

Usage Guidelines

Do not forward this document to any non-Infosys mail ID. Forwarding this document to a non-Infosys mail ID may lead to disciplinary action against you, including termination of employment.

Contents of this material cannot be used in any other internal or external document without explicit permission from ETA@infosys.com.

Angular JS

Day 5



Education, Training and Assessment

We enable you to leverage knowledge anytime,
anywhere!

Infosys® | Building
Tomorrow's Enterprise

Copyright Guideline

© 2013-2015 Infosys Limited, Bangalore, India. All Rights Reserved.

Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.

Confidential Information

- This Document is confidential to Infosys Limited. This document contains information and data that Infosys considers confidential and proprietary (“Confidential Information”).
- Confidential Information includes, but is not limited to, the following:
 - ❑ Corporate and Infrastructure information about Infosys
 - ❑ Infosys’ project management and quality processes
 - ❑ Project experiences provided included as illustrative case studies
- Any disclosure of Confidential Information to, or use of it by a third party, will be damaging to Infosys.
- Ownership of all Infosys Confidential Information, no matter in what media it resides, remains with Infosys.
- Confidential information in this document shall not be disclosed, duplicated or used – in whole or in part – for any purpose other than reading without specific written permission of an authorized representative of Infosys.
- This document also contains third party confidential and proprietary information. Such third party information has been included by Infosys after receiving due written permissions and authorizations from the party/ies. Such third party confidential and proprietary information shall not be disclosed, duplicated or used – in whole or in part – for any purpose other than reading without specific written permission of an authorized representative of Infosys.

Course Information

Course Code : LA1211
Course Name : AngularJS
Version Number : 1.2

Day 5 Objectives

- Form Handling in Angular
 - Styling Forms
 - Validation
- Backbone v/s Angular

References

- Brad Green, Shyam Seshadri, AngularJS , O'Reilly Media, 2013.
- <http://docs.angularjs.org/guide/>
- <https://github.com/Angular>

Form handling in Angular



Forms

- AngularJS forms includes a set of controls like input elements, textarea, etc.
- Angular introduces validation services for validating user provided data inside forms.
- Validation service of Angular improves the UI responsiveness and ensures better user experience.
- The user will get instant feedback on data entered.
- Form controls like input, select and textarea provide an area for the user to provide some input to the application.

Forms

- *ngModel* directive of Angular places a very important role in two-way data binding.
- This two-way data binding is ensured by synchronizing the model to the view and the vice-versa.
- *Novalidate* is used in the given example to disable the default validation of the browser.

Styling Forms

- For styling classes, we can use few CSS classes as mentioned below:
 - ng-valid
 - ng-invalid
 - ng-pristine
 - ng-dirty

Styling Forms

- Lets modify the case study 'Shop Your Bags Online' created earlier to include validations.
- Lets initially rely on html5 validations and use the styling classes given by Angular to highlight the errors.
- Lets modify index.html to include the below styles inside the <head> tag.

```
<style type="text/css">

  .css-form input.ng-invalid.ng-dirty {

    background-color: #FA787E;

  }

  .css-form input.ng-valid.ng-dirty {

    background-color: #78FA89;

  }

</style>
```

- Modify the Login template inside index.html as below..

```
<script type="text/ng-template" id="Login">
```

```
<center>
```

```
<div class="LoginFormDiv" align="center">
```

```
<br>
```

```
<form novalidate class="css-form">
```

```
<table>    <tr>
```

```
<td>Username</td>
```

```
<td><input ng-model="username" class="input"></td>
```

```
</tr>    <tr>
```

```
<td>Password</td>
```

```
<td><input ng-model="password" class="input"></td>
</tr>
```

Added a new
field to
capture the
emailid

```
<tr>
```

```
<td>Email ID</td>
```

```
<td><input type="email" ng-model="emailid"
class="input"></td>
```

```
</tr>
```

```
</table>
```

```
<input type="submit" class="button" value="Login"
ng-click="validate()"/>
```

```
</form><br>
```

```
</div> </center>
```

```
</script>
```

Styling Forms

- On viewing index.html page, the below will be shown.

