

# Generic

## Outline

Spring4 Generic supplements the conventional Autowire and Qualifier to support Generic service.

## Description

The section describes Generic-type Autowire as an functional expansion from the conventional Autowirer and Qualifier.

### Autowire and Qualifier

### Autowiring Code Example

Refer to the following code example for how to Autowire the Customer class for person property:

```
package com.egovframe.common;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;

public class Customer {

    @Autowired
    private Person person;
    //...
}
```

### Autowire Limitations

Refer to the following code example that best represent the case Person cannot be subject to Autowire:

```
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-2.5.xsd">

    <bean
class ="org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor"/>

    <bean id="customer" class="com.egovframe.common.Customer" />

    <bean id="personA" class="com.egovframe.common.Person" >
        <property name="name" value="GentlemanA" />
    </bean>

    <bean id="personB" class="com.egovframe.common.Person" >
        <property name="name" value="GentlemanB" />
    </bean>

</beans>
```

Existence of the same types causes the following error:

Caused by: org.springframework.beans.factory.NoSuchBeanDefinitionException:  
No unique bean of type [com.egovframe.common.Person] is defined:  
expected single matching bean but found 2: [personA, personB]

## How to apply Qualifier

You may specify personA for each Annotation Qualifier if you're not bothered by matching them one by one.

```
package com.egovframe.common;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Qualifier;

public class Customer {

    @Autowired
    @Qualifier("personA")
    private Person person;
    //...
}
```

## Spring 4 Generics

### How to configure Generic

You can configure Manager and Admin implementing the abstract class employee as Generic, in which case you'll need to inject Autowire to Spring 4 Generic.

```
package generic;

import org.springframework.beans.factory.annotation.Autowired;

public class InjectBeans {
    @Autowired
    Employee<Manager> emp1;

    @Autowired
    Employee<Admin> emp2;

    public void invokeManager(){
        emp1.empSection();
    }

    public void invokeAdmin(){
        emp2.empSection();
    }
}
```

### Implemented Classes subject to injection

Employee is an abstract class that is implemented via inheritance. Keep in mind, however, Admin and Manager are an implementation class that are auto-injected via Annotation Autowire.

```
package generic;

public abstract class Employee<T> {
    public void printCompanyName(){
        System.out.println("Display : Company Name");
    }
}
```

```

    public abstract void empSection();
}
package generic;

public class Admin extends Employee<Admin>{
    public void empSection(){
        System.out.println("Display : Admin Section");
    }
}
package generic;

public class Manager extends Employee<Manager>{
    public void empSection(){
        System.out.println("Display : Manager Section");
    }
}

```

## Output

Output of executing empSection() for each implementation class is as follows:

```

Jun. 23, 2015 11:23:54 AM org.springframework.context.annotation.AnnotationConfigApplicationContext
prepareRefresh
INFO: Refreshing org.springframework.context.annotation.AnnotationConfigApplicationContext@1262a85: startup
date [Tue Jun 23 11:23:54 KST 2015]; root of context hierarchy
Display : Admin Section
Display : Manager Section

