div class="col-sm-3">{{product.name}}</div>
    <div class="col-sm-3">{{product.quantity}}</div>
    </div>
</div>
```

Attach methods to the `$scope`

```
angular.module('groceryStore',[])
.controller('listCtrl', function($scope){

    $scope.addProduct = function(){
        $scope.products.push($scope.newProduct);
        $scope.newProduct = null;
    };

    $scope.addToCart = function(product){
        $scope.cart.push(product);
    };
});
```

Call a method from the `$scope`

```
<form ng-submit="addProduct()"> ... </form>
```

```
<div class="well" ng-repeat="product in products | filter:query">
    <div class="row">
        <div class="col-sm-3"></div>
        <div class="col-sm-3"></div>
        <div class="col-sm-3"></div>
        <div class="col-sm-3">
            <a href="" ng-click="addToCart(product)" class="btn btn-primary">
                Buy
            </a>
        </div>
    </div>
</div>
```

Exercise

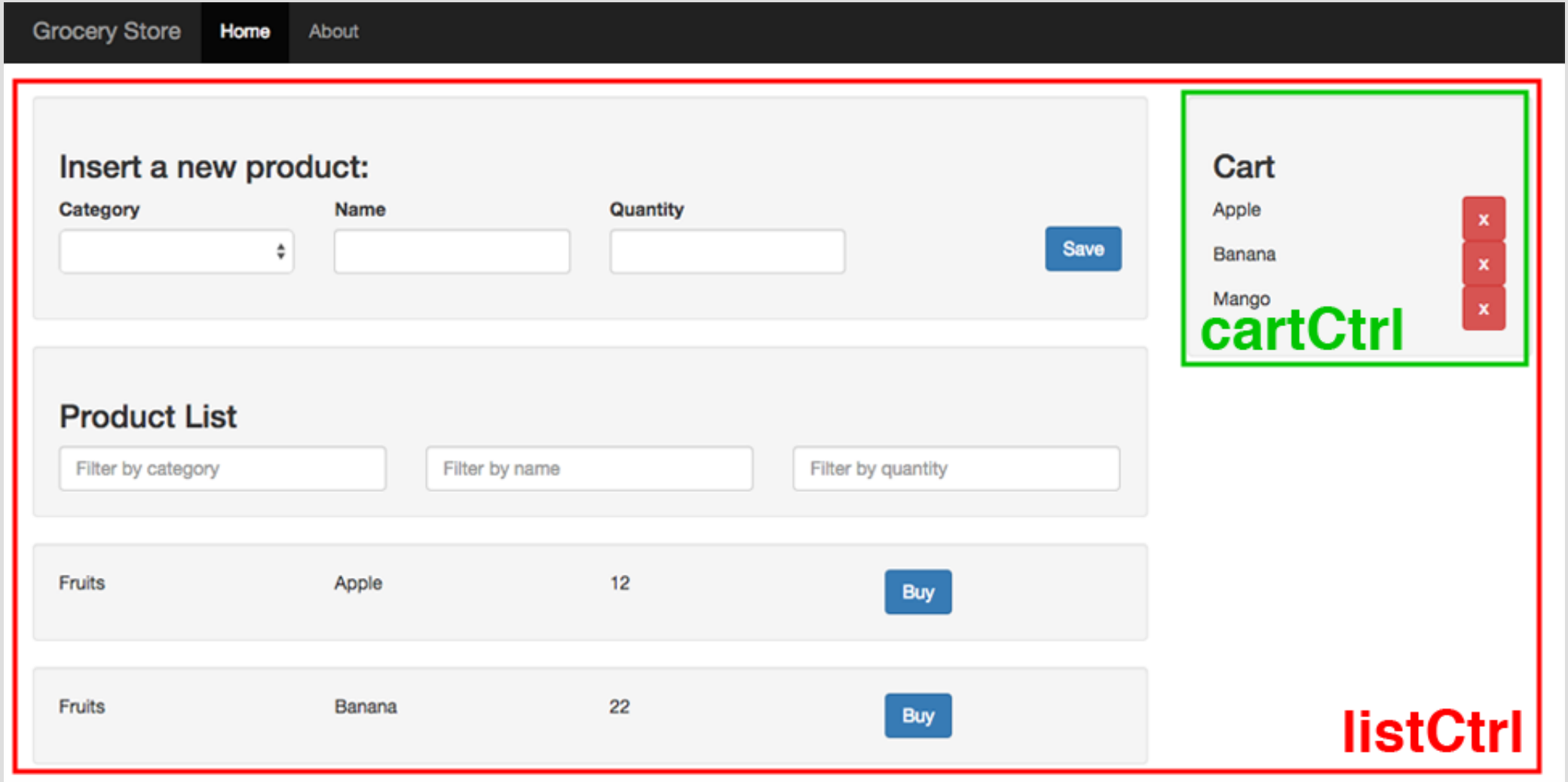
Define a `listCtrl`

Define a method to create a product (`.push`)

Controller Inheritance

Scopes are arranged in hierarchical structure which mimic the DOM structure of the application.

A child controller can inherit from a parent controller


Grocery Store
Home
About
Insert a new product:
Category
Name
Quantity
Save
Cart
Apple
Banana
Mango
cartCtrl
Product List
Filter by category
Filter by name
Filter by quantity
Fruits
Apple
12
Buy
Fruits
Banana
22
Buy
listCtrl

Controller Inheritance

