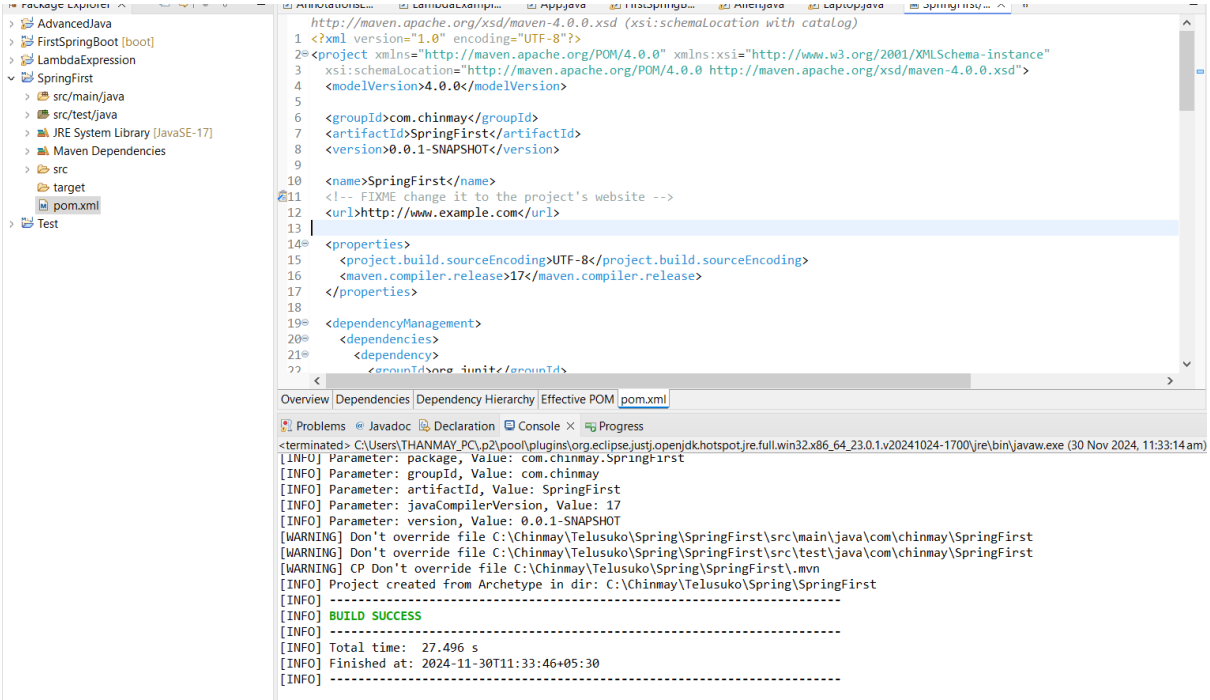
139.Spring 1st Project



Once we create a new Project it will be like above image.

If we want to create an Container we have to use ApplicationContext/BeanFactory.

ApplicationContext is responsible to work with that container.

ApplicationContext/BeanFactory helps us to create the Container and get the Objects from the container.

ApplicationContext provides all the features of BeanFactory plus some additional features.

ApplicationContext is not a part of Java it is part of Spring .So we need to include the dependencies for Spring Context.

Include the required dependencies in the pom.xml file .Post which the dependencies will be downloaded into the Project.

ApplicationContext is parent class for BeanFactory as well.

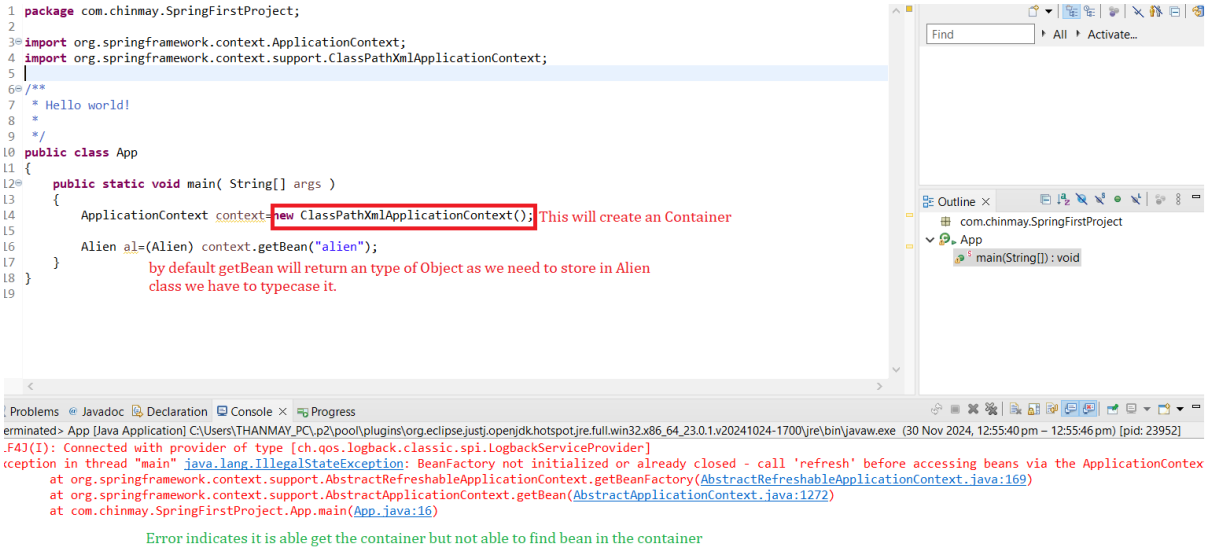
There are different to configure the Spring Application

1.Xml

2.Annotation

3.Java based

getBean()-will get the Object from the container and provide to us.



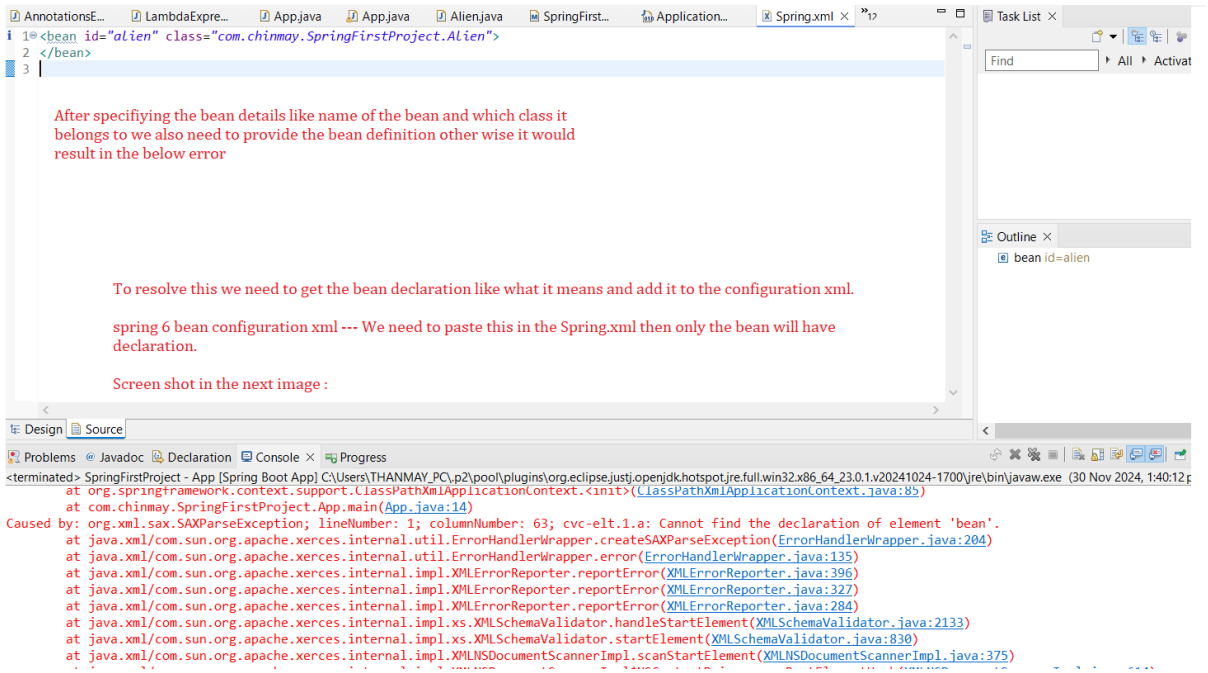
140.Spring Bean Xml Config

We need to Provide the Configurations to the container like which Objects has to be created by container and maintained by container using the xml file.

This file has to be placed in the resource section(src/main/resources).

The Same file name has to be passed as input to the container.

```
ApplicationContext context=new ClassPathXmlApplicationContext("Spring.xml");
```



AnnotationsE... LambdaExpre... App.java App.java Alien.java SpringFirst... Application... Spring.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<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.xsd">
<bean id="alien" class="com.chinmay.SpringFirstProject.Alien">
</bean>
</beans>
```

AnnotationsE... LambdaExpre... App.java App.java Alien.java SpringFirst... Application... Spring.xml

