What's a FactoryBean?

JOSH LONG (/TEAM/JLONG) AUGUST 09, 2011

5 COMMENTS (/BLOG/2011/08/09/WHAT-S-A-FACTORYBEAN#DISQUS_THREAD)

In this post, I'll look at Spring's `org.springframework.beans.factory.FactoryBean<T>` interface. The definition of this interface is:

```
public interface FactoryBean<T> {
  T getObject() throws Exception;
  Class<T> getObjectType();
  boolean isSingleton();
}
```

A `FactoryBean` is a pattern to encapsulate interesting object construction logic in a class. It might be used, for example, to encode the construction of a complex object graph in a reusable way. Often this is used to construct complex objects that have many dependencies. It might also be used when the construction logic itself is highly volatile and depends on the configuration. A `FactoryBean` is also useful to help Spring construct objects that it couldn't easily construct itself. For example, in order to inject a reference to a bean that was obtained from JNDI, the reference must first be obtained. You can use the `JndiFactoryBean` to obtain this reference in a consistent way. You may inject the result of a `FactoryBean`'s `getObject()` method into any other property.

Suppose you have a `Person` class whose definition is thus:

```
public class Person {
 private Car car ;
 private void setCar(Car car){ this.car = car;  }
}
```

and a `FactoryBean` whose definition is thus: