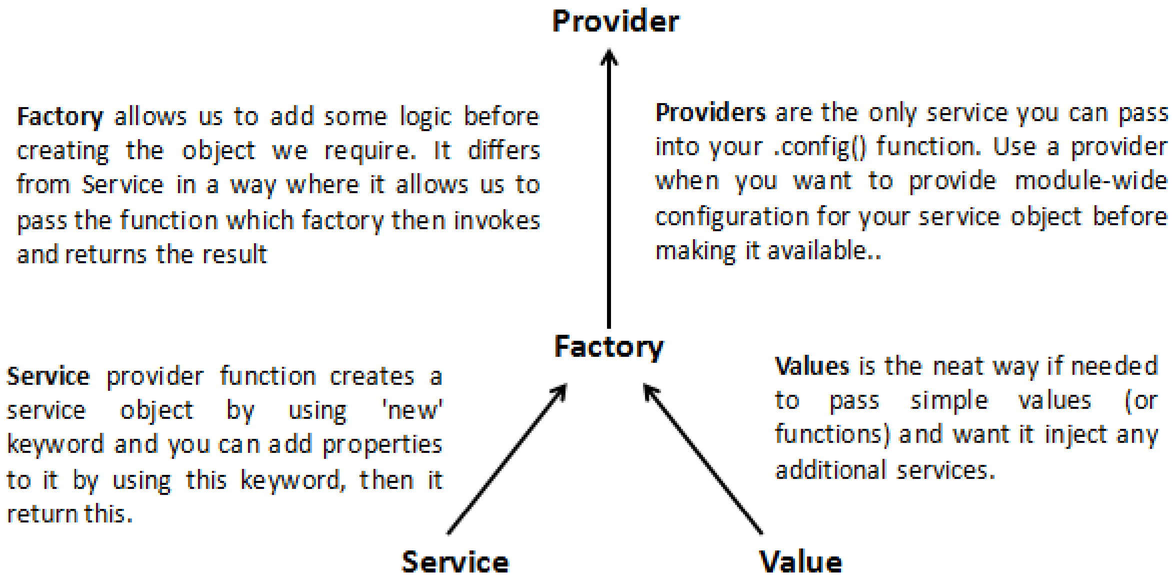
88

Service vs **Factory**

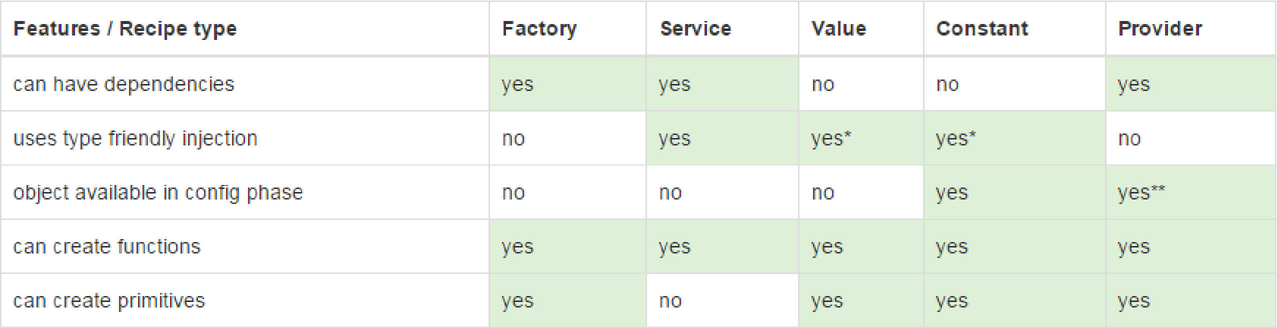


Factory allows us to add some logic before creating the object we require. It differs from Service in a way where it allows us to pass the function which factory then invokes and returns the result

Providers are the only service you can pass into your .config() function. Use a provider when you want to provide module-wide configuration for your service object before making it available..

Service provider function creates a service object by using 'new' keyword and you can add properties to it by using this keyword, then it return this.

Values is the neat way if needed to pass simple values (or functions) and want it inject any additional services.



Features / Recipe type	**Factory**	**Service**	**Value**	**Constant**	**Provider**
can have dependencies	yes	yes	no	no	yes
uses type friendly injection	no	yes	yes*	yes*	no
object available in config phase	no	no	no	yes	yes**
can create functions	yes	yes	yes	yes	yes
can create primitives	yes	no	yes	yes	yes

The difference between factory and service is just like the difference between a function and an object

Factory Provider

- Gives us the function's return value ie. You just create an object, add properties to it, then return that same object.When you pass this service into your controller, those properties on the object will now be available in that controller through your factory. (Hypothetical Scenario)
- Singleton and will only be created once
- Reusable components
- Factory are a great way for communicating between controllers like sharing data.
- Can use other dependencies
- Usually used when the service instance requires complex creation logic
- Cannot be injected in `.config()` function.
- Used for non configurable services

- If you're using an object, you could use the factory provider.
- Syntax: `module.factory('factoryName', function);`

Service Provider

- Gives us the instance of a function (object)- You just instantiated with the ‘new’ keyword and you’ll add properties to ‘this’ and the service will return ‘this’.When you pass the service into your controller, those properties on ‘this’ will now be available on that controller through your service. (Hypothetical Scenario)
- Singleton and will only be created once
- Reusable components
- Services are used for communication between controllers to share data
- You can add properties and functions to a service object by using the `this` keyword
- Dependencies are injected as constructor arguments
- Used for simple creation logic
- Cannot be injected in `.config()` function.
- If you're using a class you could use the service provider
- Syntax: `module.service(‘serviceName’, function);`