Understanding AngularJS Factory, Service and Provider

In AngularJS, services are reusable singleton objects that are used to organize and share code across your app. They can be injected into controllers, filters, directives. AngularJS provides you three ways : service, factory and provider to create a service.

Factory

A factory is a simple function which allows you to add some logic before creating the object. It returns the created object.

Syntax

```
app.factory('serviceName',function(){ return serviceObj;})
```

Creating service using factory method

```
 <script>
//creating module
 var app = angular.module('app', []);

 //define a factory using factory() function
app.factory('MyFactory', function () {

 var serviceObj = {};
 serviceObj.function1 = function () {
 //TO DO:
 };

 serviceObj.function2 = function () {
 //TO DO:
 };

 return serviceObj;
});
 </script>
```

When to use

It is just a collection of functions like a class. Hence, it can be instantiated in different controllers when you are using it with constructor function.

Service

A service is a constructor function which creates the object using new keyword. You can add properties and functions to a service object by using this keyword. Unlike factory, it doesn't return anything.

Syntax

```
app.service('serviceName',function(){})
```

Creating service using service method

```
 <script>
//creating module
var app = angular.module('app', []);

//define a service using service() function
app.service('MyService', function () {
 this.function1 = function () {
 //TO DO:
 };

 this.function2 = function () {
 //TO DO:
 };
 });
 </script>
```

When to use

It is a singleton object. Use it when you need to share a single object across the application. **For example**, authenticated user details.

Provider

A provider is used to create a configurable service object. It returns value by using $get() function.

Syntax

```
//creating a service
app.provider('serviceName',function(){});

//configuring the service
app.config(function(serviceNameProvider){});
```

Creating service using provider method

```
 <script>
//define a provider using provider() function
app.provider('configurableService', function () {
 var name = '';
 this.setName = function (newName) {
 name = newName;
 };
 this.$get = function () {
 return name;
 };
});

//configuring provider using config() function
app.config(function (configurableService) {
 configurableService.setName('www.dotnet-tricks.com');
});
 </script>
```

When to use

When you need to provide module-wise configuration for your service object before making it available.

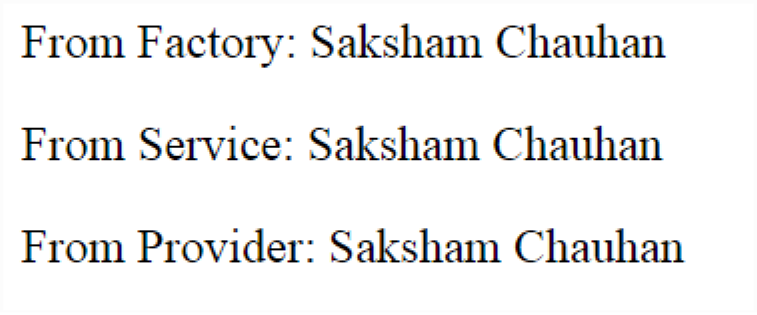
AngularJS : Factory, Service and Provider with example

```
<html>
<head>
 <title>AngularJS Service Factory and Providers</title>
 <script src="lib/angular.js"></script>
</head>
<body>
 <div class="container" style="padding-top:20px;">
 <div ng-app="myApp" ng-controller="myController">
 <p>From Service: </p>
```

```
<p>From Factory: </p>
<p>From Provider: </p>
</div>
</div>
<script>
//defining module
var app = angular.module('myApp', []);

//defining service
app.service('myService', function () {
this.name = '';
this.setName = function (newName) {
this.name = newName;
return this.name;
};
});

//defining factory
app.factory('myFactory', function () {
var serviceObj = {};
serviceObj.name = '';
serviceObj.setName = function (newName) {
serviceObj.name = newName;
};
return serviceObj;
});

//defining provider
app.provider('configurable', function () {
var privateName = '';
this.setName = function (newName) {
privateName = newName;
};
this.$get = function () {
return {
name: privateName
};
};
});

//configuring provider
app.config(function (configurableProvider) {
configurableProvider.setName("Saksham Chauhan");
});

//defining controller
app.controller('myController', function ($scope, myService, myFactory, config
urable) {
$scope.serviceName = myService.setName("Saksham Chauhan");

myFactory.setName("Saksham Chauhan");
$scope.factoryName = myFactory.name;

$scope.providerName = configurable.name;
});
```

```
 </script>
</body>
</html>
```

How it works...



```
From Factory: Saksham Chauhan

From Service: Saksham Chauhan

From Provider: Saksham Chauhan
```