

Getting Mean Practical Perspective

Prof. Dr. Alejandro Zunino Prof. Dr. Alfredo Teyseyre

ISISTAN
Department of Computer Science
UNICEN

2018

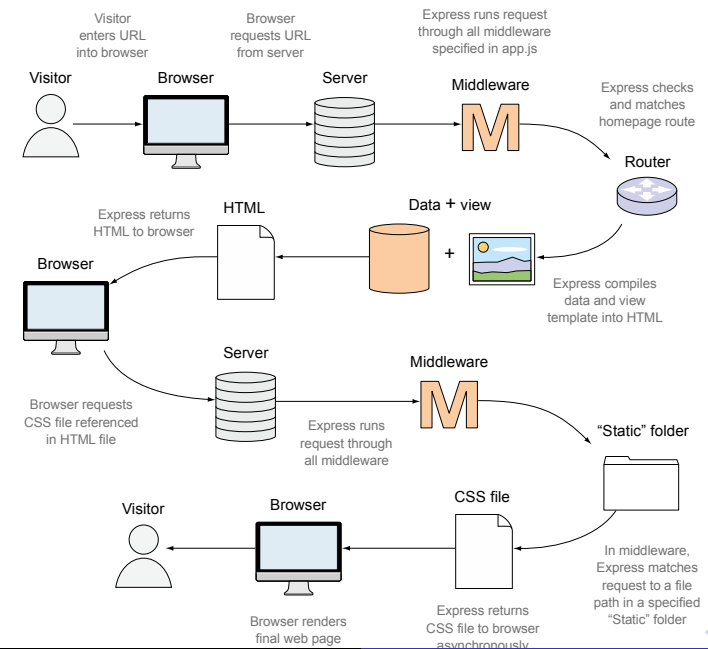
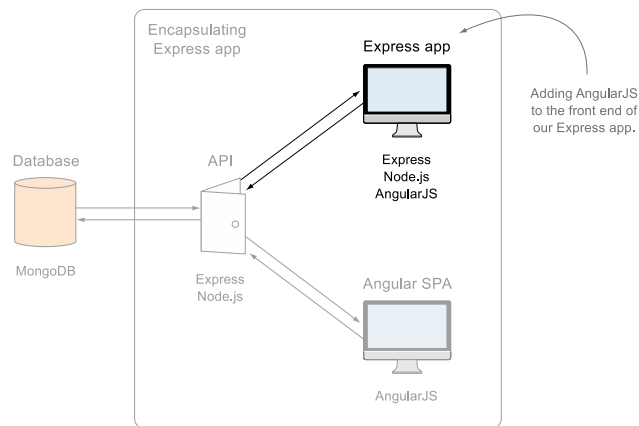
1 Express and AngularJS



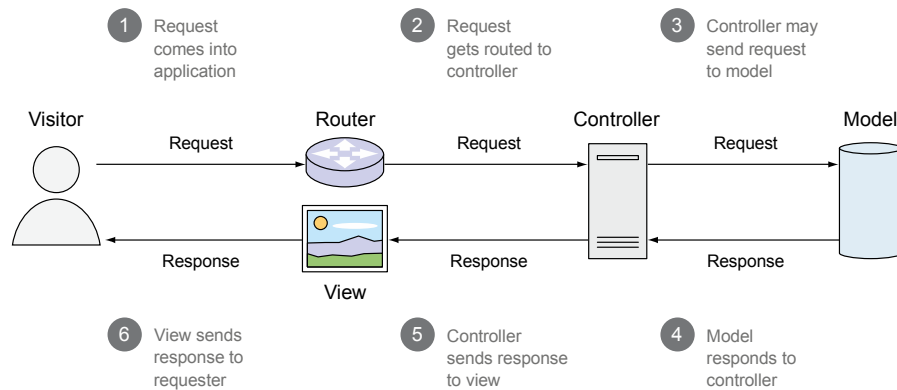
Adding Angular components

Express: The key interactions and processes

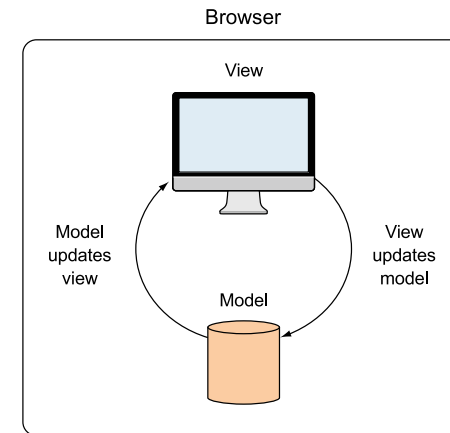
We will use the REST API to access data



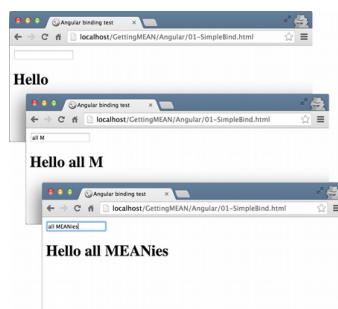
Request-response flow of a basic MVC architecture



In the browser: changes to the view update the model, and changes to the model update the view



Simple Angular binding



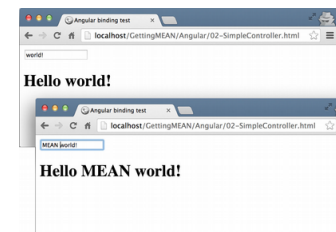
HTML:

```
<html ng-app="myApp">
<head>
  <script src="angular.min.js"></script>
  <meta charset="utf-8">
  <title>Angular binding test</title>
</head>
<body>
  <input ng-model="myInput" />
  <h1>Hello {{ myInput }}</h1>
</body> </html>
```

