Differences between service, factory and provider in AngularJs

by Benny Bottema - Friday, January 17, 2014

http://www.bennybottema.com/2014/01/17/differences-between-service-factory-and-provider-in-angularjs/

The differences between a service, factory and provider are subtle.

It turns out that a service and factory are actually providers that differ in how they return a value. Let me just link the whole thing (credits go to Ben Clinkinbeard and Miško Hevery).

Contents

Services

Syntax: module.service('serviceName', function);
Result: When declaring serviceName as an injectable argument you will be provided with an instance of the function. In other words new FunctionYouPassedToService().

Factories

Syntax: module.factory('factoryName', function);
Result: When declaring factoryName as an injectable argument you will be provided with the value that is returned by invoking the function reference passed to module.factory.

Providers

Syntax: module.provider('providerName', function);
Result: When declaring providerName as an injectable argument you will be provided with new ProviderFunction().$get(). The constructor function is instantiated before the $get method is called – ProviderFunction is the function reference passed to module.provider.

Everything is a provider:

```
provider.service = function(name, Class) {
   provider.provide(name, function() {
      this.$get = function($injector) {
         return $injector.instantiate(Class);
      };
   });
}

provider.factory = function(name, factory) {
   provider.provide(name, function() {
      this.$get = function($injector) {
         return $injector.invoke(factory);
      };
   });
}

provider.value = function(name, value) {
   provider.factory(name, function() {
      return value;
   });
};
```