```java
package com.chinmay.SpringFirstProject;


public class Alien {

    public void code()
    {
    System.out.println("Coding");
    }
}
```

Problems Javadoc Declaration Console Progress

<terminated> SpringFirstProject - App [Spring Boot App] C:\Users\THANMAY_PC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v202

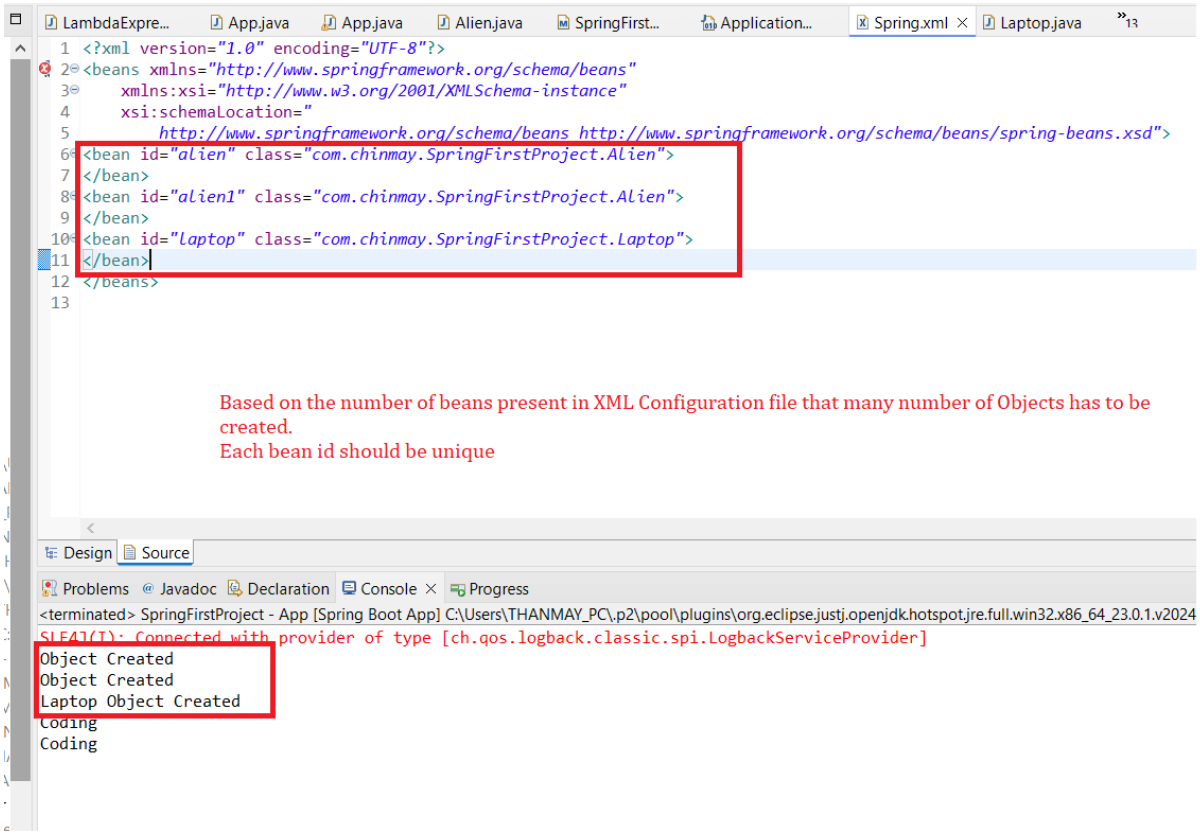
```
SLF4J(I): Connected with provider of type [ch.qos.logback.classic.spi.LogbackServiceProvider]
Coding
```

141.Object Creation

This creates Objects in the container based on the configurations provided to them in Xml(Spring.xml).

```
ApplicationContext context=new ClassPathXmlApplicationContext("Spring.xml");
```

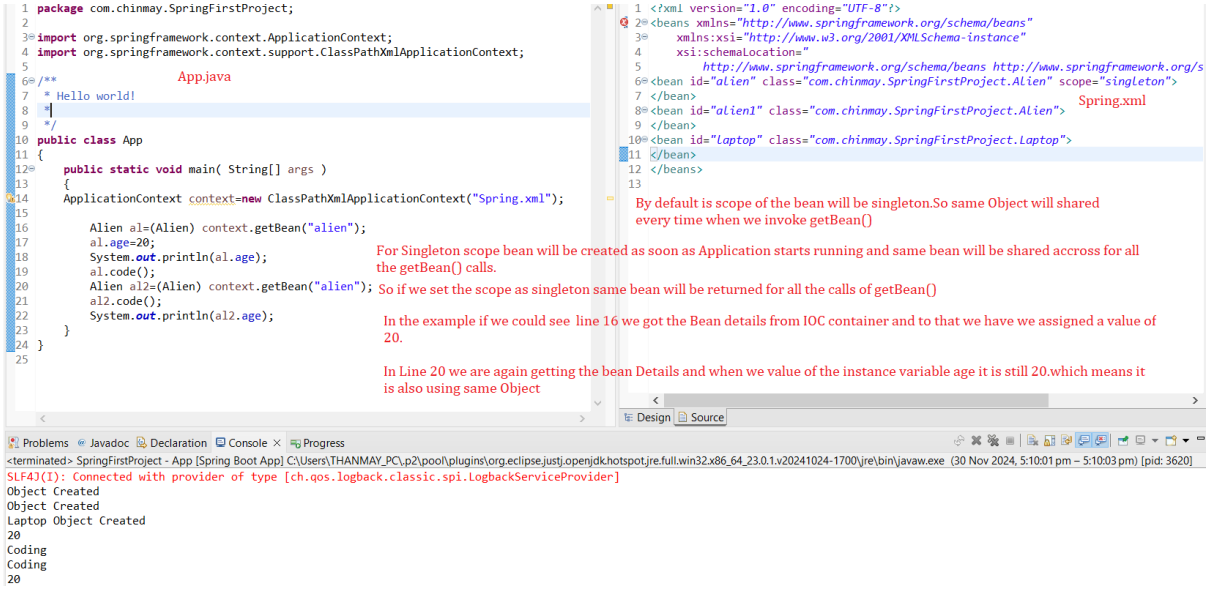
Line due to which Objects will be created in the container based on the configurations provided in the XML file



142.Scope

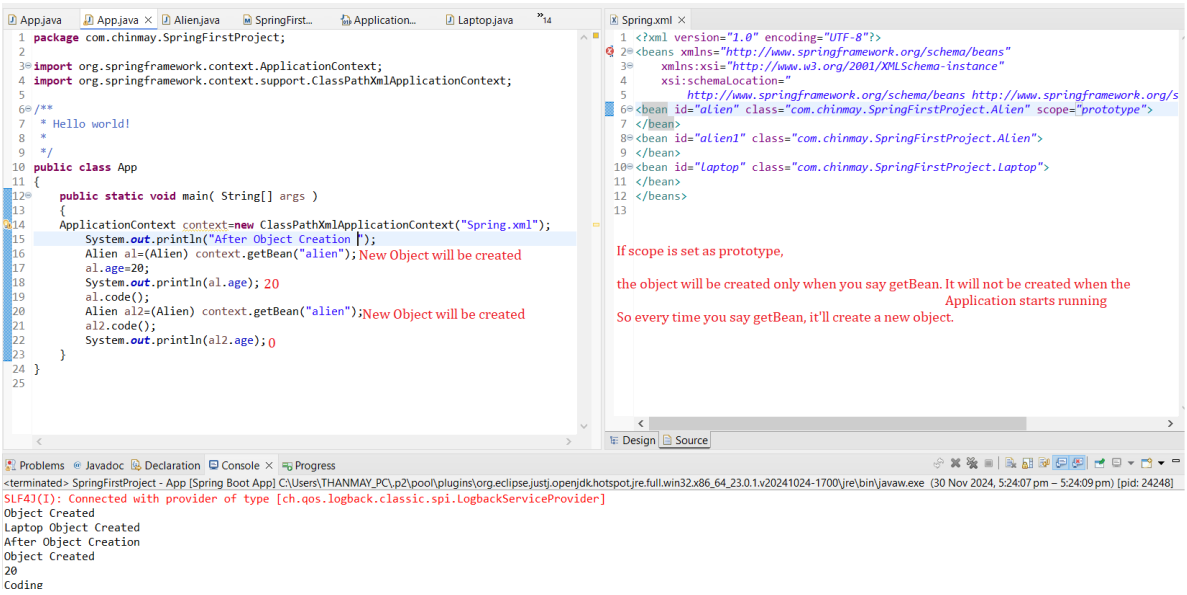
By default scope will be Single ton if we don't mention

Singleton :



Prototype :

If the scope will be set as prototype then every time a getBean() call is made a new Object will be created.

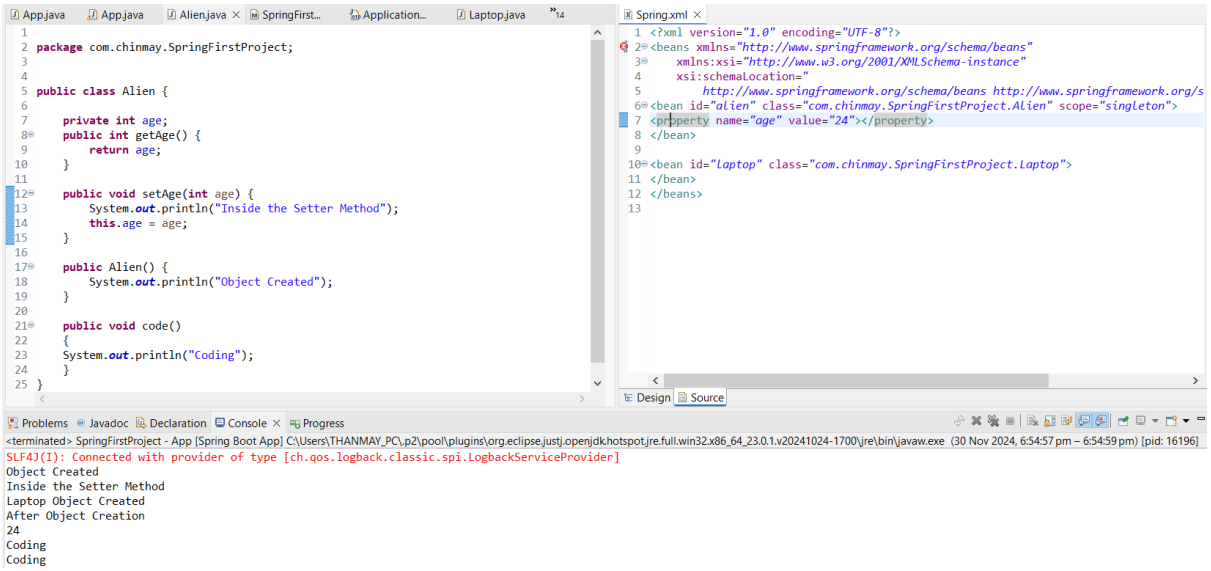


Session : Not discussed

143: Setter Injection/Setter Dependency Injection :

Consider when we want to inject something .

When we inject something into the class using setters then we call Setter Injection we can do that in the below way.

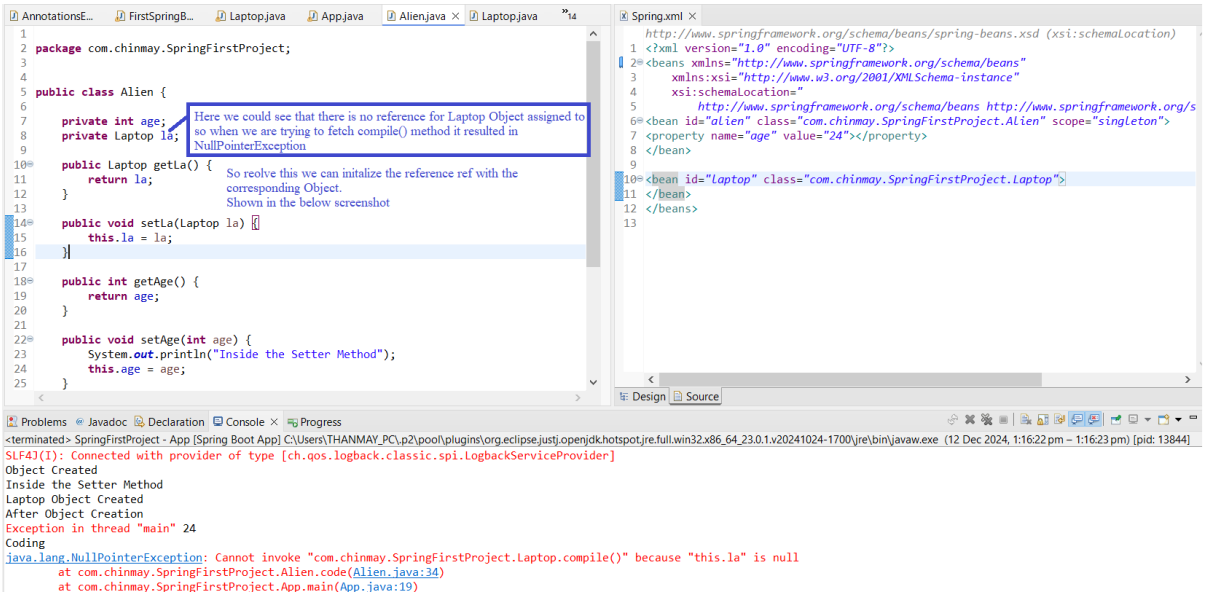


