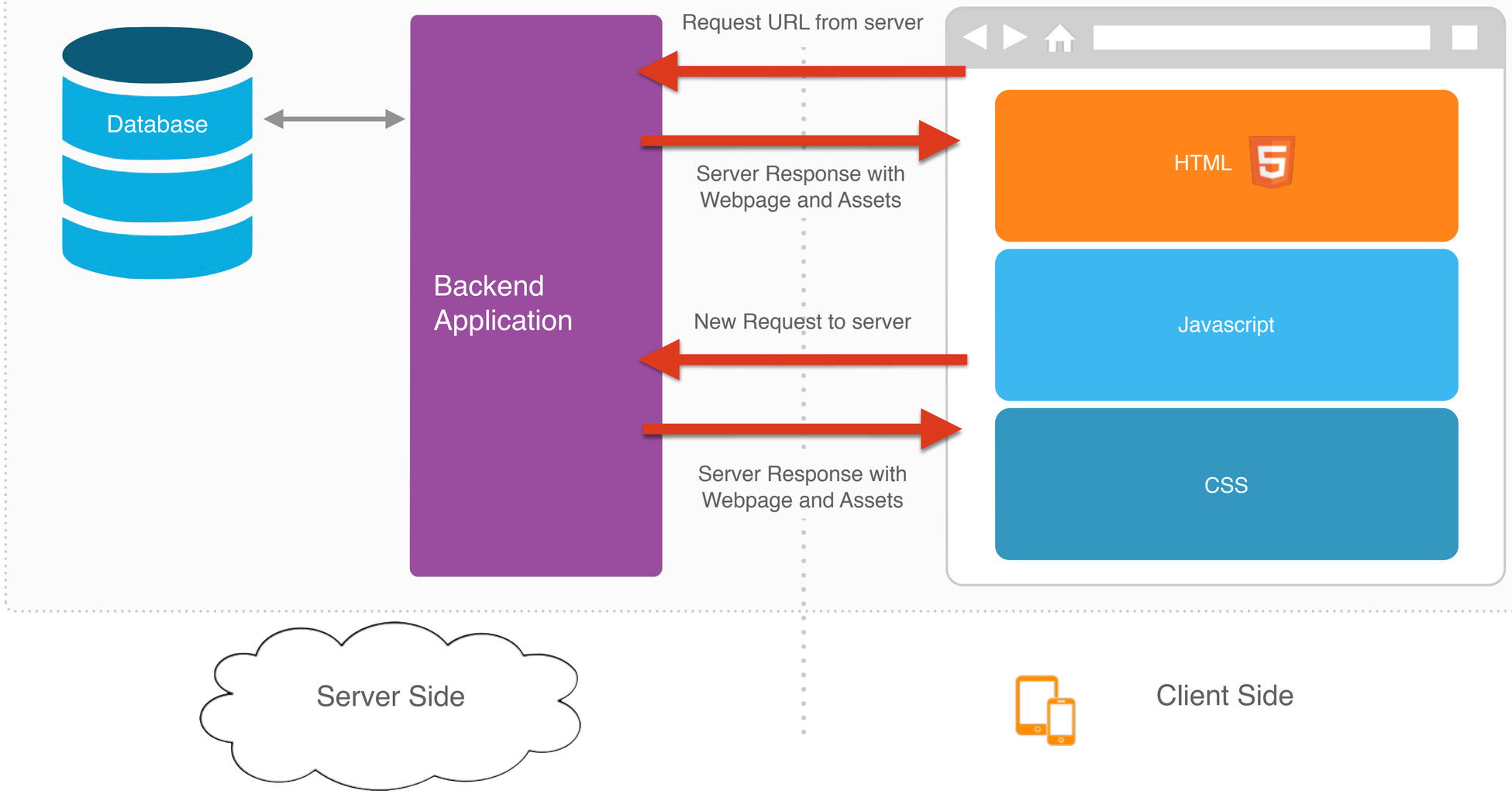
Getting Started with

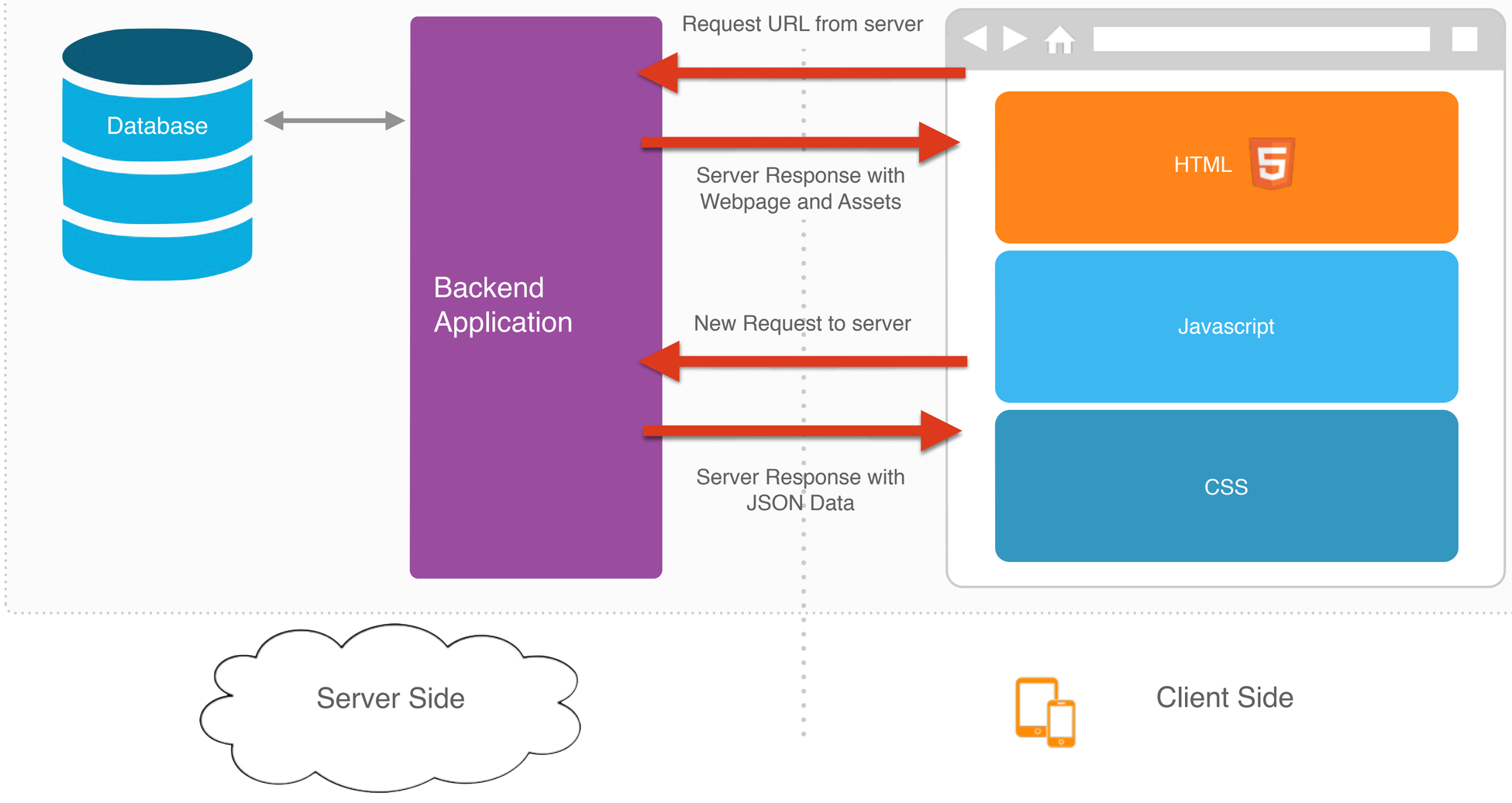
Simona Clapan

simona@leanometry.com

TRADITIONAL PAGE REQUEST

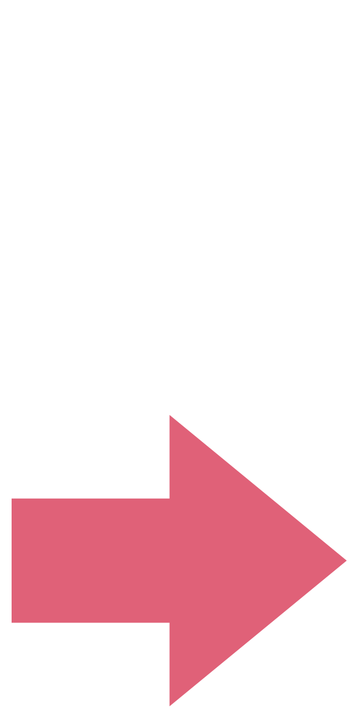