```
angular.module('groceryStore',[])
.controller('parentCtrl', function($scope){

  $scope.parentMethod = function(){/*...*/};

  $scope.parentValue = "Matteo";
})
.controller('childCtrl', function($scope){

  $scope.parentMethod(); //it'll work

  // and I can read parentValue
});
```

Controller Inheritance

```
<div ng-controller="parentCtrl">
    <!-- Some binding -->
    {{parentValue}} <!-- Matteo -->
    <div ng-controller="childCtrl">
        <!-- Some other binding -->
        {{parentValue}} <!-- Matteo -->
        <a ng-click="parentMethod()"> <!-- it'll work -->
    </div>
</div>
```

Pay Attention!

Works only one way!

Pay Attention!

```
angular.module('groceryStore',[])
.controller('parentCtrl',
  function($scope){

    $scope.parentValue = "Matteo";
})
.controller('childCtrl',
  function($scope){

    $scope.parentValue = "Giovanni";
});
```

```
<div ng-controller="parentCtrl">
  <!-- Some binding -->
  {{parentValue}} <!-- Matteo -->
  <div ng-controller="childCtrl">
    <!-- Some other binding -->
    {{parentValue}} <!-- Giovanni -->
  </div>
</div>
```

Exercise

Define a child `cartCtrl`

Define a method to add a product to the cart

`$scope.cart`

Repeat the cart in the

`cartCtrl`

Request data from a Server

Angular play nice with REST API

Respond in JSON

Use Http Statuses

JSON does NOT mean REST

The core `http` service

```
var req = {
  method: 'POST',
  url: 'http://example.com',
  data: { test: 'test' }
};

$http(req)
.success(function(res){...})
.error(function(err){...});
```

Shortcut Methods

```
$http.get('http://example.com')
.success(function(res){...})
.error(function(err){...});
```

```
var data = {firstName: 'Matteo', occupation: 'Frontend Developer'};

$http.post('http://example.com', data)
.success(function(res){...})
.error(function(err){...});
```

Response Methods

`.success`

```
.success(function(res, status, headers, config){
  // executed for 200 and 300 statuses
})
```

`.error`

```
.error(function(res, status, headers, config){
  // executed for 400 and 500 statuses
})
```

`http` *return a promise*

```
var data = {firstName: 'Matteo', occupation: 'Frontend Developer'};

$http.post('http://example.com', data)
.then(function(res){
  // this is the success case
})
.catch(function(err){
  // this is the error case
});
```

My First Request

```
angular.module('groceryStore',[])
.controller('listCtrl', function($scope, $http){

  // Retrieve data form the backend
  $http.get('../mocks/list.json')
  .success(function(list){
      $scope.products = list;
  })
  .error(function(err){
      $scope.error = err.data.message;
  });
});
```

Exercise

Load list data with `$http`
from `mocks/list.json`

REST Resources

`/users [GET]` Query the list of users

`/users/1 [GET]` Get a single user

`/users [POST]` Create a user

`/users/1 [POST]` Update a user

`/users/1 [DELETE]` Delete a user

Angular `$resource`

```
var User = $resource('/user/:userId', {userId:'@id'});
```

Query

`/users [GET]` Query the list of users

```
var users = User.query().then(successHandler, errorHandler);
```

Get

`/users/1 [GET]` Get a single user

```
var user = User.get({userId: 1}).then(successHandler, errorHandler);
```

Create

`/users [POST]` Create a user