```
package com.chinmay.SpringFirstProject;


public class Alien {

    private int age;
    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        System.out.println("Inside the Setter Method");
        this.age = age;
    }

    public Alien() {
        System.out.println("Object Created");
    }

    public void code()
    {
    System.out.println("Coding");
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<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/s
<bean id="alien" class="com.chinmay.SpringFirstProject.Alien" scope="singleton">
<property name="age" value="24"></property>
</bean>

<bean id="laptop" class="com.chinmay.SpringFirstProject.Laptop">
</bean>
</beans>
```

```
<terminated> SpringFirstProject - App [Spring Boot App] C:\Users\THANMAY_PC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe  (30 Nov 2024, 6:54:57 pm – 6:54:59 pm) [pid: 16196]
SLF4J(I): Connected with provider of type [ch.qos.logback.classic.spi.LogbackServiceProvider]
Object Created
Inside the Setter Method
Laptop Object Created
After Object Creation
24
Coding
Coding
```

144.Ref Attribute/Setter Injection with Ref Attribute



```
package com.chinmay.SpringFirstProject;


public class Alien {

    private int age;
    private Laptop la;

    public Laptop getLa() {
        return la;
    }

    public void setLa(Laptop la) {
        this.la = la;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        System.out.println("Inside the Setter Method");
        this.age = age;
    }
```

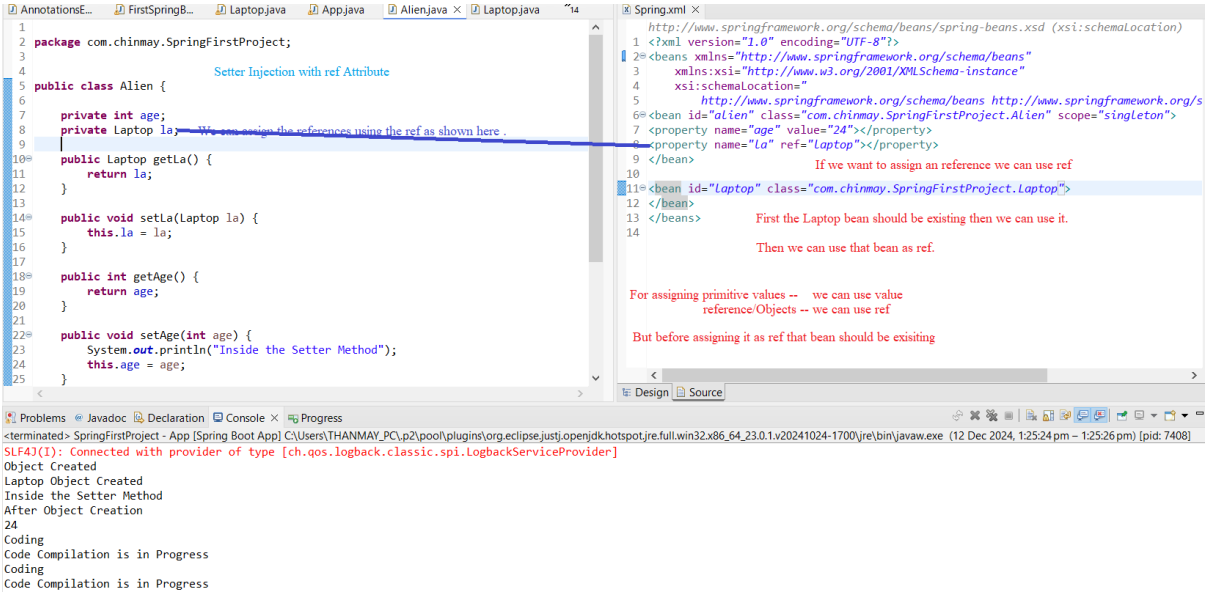
Here we could see that there is no reference for Laptop Object assigned to so when we are trying to fetch compile() method it resulted in NullPointerException

So reolve this we can initalize the reference ref with the corresponding Object.
Shown in the below screenshot

```
<?xml version="1.0" encoding="UTF-8"?>
<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/s
<bean id="alien" class="com.chinmay.SpringFirstProject.Alien" scope="singleton">
<property name="age" value="24"></property>
</bean>

<bean id="laptop" class="com.chinmay.SpringFirstProject.Laptop">
</bean>
</beans>
```

