

Getting Mean

Practical Perspective

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

ISISTAN
Department of Computer Science
UNICEN

2016

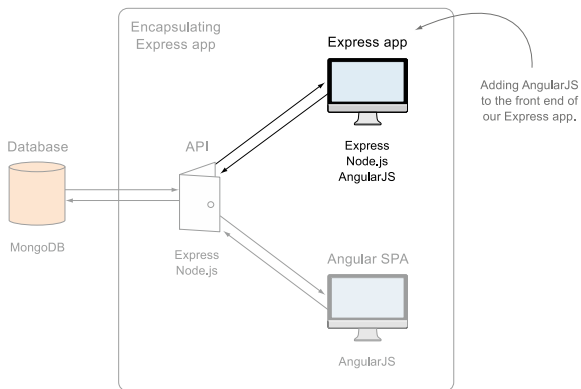
Outline

1 Express and AngularJS

2 Angular SPA

Adding Angular components

We will use the REST API to access data



Jade template engine (Express default)

```
//– index.jade
doctype html
html
  head
    style
      include style.css
  body
    h1 My Site
    p Welcome to my super lame site.
    script
      include script.js

/* style.css */
h1 { color: red; }

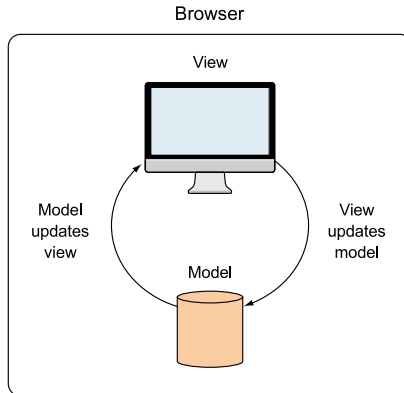
// script.js
console.log('This seem to work');
```

Jade output

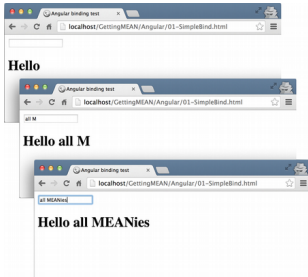
```
<!doctype html>
<html>
  <head>
    <style>
      /* style.css */
      h1 { color: red; }
    </style>
  </head>
  <body>
    <h1>My Site</h1>
    <p>Welcome to my super lame site.</p>
    <script>
      // script.js
      console.log('You are awesome');
    </script>
  </body>
</html>
```

Two way data binding

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



Simple Angular binding



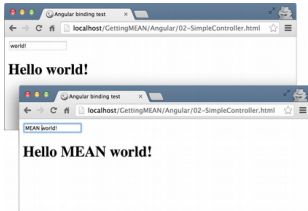
HTML:

```
<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);
```


Adding Angular to an Express application

- 1 Download the Angular library file.
- 2 Create a new JavaScript file for our code:
 - 1 loc8rApp.js: `angular.module('loc8rApp', []);`
- 3 Include these files in the HTML (Jade syntax):
 - 1 `script(src='/angular/angular.min.js')`
 - 2 `script(src='/angular/loc8rApp.js')`
- 4 Define the Angular application in the HTML:
 - 1 `html(ng-app='loc8rApp')`

Using ng-repeat

In JavaScript:

```
var myController = function($scope) {  
  $scope.items = ["one", "two", "three"];  
};
```

In HTML:

```
<body ng-controller="myController">  
  <ul>  
    <li ng-repeat="item_in_items">{{ item }}</li>  
  </ul>  
</body>
```

Using ng-repeat

Resulting HTML:

```
<ul>  
  <li>one</li>  
  <li>two</li>  
  <li>three</li>  
</ul>
```

Using filters

```
<div>{{ 123.2345 | currency }}</div>  
<!-- Output: $123.23 -->
```

```
<div>{{ 123.2345 + 321.321 | currency:"$" }}</div>  
<!-- Output: $444.56 -->
```

```
<div>{{ "Let's_shout" | uppercase }}</div>  
<!-- Output: LET'S SHOUT -->  
<!-- Requires timestamp to be set in scope as date object -->  
{{ timestamp | date:"d_MMM_yyyy" }}  
<!-- Output: 21 Aug 2014 -->
```

Handling Errors (1)

```
request(  
  requestOptions ,  
  function(err , response , body) {  
    var data = body;  
    if (response.statusCode === 200) {  
      data.coords = {  
        lng : body.coords[0] ,  
        lat : body.coords[1]  
      };  
      renderDetailPage(req , res , data);  
    } else {  
      _showError(req, res, response.statusCode);  
    }  
  }  
);
```

Handling Errors (2)

```
var _showError = function (req, res, status) {
  var title, content;
  if (status === 404) {
    title = "404, page not found";
    content = "Oh dear. Looks like we can't find this page. Sorry."
  } else {
    title = status + ", something's gone wrong";
    content = "Something, somewhere, has gone just a little bit wrong."
  }
  res.status(status);
  res.render('generic-text', {
    title : title,
    content : content
  });
}
```

For Further Reading I



Simon Holmes.

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