```
var newUser = new User({name: 'Matteo'});
newUser.$save().then(successHandler, errorHandler);
```

Update

`/users/1 [POST]` Update a user

```
user.name = 'Giovanni';
user.$save();
```

Delete

`/users/1 [DELETE]` Delete a user

```
user.$remove();
```

Lesson 3

Dependency Injection

Dependency Injection (DI) is
a software design pattern
that deals with how
components get hold of
their dependencies.

Why should I inject?

To Reuse Code!

Remote Call

Shared functionality

Shared data

What can I inject?

`.service`

`.directive`

`.filter`

Where can I inject?

`.controller`

`.service`

`.directive`

`.filter`

`.run`

`.config`

What Services Are?

A service is a function or an object and is used to share data and/or behavior.

How to define a service?

```
angular.module('groceryStore',[])
.service('myService', function(){

  // store some data
  this.myData = 'data';

  // define a method
  this.getData = function(){
    return this.myData;
  };
});
```

How to use a service?

```
angular.module('groceryStore',[])
.controller('myCtrl', function(myService){

  $scope.data = myService.getData();

});
```

Let's see an example!

Remember the `$http` request?

```
$http.get('../mocks/list.json');
.success(function(list){
    $scope.products = list;
})
.error(function(err){
    throw err;
});
```

Let's move it into a `.service`

```
angular.module('groceryStore',[])
.service('listService', function($http){
  this.getList = function(){
      return $http.get('../mocks/list.json');
  }
});
```

I can use this method around my app

If the url change, I have to change it in one place only

Use it in our `.controller`

```
angular.module('groceryStore',[])
.controller('listCtrl', function($scope, listService){

  // Retrieve data form the backend
  listService.getList()
  .success(function(list){
      $scope.products = list;
  })
  .error(function(err){
      throw err; // or better notify the user
  });

});
```

Exercise

Move the `$http` call in a `listService`

Dependency Injection Sintax

```
angular.module('groceryStore',[])
.service('listService', function($http){
  // code
});
```

This can lead to problem while building

Take care and use ngAnnotate

```
angular.module('groceryStore',[])
.service('listService', ['$http', function($http){
  // code
}]);
```

Exercise

Separate `listCtrl` and `cartCtrl`

Create a `cartService` to handle the cart with:

`this.cart` to store the data

`this.add(product)` to add a product

`this.remove(id)` to remove a product

Optional show the number if cart items in the header

Lesson 4

Resume

Dependency Injection

`.service` definition

Injecting a service

Sharing datas and methods

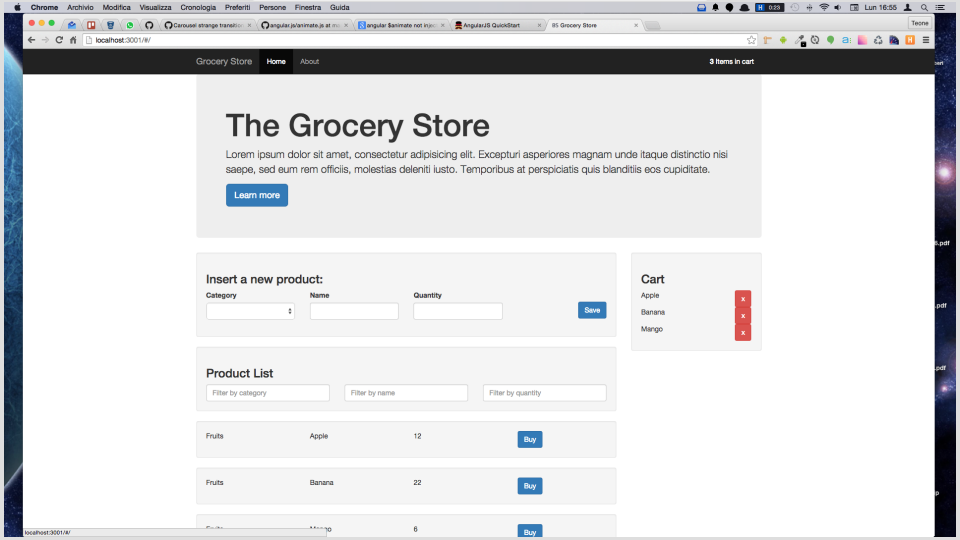
Routes Handling

What a Route is?

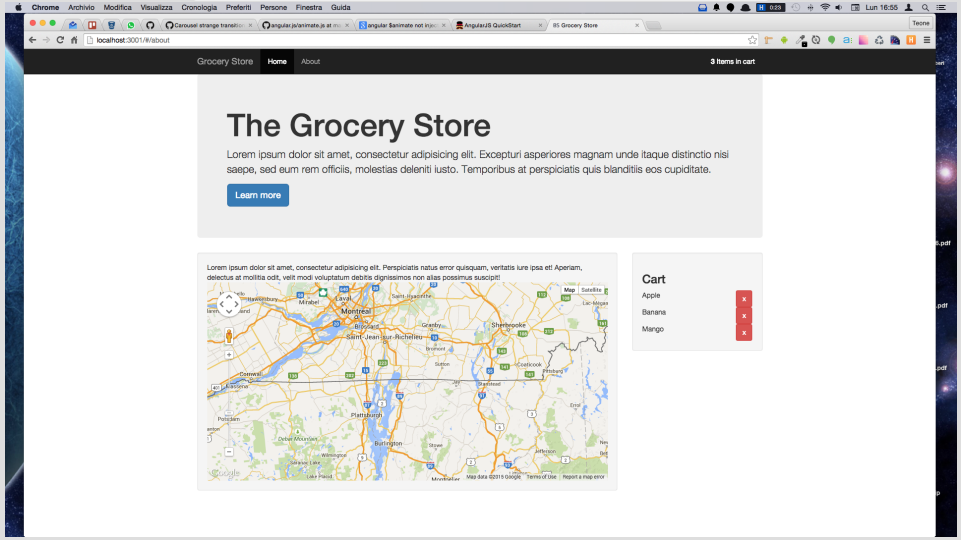
An Url matching the application state

In a more practical way:

`http://localhost:300/#`



`http://localhost:300/#/about`



A route is a page of our application

How do we define Routes?

We need an external Angular module called

`ngRoute`

We should configure some routes

We should define some templates (views)

We should define a position in wich load the template

Import an external module

Module definition