JavaScript

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

Defining a Controller



HTML:

```
<html ng-app="myApp">
<head ng-controller="myController">
  ...
```

JavaScript:

```
var myController = function($scope) {
  $scope.myInput = "world!";
};
angular
  .module('myApp')
  .controller('myController',
    myController);
```



Products Store

Sort by = ano; reverse = true

Set to unsorted

Name	Desc	Price	Updated		
REGLA	regla 20 cm	\$44.00	13 Jun 2018	Delete	Edit
LAPIZ	lapiz negro	\$24.00	13 Jun 2018	Delete	Edit

HTML:

```
<html ng-app="app">
<head ng-controller="myController">
...
```

JavaScript:

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



HTML:Libraries

```
<!doctype html>
<html ng-app="app">
<head>

<style>
.sortorder:after {
  content: '\25b2'; // BLACK UP-POINTING TRIANGLE }
.sortorder.reverse:after {
content: '\25bc'; // BLACK DOWN-POINTING TRIANGLE }
</style>

<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com
/bootstrap/3.3.7/css/bootstrap.min.css" integrity="
sha384-BVYiISiFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+
PmSTsz/K68vbdEjh4u" crossorigin="anonymous">

<script src="https://ajax.googleapis.com/ajax/libs/angularjs
/1.6.4/angular.min.js"></script>
<script src="app.js"></script>
```



HTML: Sorting

```
<title>Products Store</title> </head>

<body ng-controller="myController">
<div> <h1>Products Store</h1>

<pre>Sort by = {{propertyName}}; reverse = {{reverse}}</pre>

<hr/>
<button ng-click="propertyName = null; reverse = false">
  Set to unsorted</button>
<hr/>

<table class="table">
  <thead>
    <tr>
      <th>
        <button ng-click="sortBy('prod_name')">Name</button>
        <span class="sortorder" ng-show="propertyName === '
prod_name'" ng-class="{reverse: reverse}"></span>
      </th>
      <th>
```



HTML:Sorting

```
<th>
<button ng-click="sortBy('prod_desc')">Desc</button>
<span class="sortorder" ng-show="propertyName === '
prod_desc'" ng-class="{reverse: reverse}"></span>
</th>
<th>
<button ng-click="sortBy('prod_price')">Price</button>
<span class="sortorder" ng-show="propertyName === '
prod_price'" ng-class="{reverse: reverse}"></span>
</th>
<th>
<button ng-click="sortBy('updated_at')">Updated</button>
<span class="sortorder" ng-show="propertyName === '
updated_at'" ng-class="{reverse: reverse}"></span>
</th>
</tr>
</thead>
```



HTML:

Form

```

<tbody>
<tr>
  <td><input class="form-control" ng-model=product.prod_name
  ></td>
  <td><input class="form-control" ng-model=product.prod_desc
  ></td>
  <td><input class="form-control"
    ng-model=product.prod_price></td>
  <td><input class="form-control"
    ng-model=product.updated_at></td>

  <td><button class="btn btn-primary" ng-click=addProduct ()>
    Add Product</button></td>

  <td><button class="btn btn-info" ng-click="update ()">
    Update</button>

  <button class="btn btn-info" ng-click="deselect ()">
    Clean</button></td>
</tr>

```



HTML: Repeat & filters

```

<tr ng-repeat="product in products |
  orderBy:propertyName:reverse">
  <td>{{product.prod_name | uppercase }}</td>
  <td>{{product.prod_desc}}</td>
  <td>{{product.prod_price | currency:"$" }} </td>
  <td>{{product.updated_at | date:"d MMM yyyy"}}</td>

  <td><button class="btn btn-danger" ng-click="remove(
    product._id)">Delete</button> <td>
    <button class="btn btn-warning" ng-click="edit(
    product._id)">Edit</button></td>
</tr>
</tbody>
</table> </div> </body> </html>

```



Javascript:

```

angular.module('app', [])
  .controller('myController', function($scope, $http){

var refresh = function(){
$http.get('/products')
  .then(function(response) {
    console.log("ok", response);
    $scope.products = response.data;
    $scope.product = ({});
  }, function errorCallback(response) {
    console.log("bad!!", response); });
};

refresh();
$scope.propertyName = 'prod_name';
$scope.reverse = true;
$scope.sortBy = function(propertyName) {
  $scope.reverse = ($scope.propertyName === propertyName) ?
    !$scope.reverse : false;
  $scope.propertyName = propertyName;
};

```



Javascript:

```

//Add Product
$scope.addProduct = function (){
  console.log($scope.product);
  $http.post('/products',$scope.product)
    .then(function(response) {
      refresh(); console.log("Ok ADD", response);
    }, function errorCallback(response) {
      console.log("Bad ADD", response);
    });
}

//Remove
$scope.remove = function(id){
  console.log("Remove: ", id);
  $http.delete('/products/' + id)
    .then(function(response) {
      refresh(); console.log("Ok delete", response);
    }, function errorCallback(response) {
      console.log("BAD delete!!", response);
    });
}

```



Javascript:

```
//Edit
$scope.edit = function(id){
  $http.get('/products/' + id)
    .then(function(response) {
      $scope.product = response.data;
      console.log("OK Edit RD: ", response.data);
    }, function errorCallback(response) {
      console.log("Bad edit!!", response); });}

//update
$scope.update = function(){
  $http.put('/products/' + $scope.product._id,
    $scope.product)
    .then(function(response) {
      refresh(); console.log("OK Update", response);
    }, function errorCallback(response) {
      refresh(); console.log("BAD Update!!", response); })
    ; }

//deselect
$scope.deselect = function(){
  $scope.product = ({});
  }.
}
```



Simon Holmes.

Getting MEAN with Mongo, Express, Angular, and Node
Manning 2016.