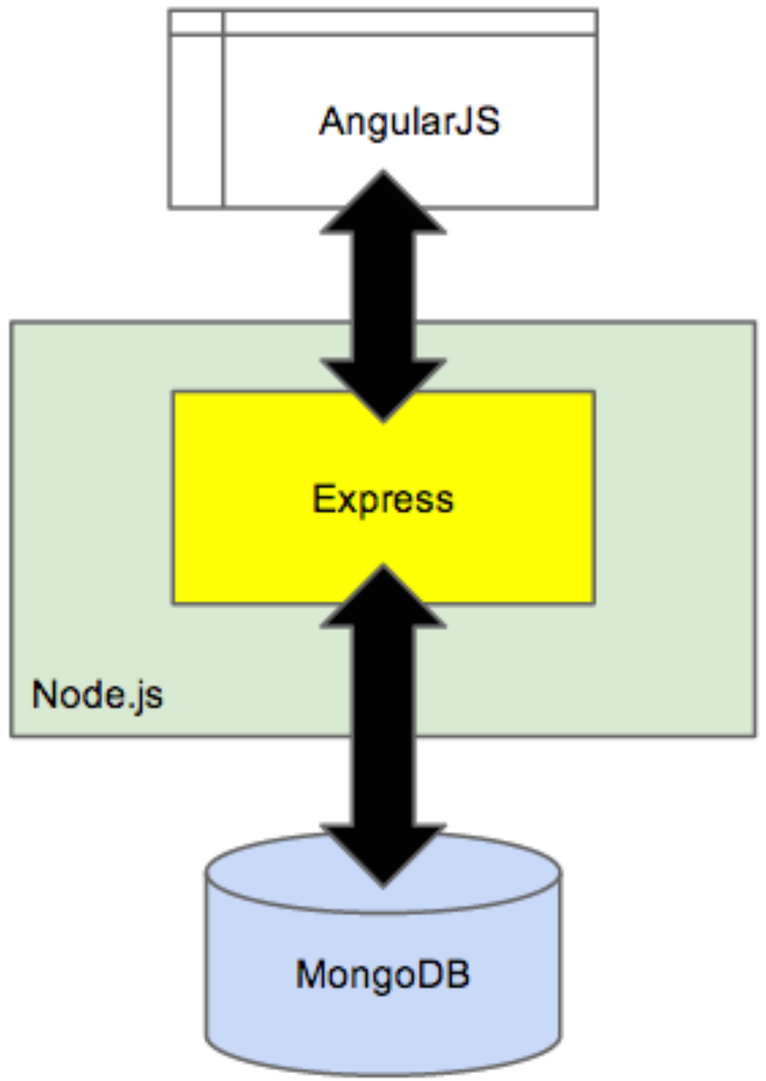
MODERN PAGE REQUEST



TYPICAL APP ARCHITECTURE

M
E
A
N





Client

Server

Database

WHAT IS ANGULARJS?

- Client-side MVC framework: http://angularjs.org
- Problem: Updating page without reload
- Solution: Angular.js declarative, 2-way data binding

```
<html>
 <head></head>
 <body>
        ...
 </body>
</html>
```

index.html

```
function Hello(){
  alert('Hi there!');
}
```

app.js

HOW TO GET STARTED?