```
<terminated> SpringFirstProject - App [Spring Boot App] C:\Users\THANMAY_PC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-1700\jre\bin\javaw.exe  (12 Dec 2024, 1:16:22 pm – 1:16:23 pm) [pid: 13844]
SLF4J(I): Connected with provider of type [ch.qos.logback.classic.spi.LogbackServiceProvider]
Object Created
Inside the Setter Method
Laptop Object Created
After Object Creation
Exception in thread "main" 24
Coding
java.lang.NullPointerException: Cannot invoke "com.chinmay.SpringFirstProject.Laptop.compile()" because "this.la" is null
        at com.chinmay.SpringFirstProject.Alien.code(Alien.java:34)
        at com.chinmay.SpringFirstProject.App.main(App.java:19)
```

Issue:



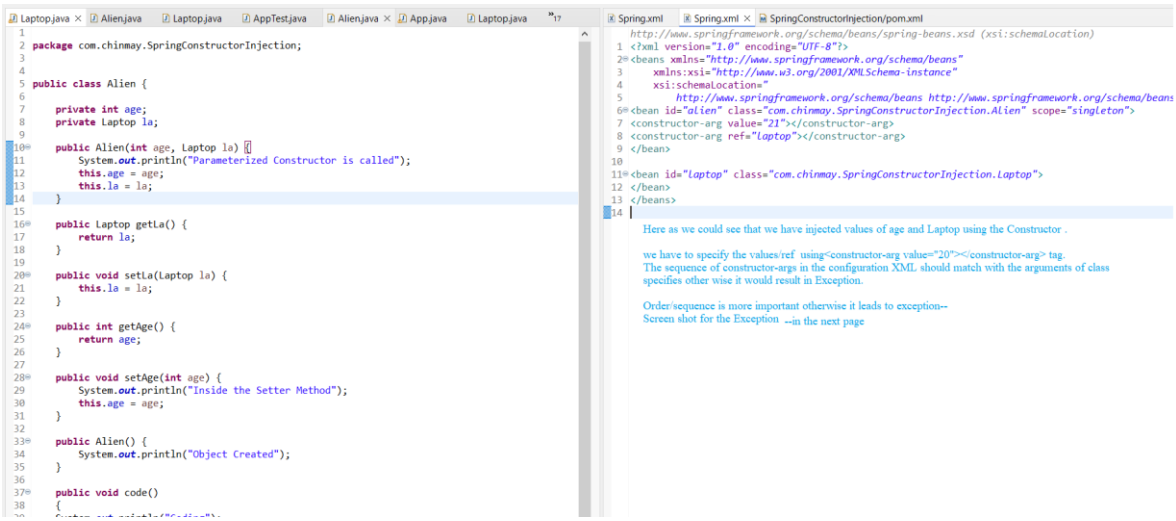
Solution

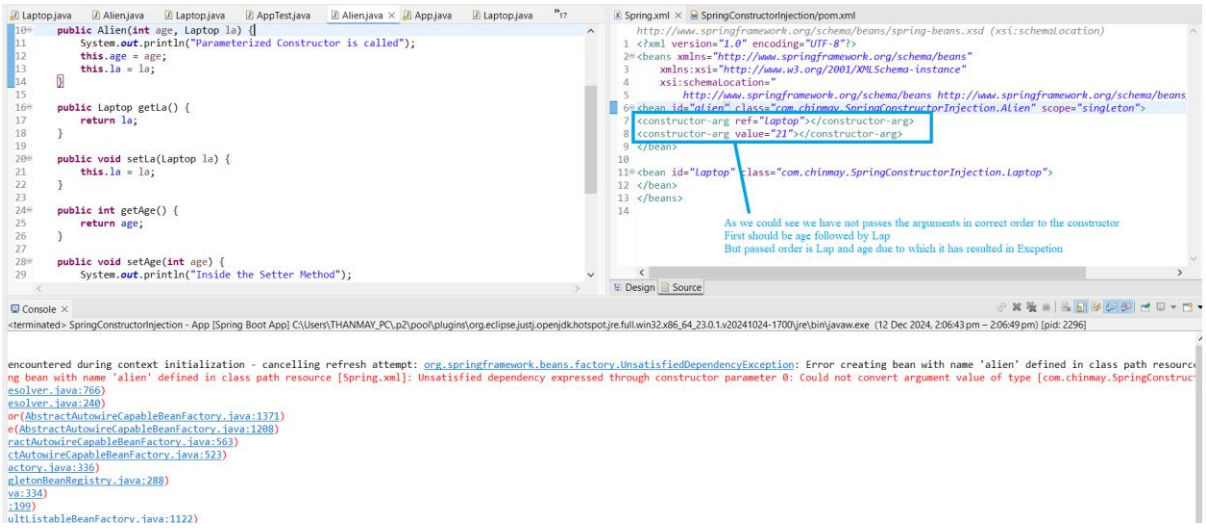
145: Constructor Injection:

We can initialize the values of class using the constructor as well.

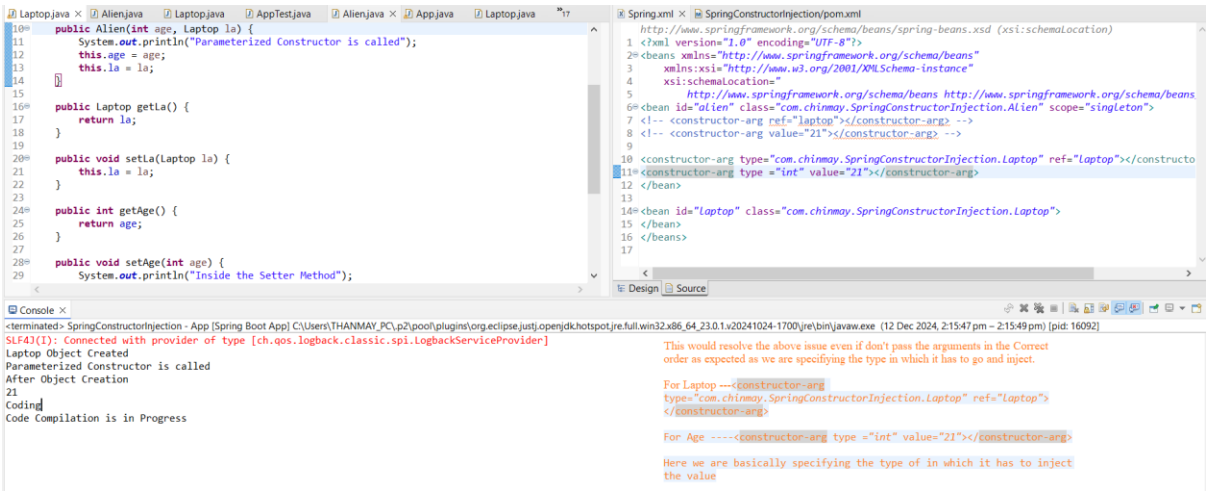
<constructor-arg value="*21*"></constructor-arg>

<constructor-arg ref="*laptop*"></constructor-arg>





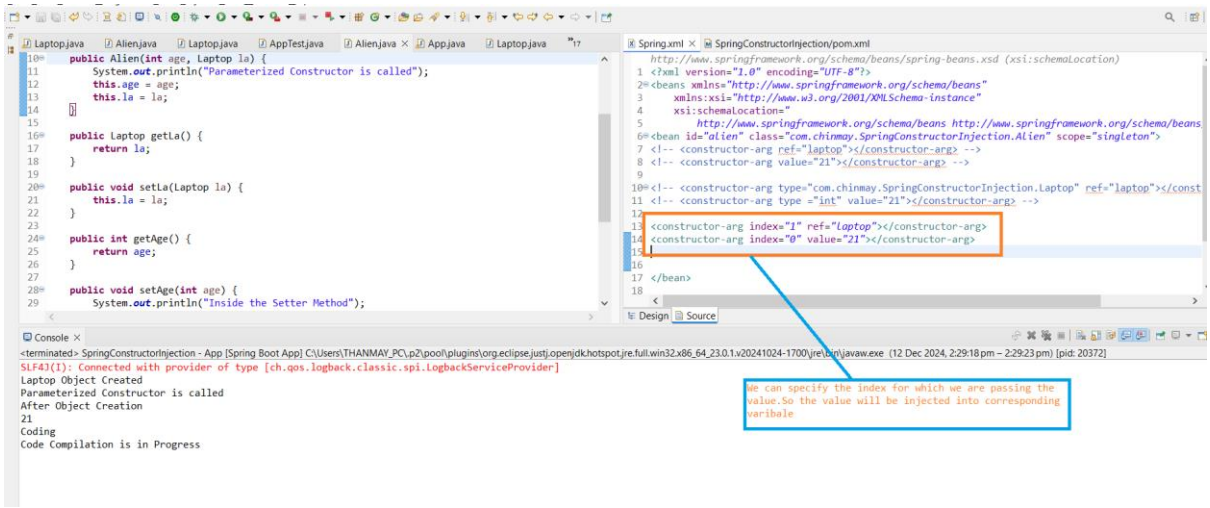
To resolve the issue we can use type to which type it has to assign that value. Find the screenshot for the same below.



But there is issue with above approach as well consider we have two int values and we are injecting using <construct-args>as we are specifying only the type only for injecting it will be ambiguity as to insert in to insert to which attribute even if we specify the value for both the values.

