PROVIDER IN ANGULARJS

- Each web application build in composed of objects. These objects need to be instantiated and wired together for the app. Angular apps most of these objects are instantiated and wired together automatically by **injector** **service**.
- A provider is actually a configurable factory. It accepts an object or constructor.
- AngularJS uses $provide to register new providers. The providers basically create new instances, but only once for each provider.
- The $provide service has additional methods to register services without specifying provider.
- These providers are available in **$provide**.
    - **Value** - It registers a value/object that can only be accessed by services not providers.
    - **Factory** - It is used to registers a service **factory function** that will be wrapped in a **service provider** object.
    - **Service** - It is used to registers a **constructor function** that will be wrapped in a **service provider** object. It is invoking a constructor with **new** operator.
    - **Constant** – It registers a value/object that can be accessed by providers and services.
    - **Provider** – It is used to register a **service provider** with **$injector**.

Syntax for providers in AngularJS:

```
module.provider('providername', function);
```

AngularJS provider:

Name	Description
$anchorScrollProvider	It is used to disable automatic scrolling at any time **$location.hash()** changes.
$animateProvider	**$animate** service is the default implementation that does not perform any animations instead of synchronously performs DOM updates.
$compileProvider	List a new directive with the compiler.
$controllerProvider	It allows controller registration through the register method. **$controller** service create a new controllers.
$filterProvider	Filters are funtions which transform input to an output. But filters need to be an Dependency Injected.
$httpProvider	It is used to change the default behavior of the $http service.
$interpolateProvider	It is used for aligning the interpolation markup. By default use this curly braces **{{** and **}}**
$locationProvider	To configure exactly how the application deep linking paths are stored.
$logProvider	It is used to configure in what way the application log messages.
$parseProvider	To configure the default behavior of the **$parse** service
$qProvider	This provider in module **ng**.

$rootScopeProvider	It is the provider of the $rootScope service.
$sceDelegateProvider	This provider allows developers to configure the **$sceDelegate** service.
$sceProvider	It allows developers to configure the **$sce** service
$templateRequestProvider	It is used to configure the options delivered to the $http service what time making a template request.

Sample Coding for providers in AngularJS:

```
<! DOCTYPE html>
<html>
   <head>
      <title>Wikitechy AngularJS tutorial</title>
      <script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.5.6/angular.min.js"></sc
ript>
   </head>
   <body>
      <div ng-app="myApp" ng-controller="providerCtrl">
         <h3>provider example in AngularJS Tutorial</h3>
         Display the name using provider: <b>{{ providervalue }}</b>
      </div>
      <script>
         var app = angular.module('myApp',[]);
         app.provider('unicorn', function() {
            this.setName = function(name) {
               this.name = name;
            };
            this.$get = function() {
               var self = this;
               return {
                  getvalue: function() {
```



```
                    return  self.name + "!!!";
                }
            };
        };
    });
    app.config(function(unicornProvider){
        unicornProvider.setName('welcome to AngularJS Tutorial');
    });
    app.controller('providerCtrl', function($scope,unicorn) {
        $scope.providervalue = unicorn.getvalue();
    });
</script>
</body>
</html>
```

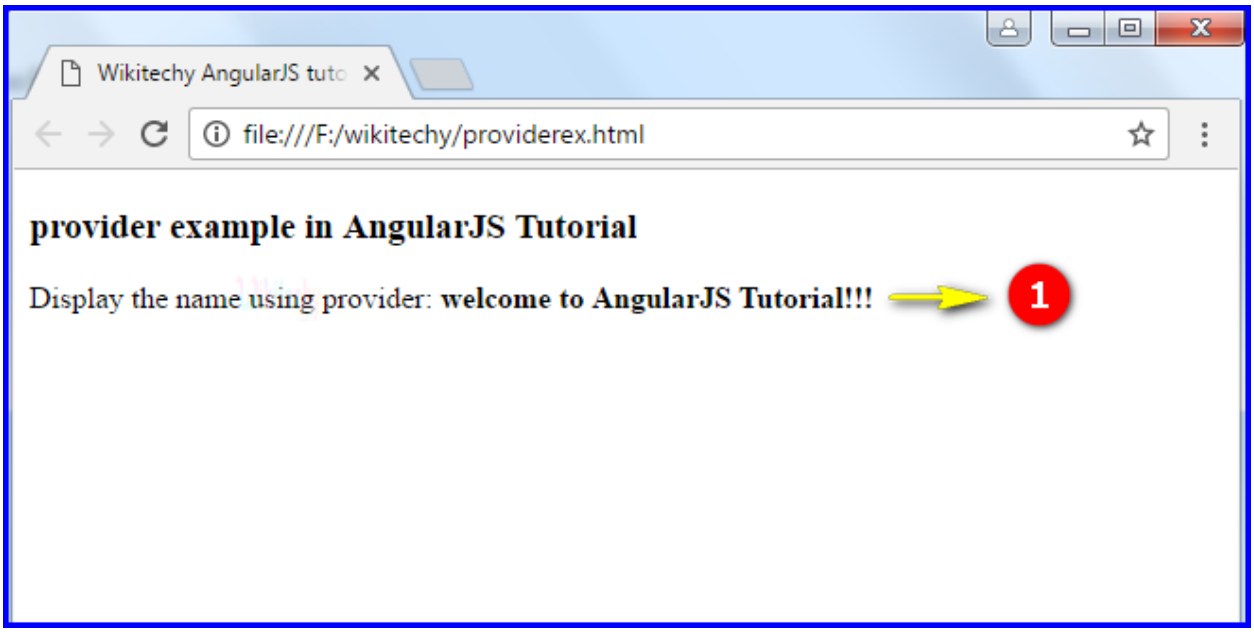

Code Explanation for providers in AngularJS:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Wikitechy AngularJS tutorial</title>
        <script src="https://ajax.googleapis.com/ajax/libs/
        angularjs/1.5.6/angular.min.js"></script>
    </head>
    <body>
        <div ng-app="myApp" ng-controller="providerCtrl">
            <h3>provider example in AngularJS Tutorial</h3>
            Display the name using provider: <b>{{ providervalue }}</b>
        </div>
        <script>
            var app = angular.module('myApp',[]);
            app.provider('unicorn', function() {
                this.setName = function(name) {
                    this.name = name;
                };
                this.$get = function() {
                    var self = this;
                    return {
                        getvalue: function() {
                            return  self.name + "!!!";
                        }
                    };
                };
            });
            app.config(function(unicornProvider){
                unicornProvider.setName('welcome to AngularJS Tutorial');
            });
            app.controller('providerCtrl', function($scope,unicorn) {
                $scope.providervalue = unicorn.getvalue();
            });
        </script>
    </body>
</html>
```

1. The **ng-app** specifies the root element ("myApp") to define AngularJS application.

2. The **ng-controller** control the data of **"providerCtrl"** in AngularJS application.
3. create a provider (app.provider) and the provider name is given as **"unicorn".**
4. Here set the name function (like this.setName=function(name)) and the name parameter is passed with function.
5. The name function value is returned by using the $get method on an instance of the function passed to unicorn provider.
6. The **provider("unicorn")** is injected to the config function.
7. The **"providerCtrl"** used to create the controller for the Application with arguments **$scope** object and **unicorn provider**.
8. The $scope.providervalue is used to get the name from the $get() method.

Sample Output for providers in AngularJS:



1. The content "welcome to AngularJS tutorial!!!" is displayed on the output.

