Chapter 4

***Listing 4-1.*** Raw Sample Data

<script>

function MyFilterDemoCtrl($scope) {

var someData = { firstName: 'JENNA', surname: 'GRANT',

dateJoined: new Date(2010, 2, 23),

consumption: 123.659855, plan: 'super-basic-plan'

};

$scope.data = someData;

}|  
</script>

***Listing 4-2.*** Angular Filter Example

<!DOCTYPE html>

<html>

<head>

<title>Listing 4-2</title>

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

<script>

function MyFilterDemoCtrl($scope) {

var someData = { firstName: 'JENNA', surname: 'GRANT',

dateJoined: new Date(2010, 2, 23),

consumption: 123.659855, plan: 'super-basic-plan'

};

**$scope.data = someData;**

}

</script>

</head>

<body ng-app ng-controller="MyFilterDemoCtrl">

<p>

<!-- Unfiltered data -->

<strong>First Name</strong>: {{data.firstName}}<br/>

<strong>Surname:</strong> {{data.surname}}

</p>

<p>

<!-- Filtered data -->

<strong>First Name</strong>: {{data.firstName **| lowercase**}}<br/>

<strong>Surname:</strong> {{data.surname **| lowercase** }}

</p>

</body>

</html>

***Listing 4-3.*** Achieving Same Result Without Filter

<script>

function MyFilterDemoCtrl($scope) {

var someData = { firstName: 'JENNA', surname: 'GRANT',

dateJoined: new Date(2010, 2, 23),

consumption: 123.659855, plan: 'super-basic-plan'

};

**// do the lowercaing here instead of using a filter**  someData.firstName = someData.firstName.**toLowerCase();**  someData.surname = someData.surname.**toLowerCase();**

$scope.data = someData;

}

</script>

***Listing 4-4.*** Rounding Up Values with the Number Filter

<!DOCTYPE html>

<html>

<head>

<title>Listing 4-4</title>

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

<script>

function MyFilterDemoCtrl($scope) {

var someData = { firstName: 'JENNA', surname: 'GRANT',

dateJoined: new Date(2010, 2, 23), **consumption: 123.659855**, plan: 'super-basic-plan'

};

**$scope.data = someData;**

}

</script>

</head>

<body ng-controller="MyFilterDemoCtrl">

<p>

Consumption: {{data.consumption }}<br/>

Consumption: {{data.consumption **| number** }}

</p>

</body>

</html>

***Listing 4-5.*** The Date Filter in Action

<!DOCTYPE html>

<html>

<head>

<title>Listing 4-5</title>

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

<script>

function MyFilterDemoCtrl($scope) {

var someData = { firstName: 'JENNA', surname: 'GRANT',

dateJoined: new Date(2010, 2, 23),

consumption: 123.659855, plan: 'super-basic-plan'

};

$scope.data = someData;

}

</script>

</head>

<body ng-app ng-controller="MyFilterDemoCtrl">

<p>medium:<strong> {{ data.dateJoined | date:'medium'}} </strong></p>

<p>mediumDate:<strong> {{ data.dateJoined | date:'mediumDate'}} </strong></p>

<p>shortDate:<strong> {{ data.dateJoined | date:'shortDate'}} </strong></p>

<p>This customer joined in the month of {{ data.dateJoined | date:'MMMM'}} on a {{

data.dateJoined | date:'EEEE'}} at {{ data.dateJoined | date:'ha'}}</p>

</body>

</html>

***Listing 4-6.*** Adding Historical Data to the Data Source

<script>

function MyFilterDemoCtrl($scope) {

var someData = { firstName: 'JENNA', surname: 'GRANT',

dateJoined: new Date(2010, 2, 23),

consumption: 123.659855, plan: 'super-basic-plan',

// Last 12 months of data usage monthlyUsageHistory:

[123.659855,

89.645222,

97.235644, 129.555555,

188.699855,

65.652545, 123.659855,

89.645222,

97.235644, 129.555555,

188.699855,

65.652545] };

$scope.data = someData;

}

</script>

We have 12 months of data in the monthlyUsageHistory array, but this could just as easily be, say, five years’ worth of data. Listing 4-7 uses the limitTo filter to display a summary view of the data (only the last three values).