```
angular.module('groceryStore',[])
```

Module definition with dependencies

```
angular.module('groceryStore',['ngRoute'])
```

Script loading order

```
<script src="vendor/angular/angular.min.js"></script>
<script src="vendor/angular-route/angular-route.min.js"></script>
<script src="js/main.js"></script>
```

Define application routes

```
angular.module('groceryStore',['ngRoute'])
.config(['$routeProvider', function($routeProvider) {
    $routeProvider.
    when('/', {
      templateUrl: 'views/list.html',
      controller: 'listCtrl'
    }).
    when('/about', {
      templateUrl: 'views/about.html',
      controller: 'aboutCtrl'
    }).
    otherwise({
      redirectTo: '/'
    });
}]);
```

About routes: Handling Parameter

```
angular.module('groceryStore',['ngRoute'])
.config(['$routeProvider', function($routeProvider) {
    $routeProvider
    .when('/myRoute/:id', {
      // ....
    })
    .otherwise({
      redirectTo: '/'
    });
}])
.controller('myCtrl', function($scope, $routeParams){
  console.log($routeParams.id);
});
```

When visiting `#/myRoute/12` will log `12`

When visiting `#/myRoute` will not match the route

About routes: Optional Parameter

```
angular.module('groceryStore',['ngRoute'])
.config(['$routeProvider', function($routeProvider) {
    $routeProvider
    .when('/myRoute/:name?', {
      // ....
    })
    .otherwise({
      redirectTo: '/'
    });
}])
.controller('myCtrl', function($scope, $routeParams){
  console.log($routeParams.name);
});
```

When visiting `#/myRoute/matteo` will log `matteo`

When visiting `#/myRoute` will log `undefined`

About routes: Query String

```
angular.module('groceryStore',['ngRoute'])
.config(['$routeProvider', function($routeProvider) {
    $routeProvider
    .when('/myRoute', {
      // ....
    })
    .otherwise({
      redirectTo: '/'
    });
}])
.controller('myCtrl', function($scope, $location){
  console.log($location.search());
});
```

When visiting `#/myRoute?name=matteo&age=29` will log `{name: matteo, age: 29}`

When visiting `#/myRoute` will log `undefined`

Create a template

A template is an Html block

such as

```
<input type="text" ng-model="name" placeholder="Insert your name!"/>
<div>Hi !</div>
```

Template can be used for:

Reuse pieces of Html

Render Routes

Reuse pieces with `ng-include`

```
<ng-include src="'path/to/template.html'"></ng-include>
```

Use as route view

An `html` template `ng-view` directive