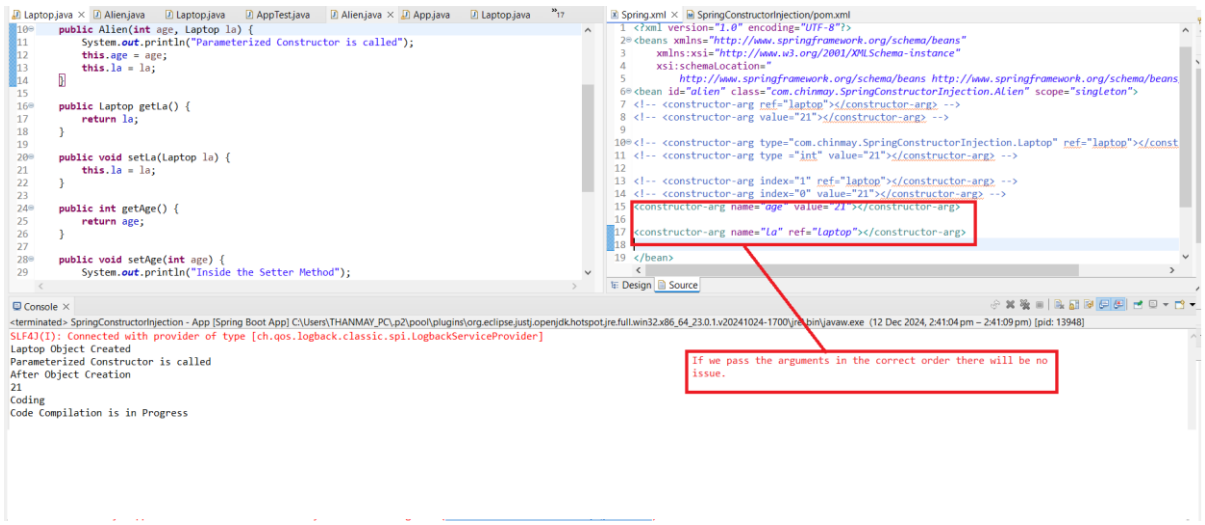
```
<constructor-arg type="com.chinmay.SpringConstructorInjection.Laptop" ref="laptop"></constructo
<constructor-arg type ="int" value="21"></constructor-arg>
<constructor-arg type ="int" value="21000"></constructor-arg>
```

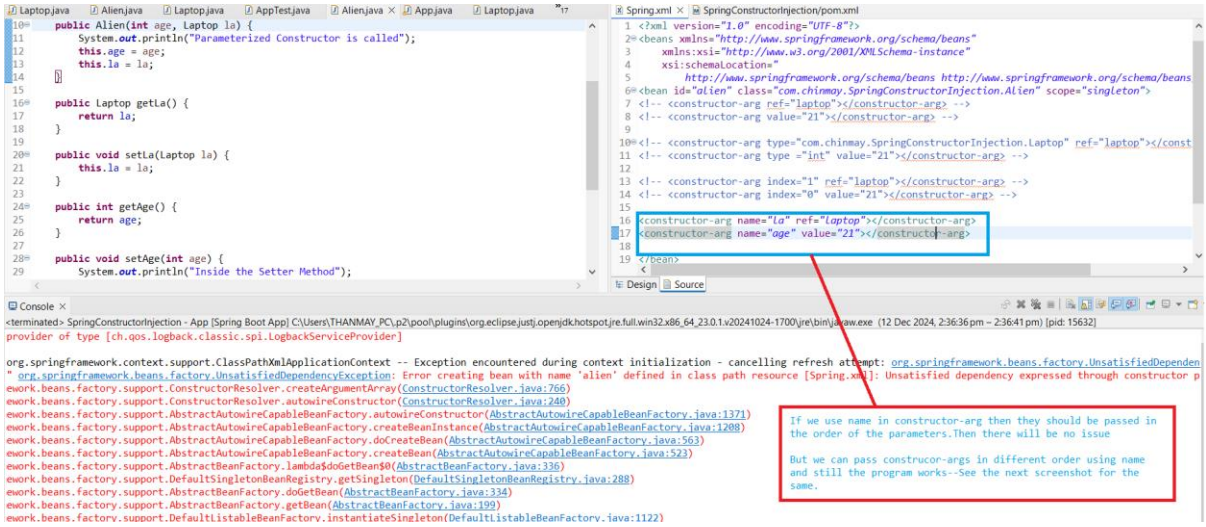
To Resolve the above issue also we can use index in the Constructor-arg.



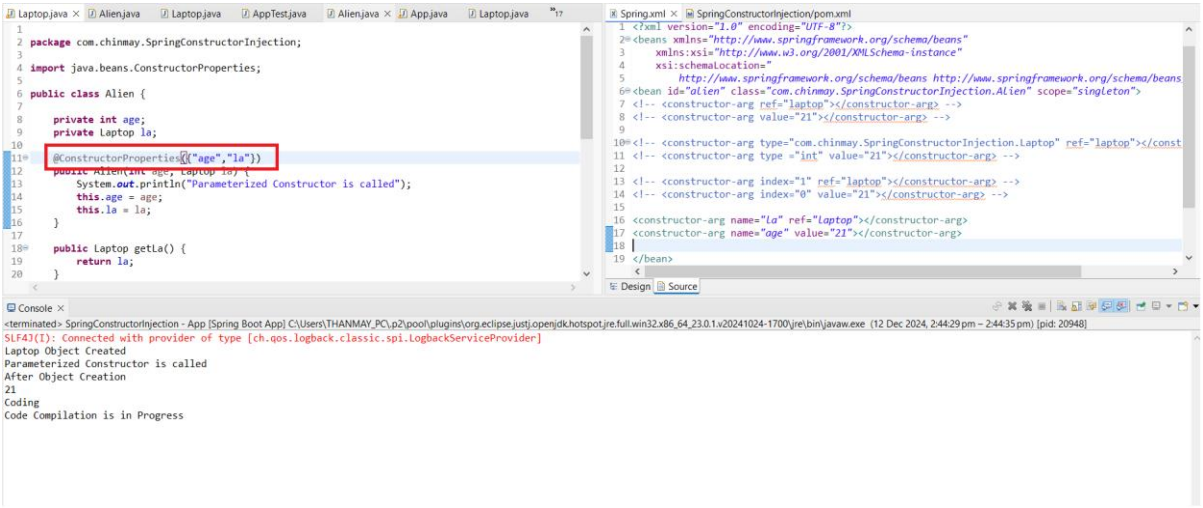
One more Way :

We can use name as well in the constructor-arg tag.

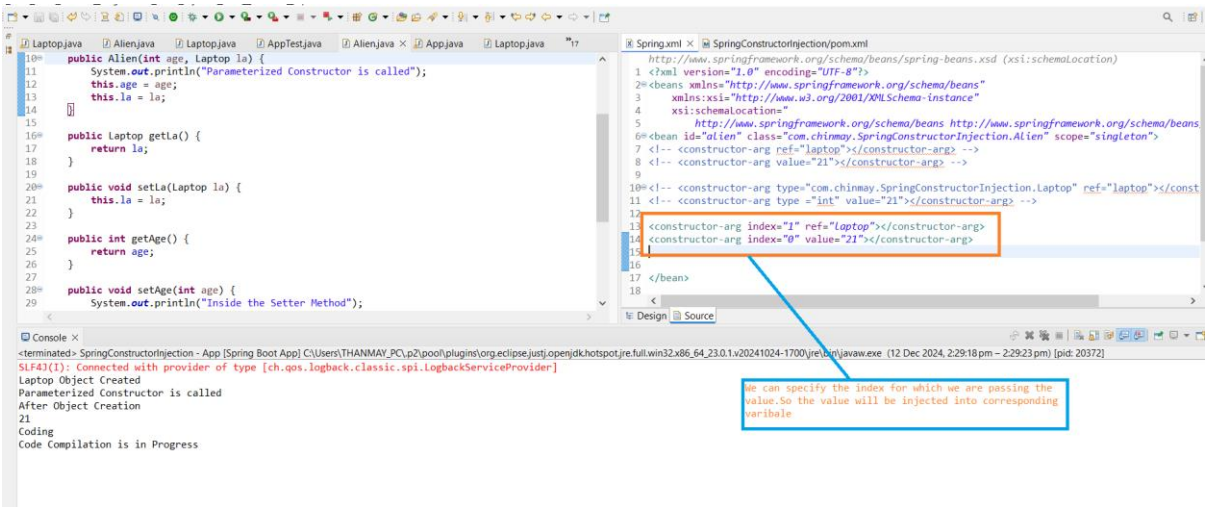




To resolve the above issue:



But it will be easier if we used index based constructor-arg.Go which index based constructor-arg



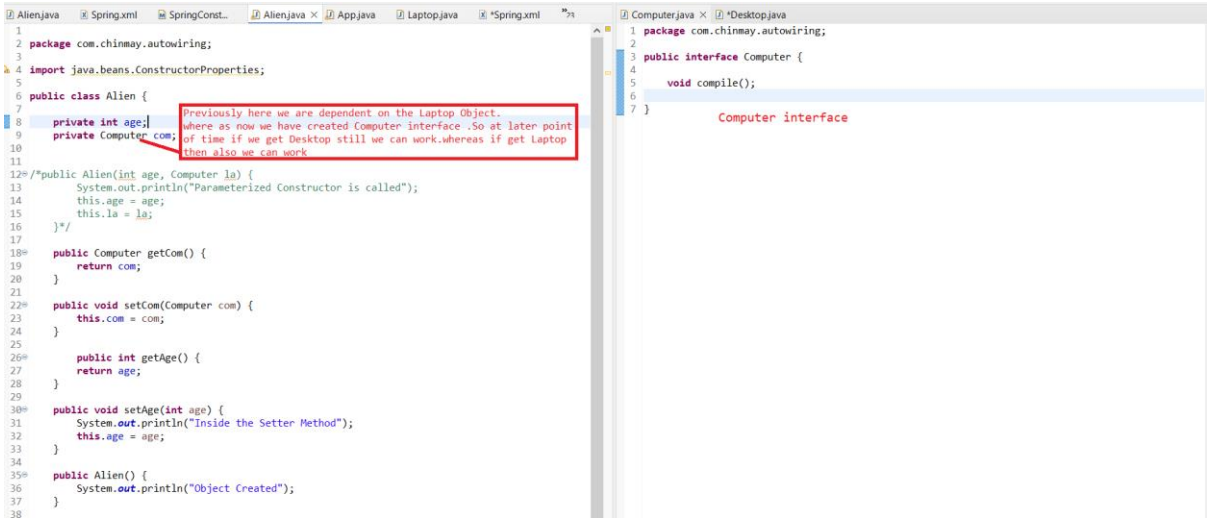
This will be better approach.

Among setter and constructor which one to use:

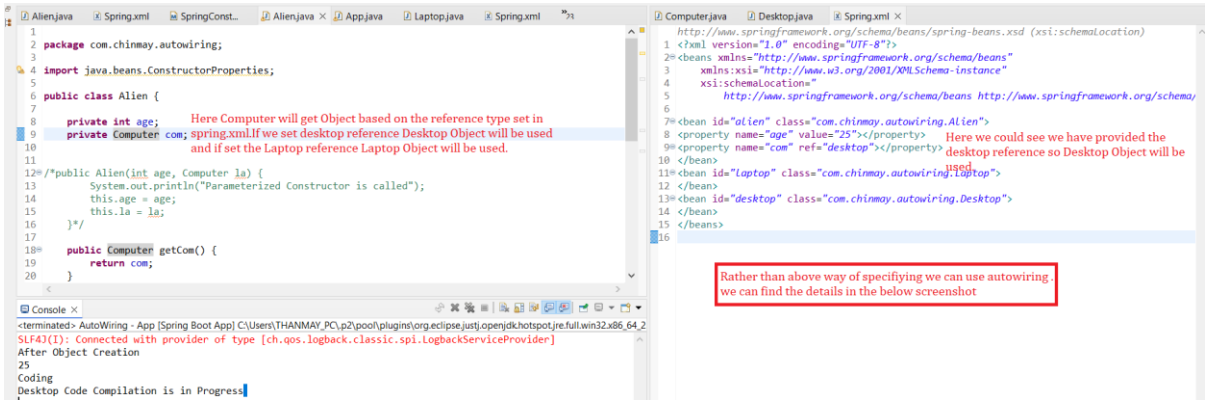
If the values for Properties are Optional, we can use setter injection.

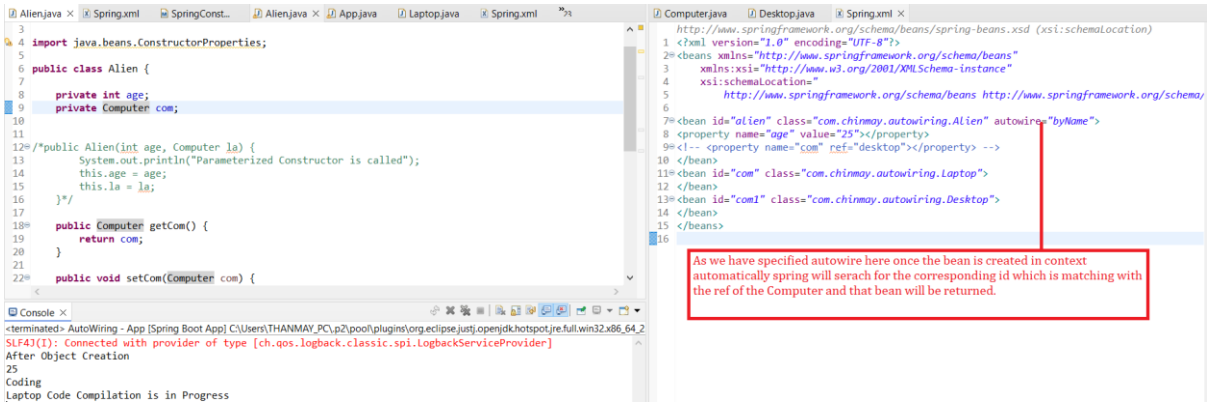
If the values for Properties are mandatory we can use constructor injection. Because we have to pass all the values for the corresponding constructor to be called.

146.Creating interface

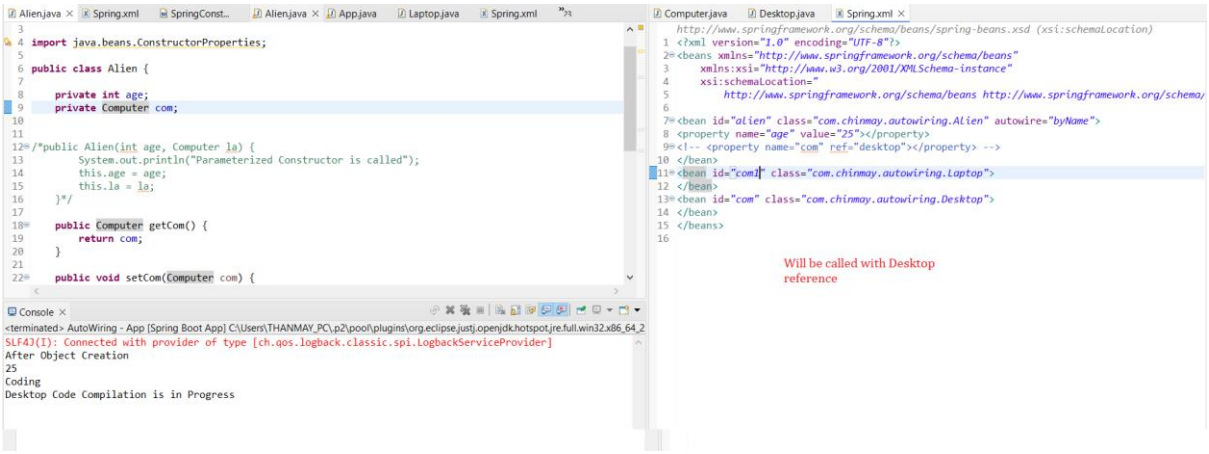


147.Autowiring





byName



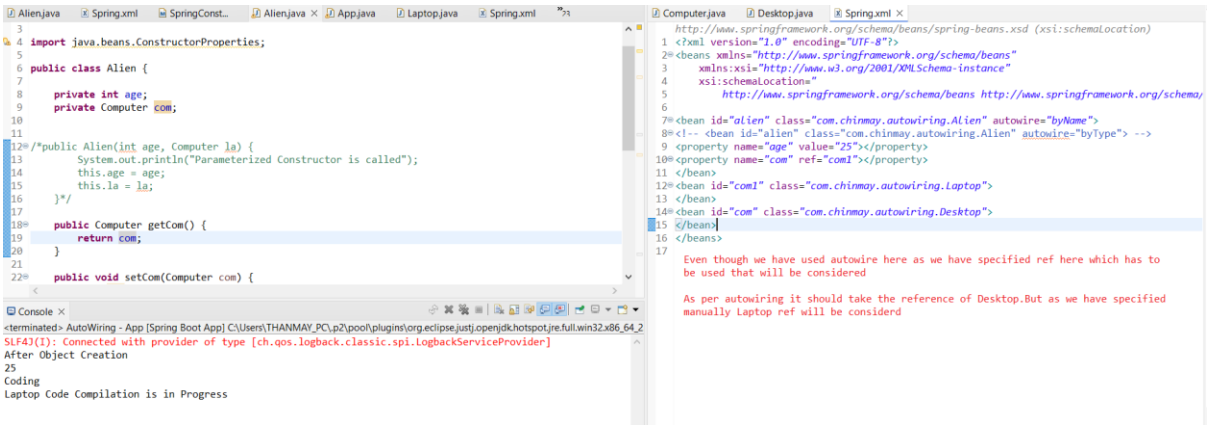
```
import java.beans.ConstructorProperties;

public class Alien {

    private int age;
    private Computer com;


/*public Alien(int age, Computer la) {
        System.out.println("Parameterized Constructor is called");
        this.age = age;
        this.la = la;
    }*/

    public Computer getCom() {
        return com;
    }

    public void setCom(Computer com) {
```

```
Console
<terminated> AutoWiring - App [Spring Boot App] C:\Users\THANMAY_PC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
SLF4J(I): Connected with provider of type [ch.qos.logback.classic.spi.LogbackServiceProvider]
After Object Creation
25
Coding
Desktop Code Compilation is in Progress
```

```
http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
<?xml version="1.0" encoding="UTF-8"?>
<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/
<bean id="alien" class="com.chinmay.autowiring.Alien" autowire="byName">
<property name="age" value="25"></property>
<!-- <property name="com" ref="desktop"></property> -->
</bean>
<bean id="com1" class="com.chinmay.autowiring.Laptop">
</bean>
<bean id="com" class="com.chinmay.autowiring.Desktop">
</bean>
</beans>
```

Will be called with Desktop reference

byName--Desktop



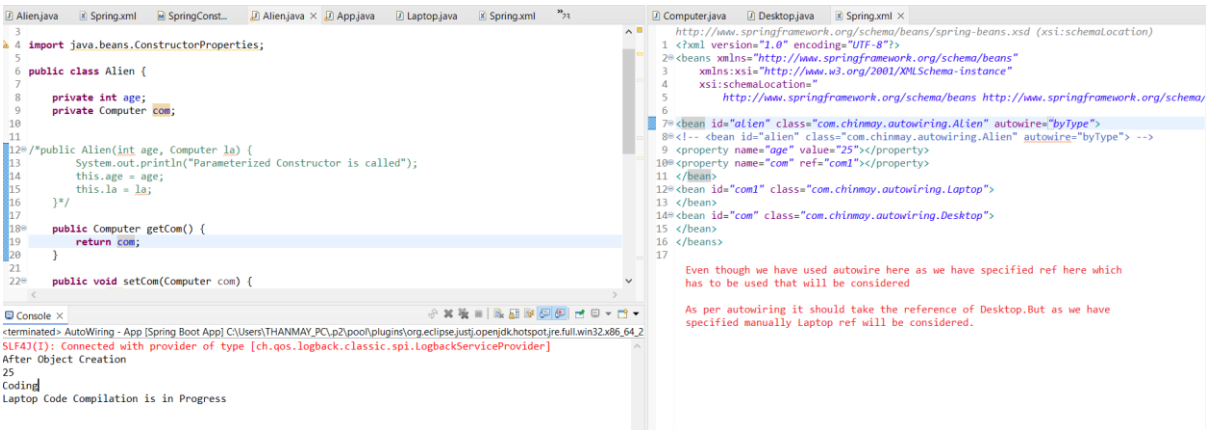
```
import java.beans.ConstructorProperties;

public class Alien {

    private int age;
    private Computer com;


/*public Alien(int age, Computer la) {
        System.out.println("Parameterized Constructor is called");
        this.age = age;
        this.la = la;
    }*/

    public Computer getCom() {
        return com;
    }

    public void setCom(Computer com) {
```

```
Console
<terminated> AutoWiring - App [Spring Boot App] C:\Users\THANMAY_PC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
SLF4J(I): Connected with provider of type [ch.qos.logback.classic.spi.LogbackServiceProvider]
After Object Creation
25
Coding
Laptop Code Compilation is in Progress
```

```
http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
<?xml version="1.0" encoding="UTF-8"?>
<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/
<bean id="alien" class="com.chinmay.autowiring.Alien" autowire="byName">
<!-- <bean id="alien" class="com.chinmay.autowiring.Alien" autowire="byType"> -->
<property name="age" value="25"></property>
<property name="com" ref="com1"></property>
</bean>
<bean id="com1" class="com.chinmay.autowiring.Laptop">
</bean>
<bean id="com" class="com.chinmay.autowiring.Desktop">
</bean>
</beans>
```

Even though we have used autowire here as we have specified ref here which has to be used that will be considered

As per autowiring it should take the reference of Desktop.But as we have specified manually Laptop ref will be considerd



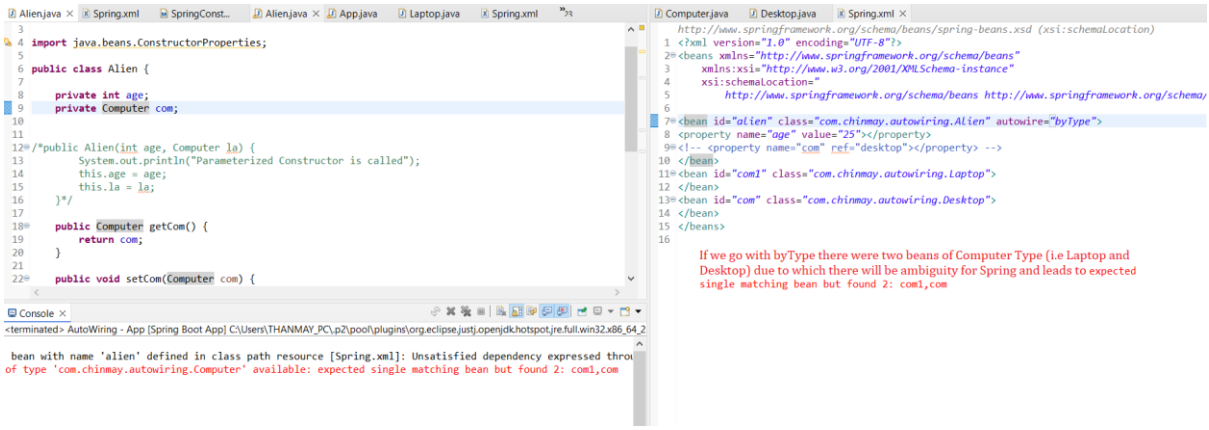
```
import java.beans.ConstructorProperties;

public class Alien {

    private int age;
    private Computer com;


/*public Alien(int age, Computer la) {
        System.out.println("Parameterized Constructor is called");
        this.age = age;
        this.la = la;
    }*/

    public Computer getCom() {
        return com;
    }

    public void setCom(Computer com) {
```

```
Console
<terminated> AutoWiring - App [Spring Boot App] C:\Users\THANMAY_PC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
SLF4J(I): Connected with provider of type [ch.qos.logback.classic.spi.LogbackServiceProvider]
After Object Creation
25
Coding
Laptop Code Compilation is in Progress
```

```
http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
<?xml version="1.0" encoding="UTF-8"?>
<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/
<bean id="alien" class="com.chinmay.autowiring.Alien" autowire="byType">
<!-- <bean id="alien" class="com.chinmay.autowiring.Alien" autowire="byType"> -->
<property name="age" value="25"></property>
<property name="com" ref="com1"></property>
</bean>
<bean id="com1" class="com.chinmay.autowiring.Laptop">
</bean>
<bean id="com" class="com.chinmay.autowiring.Desktop">
</bean>
</beans>
```

Even though we have used autowire here as we have specified ref here which has to be used that will be considered

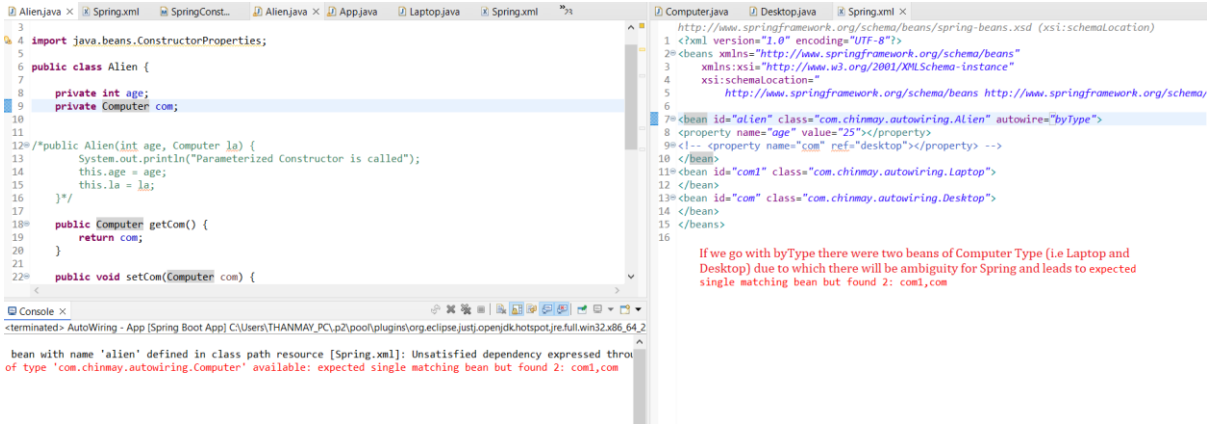
As per autowiring it should take the reference of Desktop.But as we have specified manually Laptop ref will be considered.



byType—Here there will be multiple beans of sametype so it will result in Exception.

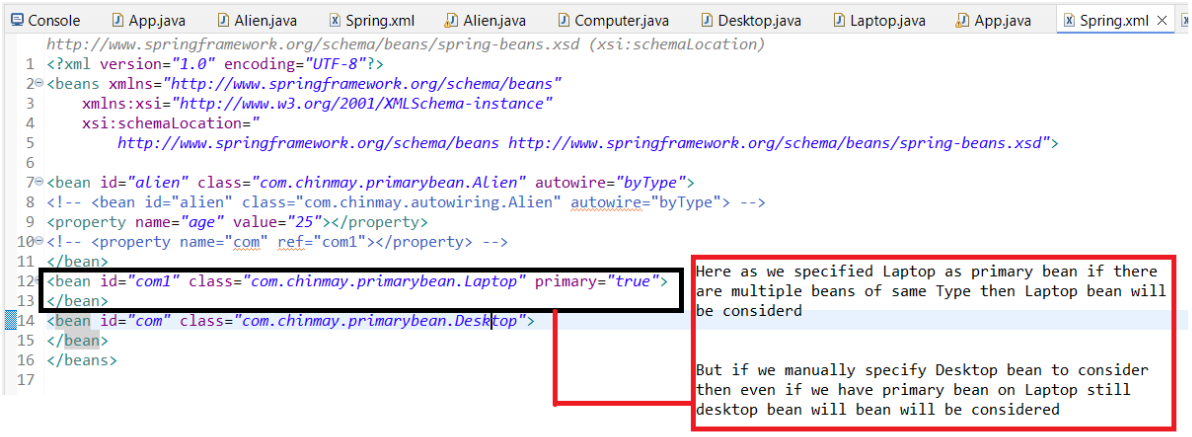
148.Primary Bean :

When we are tyring to get the Bean Details and there were multiple bean so same type then Spring will be not in position to decide which bean to consider



To resolve this we can make one bean as primary bean. So that bean which is specified as binary will be invoked when there is confusion.

Ex: There were multiple databases which will be active but we will be specifying which one will be the primitive.



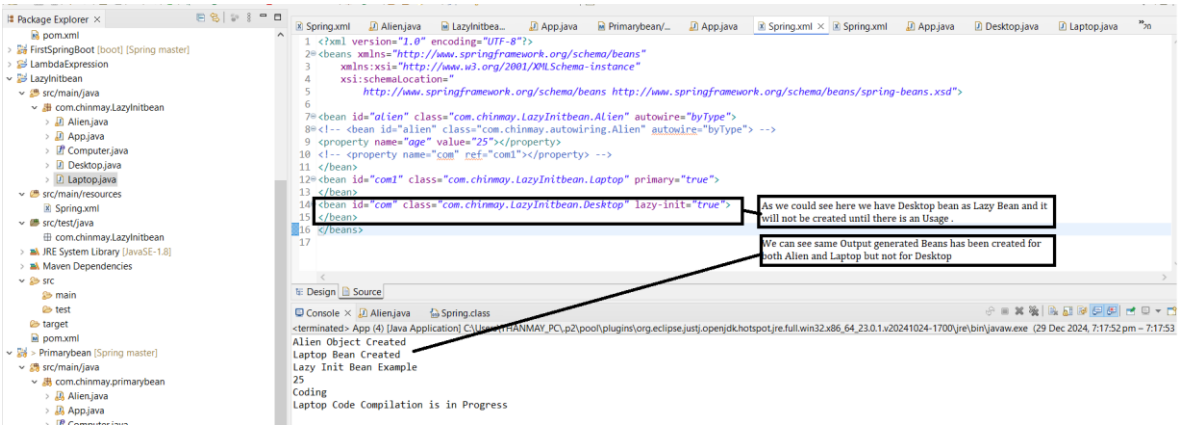
149:Lazy Init Bean

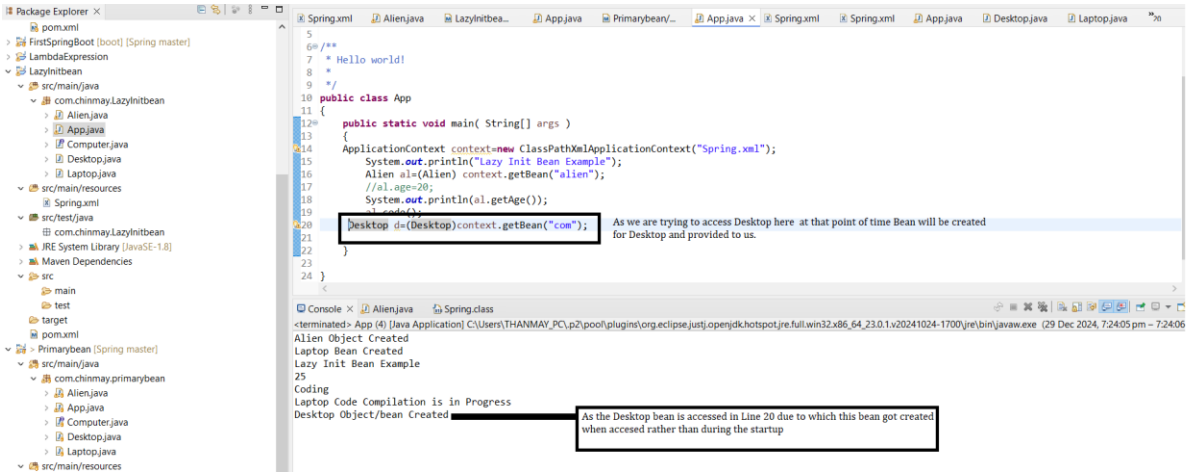
Lazy initialization of the Bean.

By default all the beans will be created during the start of the Application.

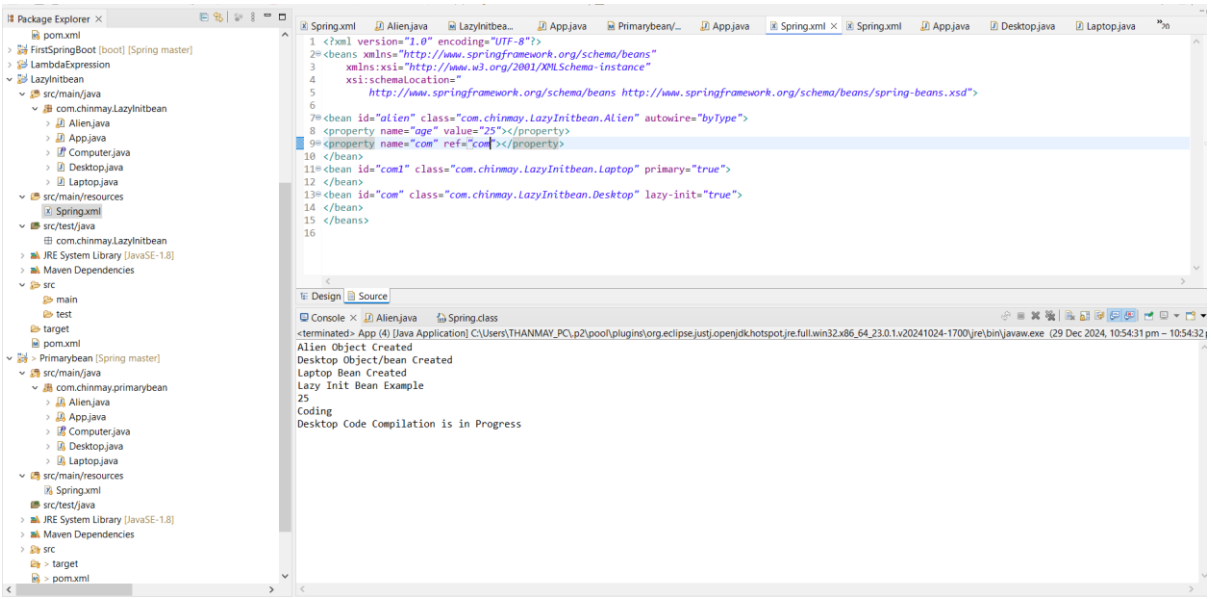
Consider a case where we want the beans to be created only if there is any usage. In this case we can make the as lazy-init bean so that bean will not be created until there is usage.

It will be useful when there are more number of beans that has to be created but all of them will not be required during the startup, Such beans we can make them as lazy-init beans so they will not be created until there is usage.





Consider a scenario where there is some other bean which is dependent on lazy-int bean in that case lazy-init will get during the startup of your application.



As we could see here even though we have specified Desktop as lazy-init bean as the Alien bean is dependent on Desktop bean it will get created.

150:GetBean By Type:

```
        Alien al=(Alien) context.getBean("alien");
        //al.age=20;
        System.out.println(al.getAge());
        al.code();
    Desktop d=(Desktop)context.getBean("com");

