

```
/*
Pro Spring
By Rob Harrop
Jan Machacek
ISBN: 1-59059-461-4
Publisher: Apress
*/
```

```
////////////////////////////////////
//File: initInterface.xml
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-
beans.dtd">
<beans>
  <bean id="simpleBean1" class="SimpleBeanWithInterface" init-method="myInit">
    <property name="name">
      <value>Rob Harrop</value>
    </property>
    <property name="age">
      <value>100</value>
    </property>
  </bean>
  <bean id="simpleBean2" class="SimpleBeanWithInterface">
    <property name="age">
      <value>100</value>
    </property>
  </bean>
  <bean id="simpleBean3" class="SimpleBeanWithInterface">
    <property name="name">
      <value>Rob Harrop</value>
    </property>
  </bean>
</beans>
```

```
////////////////////////////////////
import org.springframework.beans.factory.BeanCreationException;
import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.InitializingBean;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.FileSystemResource;

public class SimpleBeanWithInterface implements InitializingBean {

  private static final String DEFAULT_NAME = "Luke Skywalker";

  private String name = null;

  private int age = Integer.MIN_VALUE;

  public void setName(String name) {
    this.name = name;
  }

  public void setAge(int age) {
```

```

    this.age = age;
}

public void myInit() {
    System.out.println("My Init");
}

public void afterPropertiesSet() throws Exception {
    System.out.println("Initializing bean");

    if (name == null) {
        System.out.println("Using default name");
        name = DEFAULT_NAME;
    }

    if (age == Integer.MIN_VALUE) {
        throw new IllegalArgumentException(
            "You must set the age property of any beans of type " + SimpleBean.class);
    }
}

public String toString() {
    return "Name: " + name + "\nAge: " + age;
}

public static void main(String[] args) {
    BeanFactory factory = new XmlBeanFactory(new FileSystemResource(
        "build/initInterface.xml"));

    SimpleBeanWithInterface simpleBean1 = getBean("simpleBean1", factory);
    SimpleBeanWithInterface simpleBean2 = getBean("simpleBean2", factory);
    SimpleBeanWithInterface simpleBean3 = getBean("simpleBean3", factory);
}

private static SimpleBeanWithInterface getBean(String beanName,
    BeanFactory factory) {
    try {
        SimpleBeanWithInterface bean = (SimpleBeanWithInterface) factory.getBean(beanName);
        System.out.println(bean);
        return bean;
    } catch (BeanCreationException ex) {
        System.out.println("An error occured in bean configuration: "
            + ex.getMessage());
        return null;
    }
}
}

```

```

////////////////////////////////////
import org.springframework.beans.factory.BeanCreationException;
import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.XmlBeanFactory;
import org.springframework.core.io.FileSystemResource;

```

```

public class SimpleBean {

```

```

private static final String DEFAULT_NAME = "Luke Skywalker";

private String name = null;

private int age = Integer.MIN_VALUE;

public void setName(String name) {
    this.name = name;
}

public void setAge(int age) {
    this.age = age;
}

public void init() {
    System.out.println("Initializing bean");

    if (name == null) {
        System.out.println("Using default name");
        name = DEFAULT_NAME;
    }

    if (age == Integer.MIN_VALUE) {
        throw new IllegalArgumentException(
            "You must set the age property of any beans of type " + SimpleBean.class);
    }
}

public String toString() {
    return "Name: " + name + "\nAge: " + age;
}

public static void main(String[] args) {
    BeanFactory factory = new XmlBeanFactory(new FileSystemResource(
        "build/initMethod.xml"));

    SimpleBean simpleBean1 = getBean("simpleBean1", factory);
    SimpleBean simpleBean2 = getBean("simpleBean2", factory);
    SimpleBean simpleBean3 = getBean("simpleBean3", factory);
}

private static SimpleBean getBean(String beanName, BeanFactory factory) {
    try {
        SimpleBean bean =(SimpleBean) factory.getBean(beanName);
        System.out.println(bean);
        return bean;
    } catch (BeanCreationException ex) {
        System.out.println("An error occured in bean configuration: "
            + ex.getMessage());
        return null;
    }
}
}

```