```
<div ng-view></div>
```

`ng-view` load the template inside the provided container and bind the specified controller

```
when('/about', {
  templateUrl: 'views/about.html',
  controller: 'aboutCtrl'
}).
```

Notes on Route Changes

Route changes does not reload the page

Everytime a route is loaded, the associated controller is executed

Route changes emit events

Exercise

Definire due rotte per la nostra applicazione:
`/` e `/about`

Spostare `listCtrl` in un template

Creare un controller per la pagina `about`

Documentation
https://docs.angularjs.org/api/ngRoute

Do not reinvent the wheel!

Import an open source module

Chrome Archivio Modifica Visualizza Cronologia Preferiti Persone Finestra Guida

Carousel strange transition | angular.js/animate.js at ma | angular $animate not injec | AngularJS QuickStart | BS Grocery Store

localhost:3001/#/about

Grocery Store | Home | About | 3 Items in cart

The Grocery Store

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Excepturi asperiores magnam unde itaque distinctio nisi saepe, sed eum rem officiis, molestias deleniti iusto. Temporibus at perspiciatis quis blanditiis eos cupiditate.

Learn more

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Perspiciatis natus error quisquam, veritatis iure ipsa et! Aperiam, delectus at mollitia odit, velit modi voluptatum debitis dignissimos non alias possimus suscipit!

Cart

Apple x

Banana x

Mango x

angular-google-maps.org

Exercise

Insert a map in the about view

Create a marker that points to Login

Latitude: 45.506639, Logngitude: 9.228062

Practical Advice for the real world

Appiclation Structure

Small to medium apps

```
├── app
|    ├── bower_components
|   ├── images
|   ├── scripts
|   |   ├── controllers
|   |   ├── directives
|   |   ├── services
|   |   └── app.js
|   ├── styles
|   ├── views
|   └── index.html
├── node_modules
└── test
```

github.com/yeoman/generator-angular

YO

GRUNT

Appiclation Structure

Medium to large apps

```
├── bower_components
├── docs
├── e2e
├── gulp
├── node_modules
└── src
    ├── app
    |   ├── modules
    |   |   └── myModule
    |   |       ├── directives
    |   |       ├── service
    |   |       ├── views
    |   |       └── myModule.js
    |   └── main.js
    └── index.html
```

github.com/Swiip/generator-gulp-angular

Automate as much as possible

During development

Watch Files

Live Reload

Compile SASS

Transpile ES6

Lint Code

Automate the Build process

Autoprefix Css

Concat & Minify Css

Concat & Minify Js

Minify Html

Run automatic test

Deploy

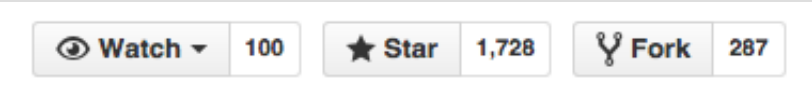
Continue to study

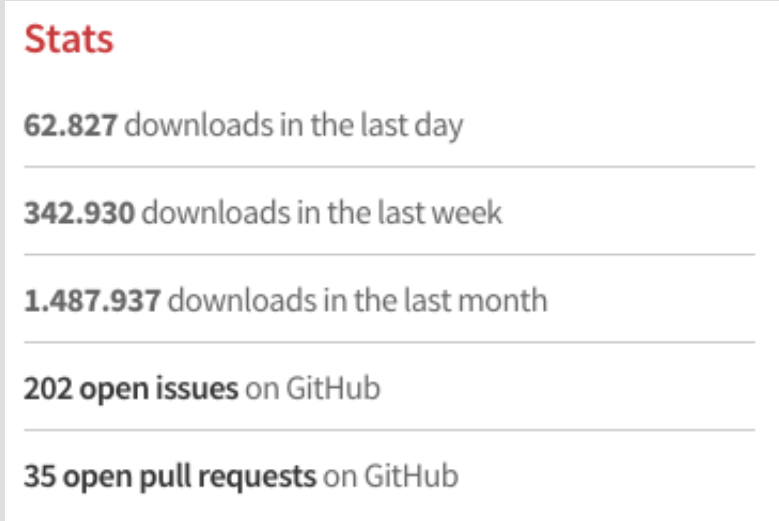
https://scotch.io/

https://egghead.io/

http://code.tutsplus.com/categories/angularjs

Use open source modules





If possible, contribute!

Don't be shy!
Ask Questions!

Thanks

Let's keep in touch:

@_teone

@lambrojos