}
```

Here in the above we could see we are getting bean y using its name .Due to which we will get Object Type and we are converting to required type we need.

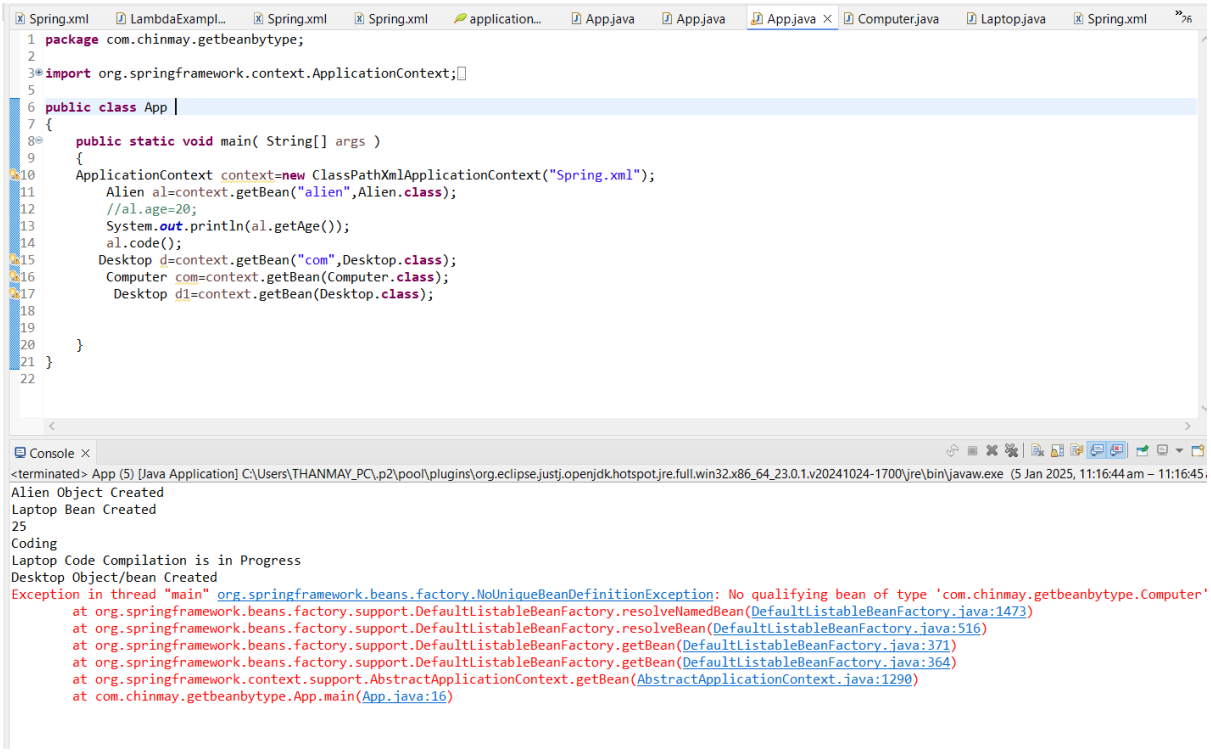
```java
public class App
{
    public static void main( String[] args )
    {
    ApplicationContext context=new ClassPathXmlApplicationContext("Spring.xml");
        Alien al=context.getBean("alien",Alien.class);
        //al.age=20;
        System.out.println(al.getAge());
        al.code();
     Desktop d=context.getBean("com",Desktop.class);

    }
}
```

```java
//  Desktop d=context.getBean("com",Desktop.class);
    Desktop d=context.getBean(Desktop.class);
```

We can specify like this with the class so we doesn't need type casting as well.

Consider a case where we have computer interface and both Laptop and Desktop provides the implementation for the Computer.

Then if we try to fetch the implementation for Computer it will lead to lead to below Exception .As there were multiple beans of Same Type .



```java
package com.chinmay.getbeanbytype;

import org.springframework.context.ApplicationContext;

public class App
{
    public static void main( String[] args )
    {
    ApplicationContext context=new ClassPathXmlApplicationContext("Spring.xml");
        Alien al=context.getBean("alien",Alien.class);
        //al.age=20;
        System.out.println(al.getAge());
        al.code();
       Desktop d=context.getBean("com",Desktop.class);
        Computer com=context.getBean(Computer.class);
         Desktop d1=context.getBean(Desktop.class);


    }
}
```

```
Alien Object Created
Laptop Bean Created
25
Coding
Laptop Code Compilation is in Progress
Desktop Object/bean Created
Exception in thread "main" org.springframework.beans.factory.NoUniqueBeanDefinitionException: No qualifying bean of type 'com.chinmay.getbeanbytype.Computer'
        at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveNamedBean(DefaultListableBeanFactory.java:1473)
        at org.springframework.beans.factory.support.DefaultListableBeanFactory.resolveBean(DefaultListableBeanFactory.java:516)
        at org.springframework.beans.factory.support.DefaultListableBeanFactory.getBean(DefaultListableBeanFactory.java:371)
        at org.springframework.beans.factory.support.DefaultListableBeanFactory.getBean(DefaultListableBeanFactory.java:364)
        at org.springframework.context.support.AbstractApplicationContext.getBean(AbstractApplicationContext.java:1290)
        at com.chinmay.getbeanbytype.App.main(App.java:16)
```

```
Spring.xml  LambdaExampl...  Spring.xml  Spring.xml  application...  App.java  App.java  App.java  Computer.java  Laptop.java  Spring.xml
   http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <beans xmlns="http://www.springframework.org/schema/beans"
 3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 4     xsi:schemaLocation="
 5         http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
 6 <bean id="alien" class="com.chinmay.getbeanbytype.Alien" autowire="byType">
 7 <property name="age" value="25"></property>
 8 <property name="com" ref="com1"></property>
 9 </bean>
10 <bean id="com1" class="com.chinmay.getbeanbytype.Laptop">
11 </bean>
12 <bean id="com" class="com.chinmay.getbeanbytype.Desktop" lazy-init="true">
13 </bean>
14 </beans>
15
```

Pom.xml

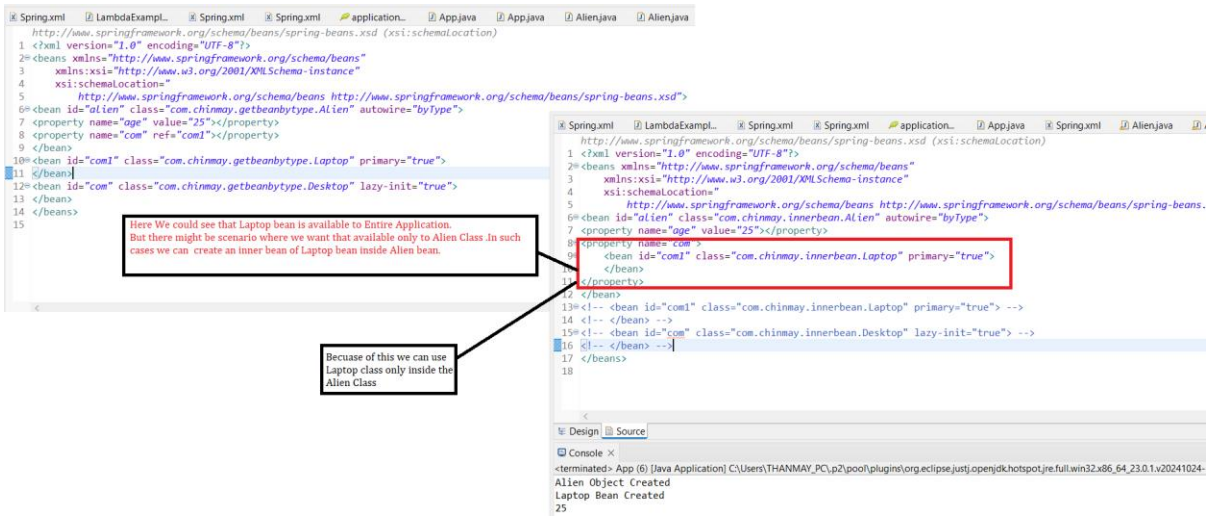
We need to made one of the bean as primary to resolve the above issue.

Resolution:

```
Spring.xml  LambdaExampl...  Spring.xml  Spring.xml  application...  App.java  App.java  Alien.java  Alien.java
   http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
 1 <?xml version="1.0" encoding="UTF-8"?>
 2 <beans xmlns="http://www.springframework.org/schema/beans"
 3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 4     xsi:schemaLocation="
 5         http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
 6 <bean id="alien" class="com.chinmay.getbeanbytype.Alien" autowire="byType">
 7 <property name="age" value="25"></property>
 8 <property name="com" ref="com1"></property>
 9 </bean>
10 <bean id="com1" class="com.chinmay.getbeanbytype.Laptop" primary="true">
11 </bean>
12 <bean id="com" class="com.chinmay.getbeanbytype.Desktop" lazy-init="true">
13 </bean>
14 </beans>
15
```

Here we had made Laptop bean as Primary.So there will no be issue .Even if we have multiple beans of Same Type the One with primary will be considered.

151.Inner Bean


Spring.xml
LambdaExampl...
Spring.xml
Spring.xml
application...
App.java
App.java
Alien.java
Alien.java
http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
<?xml version="1.0" encoding="UTF-8"?>
<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.xsd">
<bean id="alien" class="com.chinmay.getbeanbytype.Alien" autowire="byType">
<property name="age" value="25"></property>
<property name="com" ref="com1"></property>
</bean>
<bean id="com1" class="com.chinmay.getbeanbytype.Laptop" primary="true">
</bean>
<bean id="com" class="com.chinmay.getbeanbytype.Desktop" lazy-init="true">
</bean>
</beans>
Here We could see that Laptop bean is available to Entire Application.
But there might be scenario where we want that available only to Alien Class .In such cases we can create an inner bean of Laptop bean inside Alien bean.
Becuase of this we can use Laptop class only inside the Alien Class
Spring.xml
LambdaExampl...
Spring.xml
Spring.xml
application...
App.java
Spring.xml
Alien.java
http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
<?xml version="1.0" encoding="UTF-8"?>
<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.
<bean id="alien" class="com.chinmay.innerbean.Alien" autowire="byType">
<property name="age" value="25"></property>
<property name="com">
<bean id="com1" class="com.chinmay.innerbean.Laptop" primary="true">
</bean>
</property>
</bean>
<!-- <bean id="com1" class="com.chinmay.innerbean.Laptop" primary="true"> -->
<!-- </bean> -->
<!-- <bean id="com" class="com.chinmay.innerbean.Desktop" lazy-init="true"> -->
<!-- </bean> -->
</beans>
Design
Source
Console
<terminated> App (6) [Java Application] C:\Users\THANMAY_PC\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_23.0.1.v20241024-
Alien Object Created
Laptop Bean Created
25