- **ANGULAR JS**
  - download from http://angularjs.org
  - angular.min.js
- **TWITTER BOOTSTRAP**
  - download from http://getbootstrap.com
  - bootstrap.min.css

STARTING TEMPLATE

```
<!DOCTYPE html>
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
        <script type="text/javascript" src=“js/angular.min.js"></script>
    </head>
    <body>
    </body>
</html>
```

index.html

DIRECTIVES

- **Directive:** marker on a HTML element (such as an attribute, element name, comment or CSS class) that tell the angular compiler to attach a specified behavior to the HTML element.

```
<html>
 <head></head>
 <body  ng-controller="Hello">
        ...
 </body>
</html>
```

index.html

```
function Hello(){
  alert('Hi there!');
}
```

app.js

MODULES

- **Modules:** Apps are structured in modules that can depend on other modules and can contain controllers, services, directives and filters

app.js

```
// declare a module
var app =
angular.module("myApp", []);
```

module API

module name

dependencies

index.html

```
<!-- reference the new module ->
<html ng-app="myApp"></html>
```

to
bootstrap the module use the directive
ng-app to reference module by name

SAMPLE CODE

```
<!DOCTYPE html>
<!-- reference the new module module -->
<html ng-app="myApp">
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
        <script type="text/javascript" src="js/angular.min.js"></script>
        <!--include our new module-->
        <script type="text/javascript" src="js/app.js"></script>
    </head>
    <body>
    </body>
</html>
```

index.html

Expressions

- **Expressions:** JavaScript-like code snippets that are usually placed in bindings such as {{ expression }}.

```
<p> {{"I have "}} {{4 + 6}} dollars </p>
```

string operation

numeric operation

index.html

```
<p> I have 10 dollars </p>    //after evaluation
```

SAMPLE CODE

```
<!DOCTYPE html>
<html ng-app="myApp">
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />

        <script type="text/javascript" src="js/angular.min.js"></script>
        <script type="text/javascript" src="js/app.js"></script>
    </head>
    <body>
        <p> {{"I have "}} {{4 + 6}} dollars </p>
    </body>
</html>
```

index.html

CONTROLLERS

- **Controllers:** contain the application behavior. Controllers populate the scope with all the necessary data for the view.

- Using proper separation of concerns, controllers should never contain anything related to the DOM.

```
 var app = angular.module('myApp', [ ]);
  app.controller('ContactsController',
function(){

  });
```

app.js

```
var info = {
   firstName: 'John',
   lastName:'Smith',
   email: 'john@leanometry.com',
   phone:'777.922.2321'
};
```

STORE DATA IN CONTROLLER

```
var app = angular.module('myApp', [ ]);
  app.controller('ContactsController', function(){
    this.contact = info;
  });

var info = {
  firstName: 'John',
  lastName:'Smith',
  email: 'john@leanometry.com',
  phone:'777.922.2321'
};
```

app.js

WHERE TO ADD ON HTML

```
<!DOCTYPE html>
<html ng-app="myApp">
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
        <script type="text/javascript" src="js/angular.min.js"></script>
        <script type="text/javascript" src="js/app.js"></script>
    </head>
    <body>

        <h1>Contact Info</h1>
        <div>
            <label>First Name:</label><br/>
            <label>Last Name:</label><br/>
            <label>Email:</label><br/>
            <label>Phone Number:</label>
        </div>

    </body>
</html>
```

index.html

ATTACHING CONTROLLER

```
<!DOCTYPE html>
<html ng-app="myApp">
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
        <script type="text/javascript" src="js/angular.min.js"></script>
        <scr                xt/javas          rc="js/app                script>
    </head>
    <body>
      <h1>Contact Info</h1>
      <div ng-controller="ContactsController as contacts">
        <label>First Name:</label><br/>
        <label>Last Name:</label><br/>
        <label>Email:</label><br/>
        <label>Phone Number:</label>
      </div>
    </body>
</html>
```

directive

controller name

alias

index.html

DISPLAY CONTACT

```
<!DOCTYPE html>
<html ng-app="myApp">
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
        <script type="text/javascript" src="js/angular.min.js"></script>
        <script type="text/javascript" src="js/app.js"></script>
    </head>
    <body>
      <h1>Contact Info</h1>
      <div ng-controller="ContactsController as contacts">
        <label>First Name: {{contacts.contact.firstName}}</label><br/>
        <label>Last Name: {{contacts.contact.lastName}}</label><br/>
        <label>Email: {{contacts.contact.email}}</label><br/>
        <label>Phone Number: {{contacts.contact.phone}}</label>
      </div>
    </body>
</html>
```

index.html

SCOPE

- **Scope:** an object that refers to the application model. It is an execution context for expressions.