***Listing 4-7.*** Displaying a Subset of the monthlyUsageHistory Data

<body ng-app ng-controller="MyFilterDemoCtrl">

<h2>Gigabytes used over the last 3 months</h2>

<ul>

<li ng-repeat="gigabytes in data.monthlyUsageHistory **| limitTo:5**">

{{ gigabytes | number:2}}

</li>

</ul>

</body>

***Listing 4-8.*** myAppModule.js

// create a new module called 'myAppModule' and save

// a reference to it in a variable called myAppModule var myAppModule = angular.module('**myAppModule**', []);

// use the myAppModule variable to

// configure the module with a controller

myAppModule.controller('MyFilterDemoCtrl', function ($scope) {

// controller code would go here

}

);

// use the myAppModule variable to

// configure the module with a filter myAppModule.filter('stripDashes', function() {

return function(txt) {

// filter code would go here

};

});

***Listing 4-9.*** Referring to the Module by Name

// Create a new module

angular.module('**myAppModule**', []);

// configure the module with a controller

angular.module('**myAppModule**').controller('MyFilterDemoCtrl', function ($scope) {

// controller code would go here

}

);

// configure the module with a filter

angular.module('**myAppModule**'). filter('stripDashes', function() {

return function(txt) {

// filter code would go here

};

});

***Listing 4-10.*** An index.html File Set Up to Use myAppModule

<!DOCTYPE html >

<html **ng-app="myAppModule"**>

<head lang="en">

<meta charset="UTF-8">

<title>Listing 4-10</title>

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

</head>

<body ng-controller="MyFilterDemoCtrl">

</body>

</html>

***Listing 4-11.*** A Simple Replace Dashes Function

<script>

function stripDashes(txt) { return txt.split('-').join(' ');

};

console.log(stripDashes("super-basic-plan"));

console.log(stripDashes("something-with-a-lot-more-dashes-plan"));

console.log(stripDashes("noDashesPlan"));

</script>

***Listing 4-12.*** An Angular Filter Implementation myAppModule.filter('stripDashes', function () {

// the function we are in returns

// the function below return function(txt) {

return textToFilter.split('-').join(' ');

};

});

***Listing 4-13.*** Trying Out the stripDashes Filter

<!DOCTYPE html>

<html>

<head>

<title>Filter Demo</title>

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

<script src="js/myModules/myAppModule.js"></script>

</head>

<body ng-app="myAppModule" ng-controller="MyFilterDemoCtrl">

<p>Plan type: {{data.plan}}</p>

<p>Plan type: {{data.plan | stripDashes}}</p>

</body>

</html>

***Listing 4-14.*** A Basic Title Casing Function

<script>

function toTitleCase(str)

{

return str.charAt(0).toUpperCase() + str.substr(1).toLowerCase();

}

console.log(toTitleCase("jennifer")); console.log(toTitleCase("jENniFEr"));

console.log(toTitleCase("jENniFEr amanda Grant")); </script>

word, it is not a very forward-thinking approach. Let’s add the ability to handle multiple words (see Listing 4-15).

***Listing 4-15.*** A Better Title Casing Function

<script>

function toTitleCase(str) { return str.replace(/\w\S\*/g, function(txt){return txt.charAt(0).toUpperCase() + txt. substr(1).toLowerCase();});

}

console.log(toTitleCase("jennifer")); console.log(toTitleCase("jENniFEr"));

console.log(toTitleCase("jENniFEr amanda Grant"));

</script>

***Listing 4-16.*** An Angular Filter Implementation myAppModule.filter("toTitleCase", function () {

return function (str) { return str.replace(/\w\S\*/g, function(txt){ return txt.charAt(0).toUpperCase() + txt. substr(1).toLowerCase();});

};

});

***Listing 4-17.*** Using the toTitleCase Filter

<!DOCTYPE html>

<html>

<head>

<title>Filter Demo</title>

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

<script src="js/myModules/myAppModule.js"></script>

</head>

<body ng-app="myAppModule" ng-controller="MyFilterDemoCtrl">

<!-- Display customer name in title case -->

<p>First Name: {{data.firstName | toTitleCase}}</p>

<p>Surname: {{data.surname | toTitleCase}}</p>

<p>Full Name: {{ data.firstName + data.surname | toTitleCase}}</p>

</body></html>