Spring Framework 3.1 Tutorial

i 68 00125 012134 61209 79212 13684656 25464 51898 072003

Table of Contents

Spring Framework Architecture

This section describes the basic architecture of the framewr iism(illegible)

Step 4 - Setup Spring Framework Libraries

Spring IoC Containers

T

4	Create Beans configuration file *Beans.xml* under the **src** folder.

```
<?xml version=C2 RG encoding=CUTF-8G?>
```

Second method of injecting dependency is through **Setter Methods**

Here is the content of **TextEditor.java** file:

```
package com.tutorialspoint;

public class TextEditor {
   private SpellChecker spellChecker;

   // a setter method to inject the dependency.
   public void setSpellChecker(SpellChecker spellChecker) {
      System.out.println("Inside setSpellChecker." );     .
      this
```

```
package
```

Limitations with autowiring

```
    }
```

70 BF2 8.04 Tf1 1 0 0 1 277.01 1555275 T

Following is the configuration file

```
   public void spellCheck(){
```

```
      this.name = name;
   }
   public String getName() { }
```

```
}
```

e, the foo bean receives a reference to bar via constructor injection. Now let us see one king example:

ample

us have working Eclipse IDE in place and follow the following steps to create a Spring lication:

p	Description

```
        this.spellChecker W>> BDC BT
```

```
   @Bean
   @Scope("prototype")
   public Foo foo() {
      return new Foo();
   }
}
```

```
package com.tutorialspoint;
```

```
</beans>
```

Target object	

```
</aop:config>
```

```
    xsi:schemaLocation="http://www.springframework.org/schema/beans
```

Let me explain that above defined <aop:pointcut> selects all t

Spring JDBC Framework

While working with database using plain old JDBC, it becomes cumbersome to write

```
   */
```

```
   private Integer id;

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

```
   public void
```

```
      (StudentJDBCTemplate)context.getBean("studentJDBCTemplate"
```

6	**TransactionDefinition.PROPAGATION_REQUIRES_NEW** Create a new transaction, suspending the current transaction if one exists.

```
   /**
    * This is the method to be used to initialize
    * database resources ie. connection.
    */
```

```
package com
```

```
public class StudentMarksMapper
```

2.

Here **${message}** is the attribute which we have setup inside the Controller. You can have

```
<html>
<head>
<title>Hello World</title>
</head>
<body>
   <h2>${message}</h2>
</body>
</html>
```

```
   private String name;
   private Integer id;

   public void setAge(Integer age) {
      this.age = age;
```

```
<%@taglib
```

```
</html>
```

Finally, following is the list of Spring and other libraries to be included in your web application. You simply drag these files and drop them in **WebContent/WEB-INF/lib**