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

- Scopes can watch expressions and propagate events.

UNDERSTANDING SCOPE

```
<!DOCTYPE html>
<html ng-app="myApp">
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
        <script type="text/javascript" src="js/angular.min.js"></script>
        <script type="text/javascript" src="js/app.js"></script>
    </head>
    <body>
      <h1>Contact Info</h1>
      <div ng-controller="ContactsController as contacts">
        <label>First Name: {{contacts.contact.firstName}}</label><br/>
        <label>Last Name: {{contacts.contact.lastName}}</label><br/>
        <label>Email: {{contacts.contact.email}}</label><br/>
        <label>Phone Number: {{contacts.contact.phone}}</label>
      </div>
      {{contacts.contact.phone}} <!--will not print a value -->
    </body>
</html>
```

index.html

INSERT NEW CONTACT

```
<body ng-controller="ContactsController as contacts">
    <div >
        <h1>New Contact Info</h1>
        <label>First Name:</label>
        <input type="text" ng-model="contacts.newContact.firstName"/> <br/>

        <label>Last Name:</label>
        <input type="text" ng-model="contacts.newContact.lastName"/><br/>

        <label>Email:</label>
        <input type="text" ng-model="contacts.newContact.email"/><br/>

        <label>Phone Number:</label>
        <input type="text" ng-model="contacts.newContact.phone"/>
    </div>
</body>
```

index.html

ADDING A BUTTON

```
<body ng-controller="ContactsController as contacts">
    <button type="button">Add Contact</button>

    <div >
        <h1>New Contact Info</h1>
        <label>First Name:</label>
        <input type="text" ng-model="contacts.newContact.firstName"/> <br/>

        <label>Last Name:</label>
        <input type="text" ng-model="contacts.newContact.lastName"/><br/>

        <label>Email:</label>
        <input type="text" ng-model="contacts.newContact.email"/><br/>

        <label>Phone Number:</label>
        <input type="text" ng-model="contacts.newContact.phone"/>
    </div>
</body>
```

index.html

ADDING A BUTTON

event name

method name

parameters

```
<body ng-controller="Contacts Controller as contacts">
    <button type="button" ng-click="contacts.addNewContact()">Add Contact</button>

    <div >
        <h1>New Contact Info</h1>
        <label>First Name:</label>
        <input type="text" ng-model="contacts.newContact.firstName"/> <br/>

        <label>Last Name:</label>
        <input type="text" ng-model="contacts.newContact.lastName"/><br/>

        <label>Email:</label>
        <input type="text" ng-model="contacts.newContact.email"/><br/>

        <label>Phone Number:</label>
        <input type="text" ng-model="contacts.newContact.phone"/>
    </div>
</body>
```

directive

index.html

SET VISIBILITY

```
<body ng-controller"ContactsController as contacts”>
    <button type="button" ng-click="contacts.addNewContact()">Add Contact</button>
    <div ng-show="contacts.isAddNewContact" >
        <h1>New Contact Info</h1>
        <label>First Name:</label>
        <input type="text" ng-model="contacts.newContact.firstName"/> <br/>

        <label>Last Name:</label>
        <input type="text" ng-model="contacts.newContact.lastName"/><br/>

        <label>Email:</label>
        <input type="text" ng-model="contacts.newContact.email"/><br/>

        <label>Phone Number:</label>
        <input type="text" ng-model=“contacts.newContact.phone"/>
    </div>
</body>
```

index.html

ADD DATA IN CONTROLLER

```
var app = angular.module('myApp', [ ]);
  app.controller('ContactsController', function(){
    this.contact = info;
    this.newContact ={};
    this.isAddNewContact = false;
    this.addNewContact = function(){
      this.isAddNewContact = true;
    }
  });

var info = {
  firstName: 'John',
  lastName:'Smith',
  email: 'john@leanometry.com',
  phone:'777.922.2321'
});
```