The image displays two screenshots of a web application titled "Shop Your Bags Online". Both screenshots show a login form with three input fields: "Username", "Password", and "Email ID", each followed by a "Login" button. In the top screenshot, the "Email ID" field contains the text "Infy" and has a red background, indicating an invalid email. In the bottom screenshot, the "Email ID" field contains the text "Infy@infosys.com" and has a green background, indicating a valid email. The browser address bar in both screenshots shows "localhost:8090/AngularJS_Demos/Day5/1-StyleClasses_ShopBagsOnline#/main".

Email Id entered is not valid and the respective style for red background is applied

The color changes to green when valid email is entered.

Binding between form and control state

- In Angular, form is an instance of *FormController*.
- To access the form inside the Scope, we just need to use the name of the form.
- Like forms, control is also an instance of *NgModelController*, in Angular.
 - Control instance will be available in the form using the name attribute.
- Since we can use the control options, we can update the previous.

Built-in Validations

16

- Let's modify the case study 'Shop Your Bags Online' created in the previous slide by adding built-in validations. Let's move the code inside Login template page, Login.html inside the partials folder and add the below

```
<center><div class="LoginForm" style="border: 1px solid #ccc; padding: 10px; width: 40%; margin: auto; text-align: center;" align="center"><br>
<form novalidate class="css-form" name="form">
<table>
<tr>
<td><input ng-model="username" class="input" type="text" name="username" required />
<div ng-show="form.username.$error.pattern" style="color: red; font-size: 0.9em; margin-top: 5px;">Plz enter proper username</div>
</td>
<td>Password</td>
</tr>
<tr>
<td><input ng-model="password" class="input" type="password" name="password" required />
</td>
<td></td>
</tr>
<tr>
<td><input type="email" ng-model="emailid" class="input" name="emailid" required />
<div ng-show="form.emailid.$error.required" style="color: red; font-size: 0.9em; margin-top: 5px;">This span will be shown only if email is not entered</div>
<div ng-show="form.emailid.$error.pattern" style="color: red; font-size: 0.9em; margin-top: 5px;">This span will be shown only if email pattern is invalid</div>
</td>
<td></td>
</tr>
<tr>
<td colspan="2"><input type="submit" class="button" value="Login" ng-click="validate()" />
</td>
</tr>
</table>
</div></center>
```

Disabling html5 validations

Name and css class of the form

Adding required validation

Adding pattern validation to username

This div will be shown only if email is not entered

This span will be shown only if email pattern is invalid

This div will be shown only if username is invalid

Built-in Validations



localhost:8090/AngularJS_Demos/Day5/2-StyleClasses_ShopBagsOnline#/main

Shop Your Bags Online

Username

Password

Email ID

Plz enter proper username

Login



localhost:8090/AngularJS_Demos/Day5/2-StyleClasses_ShopBagsOnline#/main

Shop Your Bags Online

Username

Password

Email ID

Plz enter email

Login



localhost:8090/AngularJS_Demos/Day5/2-StyleClasses_ShopBagsOnline#/main

Shop Your Bags Online

Username

Password

Email ID

Login

Custom Validation in Angular

| 18

- Validation may occur in two places:
 - When the model is updated, the view should be aligned accordingly.
 - When the user of the application, interacts with the control and updates the value, the model should be aligned accordingly.
 - Add the below code to controllers.js file.

```
bagControllers.directive('ngUnique', ['$http', function
(async) {

    return {
        require: 'ngModel',
        link: function (scope, elem, attrs, ctrl) {
            elem.bind('blur', function (evt) {
                scope.$apply(function () {
                    var val = elem.val();

                    if(val == 'admin@infy.com'){
                        ctrl.$setValidity('unique', false);
                    }
                    else{
                        ctrl.$setValidity('unique', true);
                    }
                });
            });
        }
    }
}]);
```

Custom Validation

- Modify the emailID textbox as below in Login.html.

```
<input type="email" ng-model="emailid" class="input" name="emailid" required ng-unique>
```

Added custom validation for checking

- The below screen is rendered.