```

## Compilation and Execution in Spring 3

Though we have no compilation problems in Spring 3, you will encounter the following error when the Type Generic is not infused over runtime:

```

Jun. 23, 2015 10:38:25 AM org.springframework.context.annotation.AnnotationConfigApplicationContext
prepareRefresh
INFO: Refreshing org.springframework.context.annotation.AnnotationConfigApplicationContext@1110f31: startup
date [Tue Jun 23 10:38:25 KST 2015]; root of context hierarchy
Jun. 23, 2015 10:38:26 AM org.springframework.beans.factory.support.DefaultListableBeanFactory
preInstantiateSingletons
INFO: Pre-instantiating singletons in
org.springframework.beans.factory.support.DefaultListableBeanFactory@17f70bf: defining beans
[org.springframework.context.annotation.internalConfigurationAnnotationProcessor,org.springframework.context.ann
otation.internalAutowiredAnnotationProcessor,org.springframework.context.annotation.internalRequiredAnnotationPr
ocessor,org.springframework.context.annotation.internalCommonAnnotationProcessor,springMVCConfiguration,org.
springframework.context.annotation.ConfigurationClassPostProcessor.importAwareProcessor,getManager,InjectBean
s,getAdmin]; root of factory hierarchy
Jun. 23, 2015 10:38:26 AM org.springframework.beans.factory.support.DefaultListableBeanFactory
destroySingletons
INFO: Destroying singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@17f70bf:
defining beans
[org.springframework.context.annotation.internalConfigurationAnnotationProcessor,org.springframework.context.ann
otation.internalAutowiredAnnotationProcessor,org.springframework.context.annotation.internalRequiredAnnotationPr
ocessor,org.springframework.context.annotation.internalCommonAnnotationProcessor,springMVCConfiguration,org.
springframework.context.annotation.ConfigurationClassPostProcessor.importAwareProcessor,getManager,InjectBean
s,getAdmin]; root of factory hierarchy
Exception in thread "main" org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'InjectBeans': Injection of autowired dependencies failed; nested exception is
org.springframework.beans.factory.BeanCreationException: Could not autowire field: generic.Employee
generic.InjectBeans.emp1; nested exception is org.springframework.beans.factory.NoUniqueBeanDefinitionException:

```

No qualifying bean of type [generic.Employee] is defined: expected single matching bean but found 2:  
getManager.getAdmin  
at  
org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor.postProcessPropertyValues(A  
utowiredAnnotationBeanPostProcessor.java:289)  
at  
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.populateBean(AbstractAutowireCa  
pableBeanFactory.java:1146)  
at  
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCreateBean(AbstractAutowireC  
apableBeanFactory.java:519)  
at  
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.createBean(AbstractAutowireCapa  
bleBeanFactory.java:458)  
at  
org.springframework.beans.factory.support.AbstractBeanFactory\$1.getObject(AbstractBeanFactory.java:296)  
at  
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleton(DefaultSingletonBeanRegistr  
y.java:223)  
at  
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBeanFactory.java:293)  
at org.springframework.beans.factory.support.AbstractBeanFactory.getBean(AbstractBeanFactory.java:194)  
at  
org.springframework.beans.factory.support.DefaultListableBeanFactory.preInstantiateSingletons(DefaultListableBean  
Factory.java:633)  
at  
org.springframework.context.support.AbstractApplicationContext.finishBeanFactoryInitialization(AbstractApplicatio  
nContext.java:932)  
at  
org.springframework.context.support.AbstractApplicationContext.refresh(AbstractApplicationContext.java:479)  
at  
org.springframework.context.annotation.AnnotationConfigApplicationContext.<init>(AnnotationConfigApplicationC  
ontext.java:73)  
at generic.SpringMVCAplicationDemo.main(SpringMVCAplicationDemo.java:15)  
Caused by: org.springframework.beans.factory.BeanCreationException: Could not autowire field: generic.Employee  
generic.InjectBeans.emp1; nested exception is org.springframework.beans.factory.NoUniqueBeanDefinitionException:  
No qualifying bean of type [generic.Employee] is defined: expected single matching bean but found 2:  
getManager.getAdmin  
getManager.getAdmin  
at  
org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor\$AutowiredFieldElement.inje  
ct(AutowiredAnnotationBeanPostProcessor.java:517)  
at org.springframework.beans.factory.annotation.InjectionMetadata.inject(InjectionMetadata.java:87)  
at  
org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor.postProcessPropertyValues(A  
utowiredAnnotationBeanPostProcessor.java:286)  
... 12 more  
Caused by: org.springframework.beans.factory.NoUniqueBeanDefinitionException: No qualifying bean of type  
[generic.Employee] is defined: expected single matching bean but found 2: getManager.getAdmin  
getManager.getAdmin  
at  
org.springframework.beans.factory.support.DefaultListableBeanFactory.doResolveDependency(DefaultListableBeanF  
actory.java:870)  
at  
org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveDependency(DefaultListableBeanFact  
ory.java:775)  
at  
org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor\$AutowiredFieldElement.inje  
ct(AutowiredAnnotationBeanPostProcessor.java:489)  
... 14 more