app.js

ADD SAVE BUTTON

```
<body ng-controller="ContactsController as contacts">
    <button type="button" ng-click="contacts.addNewContact()">Add Contact</button>
    <div ng-show="contacts.isAddNewContact" >
        <h1>New Contact Info</h1>
        <label>First Name:</label>
        <input type="text" ng-model="contacts.newContact.firstName"/>
        <label>Last Name:</label>
        <input type="text" ng-model="contacts.newContact.lastName"/>
        <label>Email:</label>
        <input type="text" ng-model="contacts.newContact.email"/>
        <label>Phone Number:</label>
        <input type="text" ng-model="contacts.newContact.phone"/>
        <button type="button" ng-click="contacts.saveContact()">Save</button>
    </div>
</body>
```

index.html

SAVE IN CONTROLLER

```
var app = angular.module('myApp', [ ]);
  app.controller('ContactsController', function(){
    this.contact = info;
    this.newContact ={};
    this.isAddNewContact = false;

  this.addNewContact = function(){
    this.isAddNewContact = true;
  };

  this.saveContact = function(){
    this.contact = this.newContact;
    this.isAddNewContact = false;
  }
});
```

app.js

THANK YOU!

Simona Clapan

simona@leanometry.com

Resources

- Courses -http://campus.codeschool.com/courses/shaping-up-with-angular-js

- AngularJS - http://angularjs.org

FULL HTML

```
<!DOCTYPE html>
<html ng-app="myApp">
    <head>
        <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css" />
        <script type="text/javascript" src="js/angular.min.js"></script>
        <script type="text/javascript" src="js/app.js"></script>
    </head>
    <body ng-controller="ContactsController as contacts">
        <button type="button" ng-click="contacts.addNewContact()">Add Contact</button>
        <div ng-show="contacts.isAddNewContact"  >
       <h1>New Contact Info</h1>
       <label>First Name:</label>
            <input type="text" ng-model="contacts.newContact.firstName"/><br/>
            <label>Last Name:</label>
            <input type="text" ng-model="contacts.newContact.lastName"/><br/>
            <label>Email:</label>
            <input type="text" ng-model="contacts.newContact.email"/><br/>
            <label>Phone Number:</label>
            <input type="text" ng-model="contacts.newContact.phone"/><br/>

            <button type="button" ng-click="contacts.saveContact()">Save Contact</button>

        </div>

        <h1>Contact Info</h1>
        <div >
            <label>First Name: {{contacts.contact.firstName}}</label>
            <label>Last Name: {{contacts.contact.lastName}}</label>
            <label>Email: {{contacts.contact.email}}</label>
            <label>Phone Number: {{contacts.contact.phone}}</label>
        </div>
</body>
</html>
```

index.html

FULL JS

```
(function(){

 var app = angular.module('myApp', [ ]);
 app.controller('ContactsController', function(){
   this.contact = info;
   this.newContact ={};
   this.isAddNewContact = false;
   this.addNewContact = function( ){
   this.isAddNewContact = true;
};

};

this.saveContact = function(){
   this.contact = this.newContact;
   this.isAddNewContact = false;
                }
});

var info = {
   firstName: 'John',
   lastName:'Smith',
   email: 'john@leanometry.com',
   phone:'777.922.2321'
};

})();
```

app.js

Angular Components: 1 of 3

- **Modules:** Apps are structured in modules that can depend on other modules and can contain controllers, services, directives and filters

- **Controllers** contain the application behavior. Controllers populate the scope with all the necessary data for the view. Using proper separation of concerns, controllers should never contain anything related to the DOM.

- **Scope** is used to link the controllers and the views to which they are binded

Angular Components: 2 of 3

- **Directives:** allows you to extend HTML to answer the needs of web applications. Directives let you specify how your page should be structured for the data available in a given scope.

- **Data Binding:** allow defining the binding between the data in the scope and the content of the views.

- **Filters:** allow modifying the way data is displayed.

- **Partial Views:** used specially in single page applications.

Angular Components: 3 of 3

- **Services:** allow reusing code that should be abstracted from controller. Services can be injected in controllers or in other services.

- **Dependency Injection:** retrieves some elements of the application that should be configured when the module will be loaded

- **Events:** $broadcast and $emit