Shop Your Bags Online

Username

Password

Email ID

emailid already exists.

Login

Error message is displayed on entering inappropriate emailed.

Backbone v/s Angular



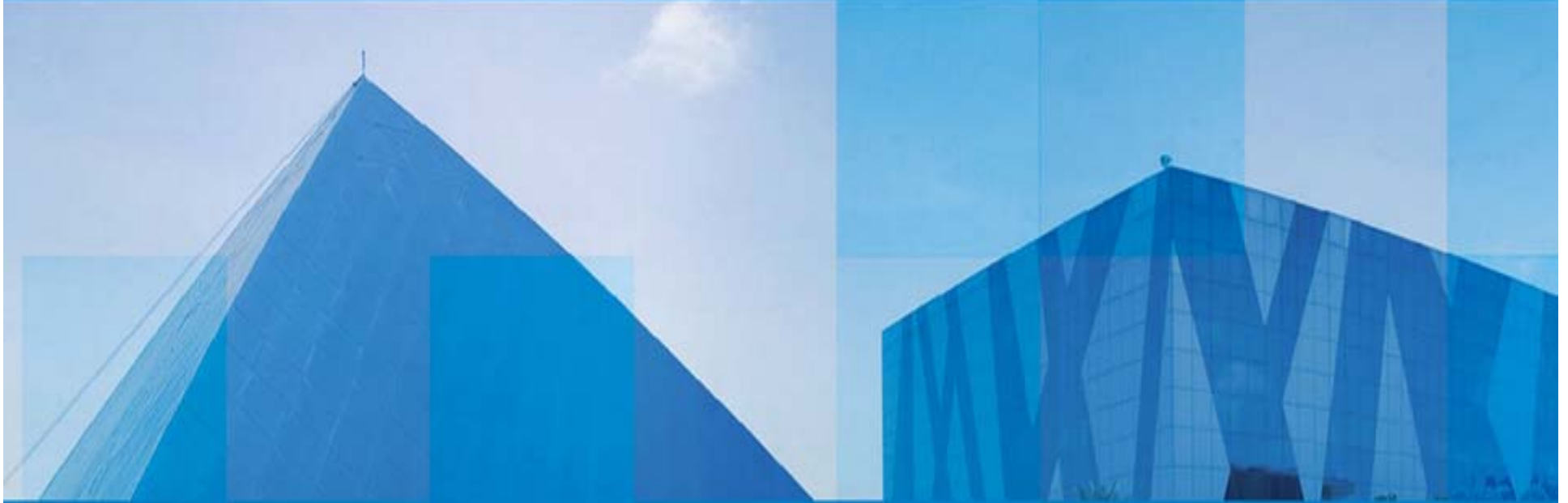
Backbone v/s Angular

- Backbone.js is a very light-weight library. This eventually means that we need to type more, as a developer. Backbone can hence be used with a broader spectrum of applications.
- Angular.js provides much more abstraction to the user and has a very specific structure in which apps should be created. This means it is not necessary that the framework fits into every requirement
- Backbone just splits your concerns into views and models.

Backbone v/s Angular

- AngularJS doesn't force the developer to write the event handling and triggering code. This is automatically handled in Angular. This can also be considered as a disadvantage when you do not want to update the view on a event trigger.
- To use Backbone in our app, we need to extend Backbone's classes.
- While using AngularJS, we need not extend any special classes. The code written by the developer is more like writing HTML and JS.

Thank You



© 2013 Infosys Limited, Bangalore, India. All Rights Reserved. Infosys believes the information in this document is accurate as of its publication date; such information is subject to change without notice. Infosys acknowledges the proprietary rights of other companies to the trademarks, product names and such other intellectual property rights mentioned in this document. Except as expressly permitted, neither this documentation nor any part of it may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, printing, photocopying, recording or otherwise, without the prior permission of Infosys Limited and/ or any named intellectual property rights holders under this document.

Infosys[®] | Building
Tomorrow's Enterprise