Chapter 7  
***Listing 7-1.*** The $window Service

<!DOCTYPE html >

<html ng-app="myapp">

<head>

<title>Angular Services</title>

<script src="js/angular.min.js"></script>

<script>

var module = angular.module('myapp', []);

module.controller("MyController", function ($scope, **$window**) {

$scope.winWidth = **$window.innerWidth**;

});

</script>

</head>

<body ng-controller="MyController">

<p>Window width: {{winWidth}}px</p>

</body>

</html>

***Listing 7-2.*** Using the $location Service

<!DOCTYPE html >

<html ng-app="myapp">

<head>

<title>Angular Services</title>

<script src="js/angular.min.js"></script>

<script>

var module = angular.module('myapp', []);

module.controller("MyController", function ($scope, $location) {

$scope.url = $location.absUrl();

$scope.protocol = $location.protocol();

$scope.host = $location.host();

$scope.port = $location.port();

});

</script>

</head>

<body ng-controller="MyController">

<p>The URL is: {{url}}</p>

<ul>

<li>{{protocol}}</li>

<li>{{host}}</li>

<li>{{port}}</li>

</ul>

</body>

</html>

***Listing 7-3.*** Using the $document Service to Access the Page Title

<!DOCTYPE html >

<html ng-app="myapp">

<head>

<title>Angular Services</title>

<script src="js/angular.min.js"></script>

<script>

var module = angular.module('myapp', []);

module.controller("MyController", function ($scope, **$document**) {

$scope.docTitle = **$document[0].title**;

});

</script>

</head>

<body ng-controller="MyController">

<p>The page title is: {{docTitle}}</p>

</body>

</html>

***Listing 7-4.*** A Basic Angular Service

<!DOCTYPE html >

<html ng-app="myapp">

<head>

<title>Angular Services</title>

<script src="js/angular.min.js"></script>

<script>

var module = angular.module('myapp', []);

module.factory(**'dateTimeService'**, function () {

var dateTimeSvc = {};

dateTimeSvc.getDate = function () { return new Date().toDateString();

}

dateTimeSvc.getTime = function () { return new Date().toTimeString();

}

return dateTimeSvc;

}).controller("MyController", function ($scope, **dateTimeService**) {

$scope.theDate = dateTimeService.getDate();

$scope.theTime = dateTimeService.getTime();

});

</script>

</head>

<body ng-controller="MyController">

<p>{{theDate}}</p>

<p>{{theTime}}</p>

</body>

</html>

***Listing 7-5.*** A First Look at the $http Service var promise = $http({method: 'POST', url: 'memberservices/register', data: theData});

promise.success(function (data, status, headers, config, statusText) {

// this callback will be called asynchronously

// when the response is available

});

promise.error(function (data, status, headers, config, statusText) { // called asynchronously if an error occurs // or server returns response with an error status.

});

***Listing 7-6.*** The memberDataStoreService Service var module = angular.module('myapp', []);

module.factory('memberDataStoreService', function ($http) {

var memberDataStore = {};

memberDataStore.doRegistration = function (theData) {

var promise = $http({method: 'POST', url: 'memberservices/register', data: theData}); return promise;

}

return memberDataStore;

});

***Listing 7-7.*** Making Use of Our Custom Service

<!DOCTYPE html>

<html ng-app="myapp">

<head lang="en">

<meta charset="UTF-8">

<title>Registration Form</title>

<script src="js/angular.min.js"></script>

<script>

var module = angular.module('myapp', []);

module.factory('memberDataStoreService', function ($http) {

var memberDataStore = {};

memberDataStore.doRegistration = function (theData) {

var promise = $http({method: 'POST', url: 'memberservices/register', data: theData}); return promise;

}

return memberDataStore;

}).controller("MyController", function ($scope, **memberDataStoreService**) {

$scope.person = {};

$scope.person.newsletterOptIn = true;

$scope.person.channels = [

{ value: "television", label: "Television" },

{ value: "radio", label: "Radio" },

{ value: "social-media", label: "Social Media"},

{ value: "other", label: "Other"}

];

$scope.register = function () {

$scope.firstNameInvalid = false;

$scope.lastNameInvalid = false;

$scope.emailInvalid = false;

$scope.researchInvalid = false;

$scope.showSuccessMessage = false;

$scope.showErrorMessage = false;

if (!$scope.registrationForm.firstName.$valid) {

$scope.firstNameInvalid = true;

}

if (!$scope.registrationForm.lastName.$valid) {

$scope.lastNameInvalid = true;

}

if (!$scope.registrationForm.email.$valid) {

$scope.emailInvalid = true;

}

if (!$scope.registrationForm.research.$valid) {

$scope.researchInvalid = true;

}

// If the registration form is valid, use the

// memberDataStoreService to submit the form data **if ($scope.registrationForm.$valid) {**

**var promise = memberDataStoreService.doRegistration($scope.person);**

**promise.success(function (data, status) {**

**$scope.showSuccessMessage = true;**

**});**

**promise.error(function (data, status) {**

**$scope.showErrorMessage = true;**

**});**

**$scope.doShow = true;**

**}**

}

})

</script>

<style>

body, input, select {

font: normal 16px/1.4 Georgia;

}

input:not([type='checkbox']), select {

width: 250px;

}

input, select { padding: 5px; margin-top: 12px;

}

input[name='email'].ng-dirty.ng-invalid-email {

color: red;

}

input[name='email'].ng-dirty.ng-valid-email {

color: green;

}

form span, .error { color: red;

}

</style>

</head>

<body>

<div>

<div ng-controller="MyController">

<form name="registrationForm" ng-submit="register()" novalidate>

**<div ng-show="showSuccessMessage">**  **Thank you for taking the time to register!**

**</div>**

**<div class="error" ng-show="showErrorMessage">**

**There appears to have been a problem with your registration.<br/>**

**</div>**

<input type="text" placeholder="First Name" name="firstName" ng-model="person.firstName" required/>

<span ng-show="firstNameInvalid"><br/>Please enter a value for First name</span>

<br/>

<input type="text" placeholder="Last Name" name="lastName" ng-model="person.lastName" required/>

<span ng-show="lastNameInvalid"><br/>Please enter a value for Last name</span>

<br/>

<input type="email" placeholder="Email" name="email" ng-model="person.email" required/>

<span ng-show="emailInvalid"><br/>A valid email address is required</span>

<br/>

<select name="research" ng-model="person.levels"

ng-options="obj.label as obj.value for obj in person.channels" required>

<option value="">Where did you hear about us?</option>

</select>

<span ng-show="researchInvalid"><br/>Please tell us where you heard about us</span>

<br/>

<input ng-model="person.newsletterOptIn" type="checkbox" name="newsletterOptIn"

id="newsletterOptIn" value="newsletterOptIn"/>

<label for="newsletterOptIn">Recieve monthly newsletter</label>

<br/>

<input type="submit" value="Register"/>

</form>

</div>

</div>

</body>

</html>

***Listing 7-8.*** Using the memberDataStoreService

// If the registration form is valid, use the

// memberDataStoreService to submit the form data

if ($scope.registrationForm.$valid) {

var promise = memberDataStoreService.doRegistration($scope.person);

promise.success(function (data, status) {

$scope.showSuccessMessage = true;

});

promise.error(function (data, status) {

$scope.showErrorMessage = true;

});

$scope.doShow = true;

}

***Listing 7-9.*** Honing In on the Success and Error Messages

<div ng-show="showSuccessMessage"> Thank you for taking the time to register!

</div>

<div class="error" ng-show="showErrorMessage">

There appears to have been a problem with your registration.<br/>

</div>

***Listing 7-10.*** Adding a Loading Animation and Disabling the Register Button

<input **ng-disabled="working"** type="submit" value="Register"/>

<span **ng-show="working"** style="padding-left:10px;">

<img src="images/loading.gif"/>

</span>

***Listing 7-11.*** Indicating That Work Is in Progress

// If the registration form is valid, use the

// memberDataStoreService to submit the form data

if ($scope.registrationForm.$valid) {

**$scope.working = true;**  var promise = memberDataStoreService.doRegistration($scope.person);

promise.success(function (data, status) {

$scope.showSuccessMessage = true;

});

promise.error(function (data, status) {

$scope.showErrorMessage = true;

});

**promise.finally(function () {**  **$scope.working = false;**

**});**

$scope.doShow = true;

}

***Listing 7-12.*** Handling Request Data

promise.success(function (**data**, status) {

**$scope.successMessage = "Your transaction identifier is " + data.transactionID;**

$scope.showSuccessMessage = true;

});

***Listing 7-13.*** Handling Errors promise.error(function (data, **status**) {

if (**status** === 0) {

$scope.errorMessage = "network or http level issue";

} else {

$scope.errorMessage = "response HTTP status is " + **status**;

}

$scope.showErrorMessage = true;

});