First Simple Spring Application-

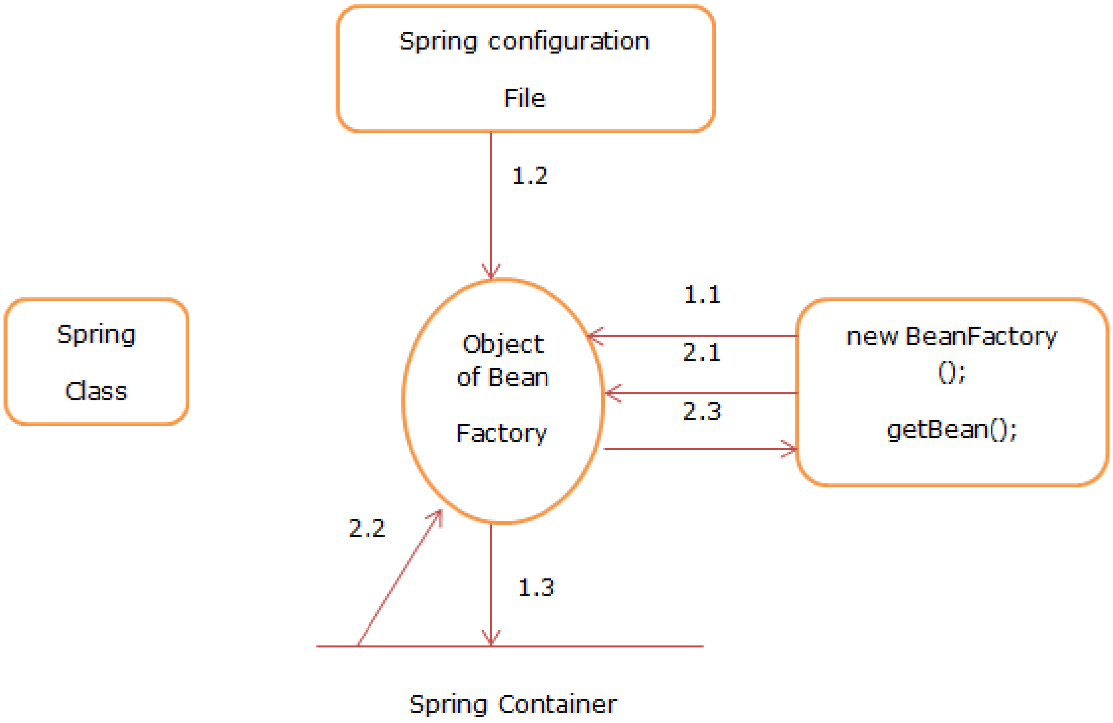
The following resources are required to build the first spring application.

Spring bean class (Java class)

Spring configuration files (.xml)

Client application (Java class with main method)-

Internal working of Spring-



1.1 BeanFactory object is created.

1.2 Spring Configuration file is injected into bean factory object.

1.3 Spring bean (POJO) object created and stored into Spring/IOC container.

2.1 getBean () method is invoked.

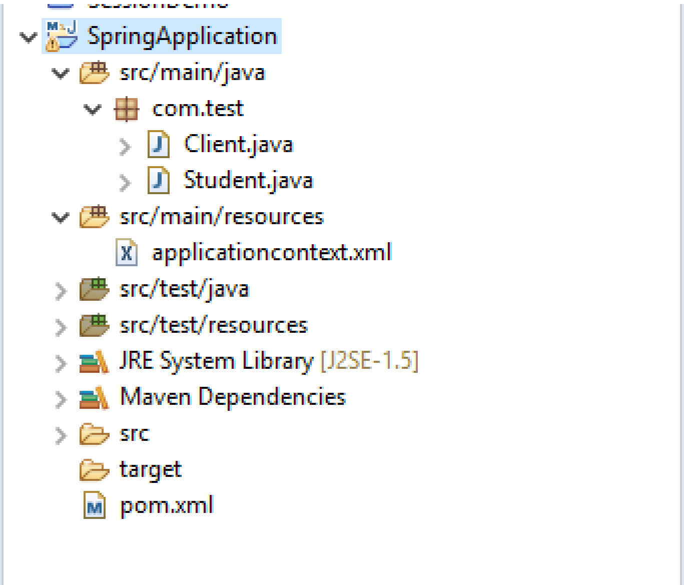
2.2 get the spring bean object (Student class object)

2.3 Bean object provided to user program.

Note –

1. If we use ApplicationContext then it will create the object of ApplicationContext not BeanFactory.
2. We can mention named as spring.xml file or applicalicationcontext.xml.

Folder structure of Spring Application.



```
SpringApplication
  src/main/java
    com.test
      Client.java
      Student.java
  src/main/resources
    applicationcontext.xml
  src/test/java
  src/test/resources
  JRE System Library [J2SE-1.5]
  Maven Dependencies
  src
  target
  pom.xml
```

Student.java

```
package com.test;

public class Student {

	private String firstname;
	private String age;
	private String city;

	public String getFirstname() {
		return firstname;
	}

	public void setFirstname(String firstname) {
        this.firstname = firstname;
    }

    public String getAge() {
        return age;
    }

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

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    @Override
    public String toString() {
        return "Student [firstname=" + firstname + ", age="
+ age + ", city=" + city + "]";
    }
}

}
```

Client.java

```
package com.test;

import org.springframework.beans.factory.BeanFactory;
import org.springframework.beans.factory.xml.~~XmlBeanFactory~~;
import org.springframework.core.io.ClassPathResource;
import org.springframework.core.io.Resource;

public class Client {

    public static void main(String[] args) {
```

```
		ApplicationContext context=new
ClassPathXmlApplicationContext("spring.xml");
		Student student=(Student)context.getBean("s");
		System.out.println("Student data>>"+student);  }
}
```

applicationContext.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/schem
a/beans
    http://www.springframework.org/schema/beans/spring-
beans-2.5.xsd">

	<bean id="s" class="com.test.Student">
		<property name="firstname" value="ram"></property>
		<property name="age" value="25"></property>
		<property name="city" value="pune"></property>
	</bean>
</beans>
```

Pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
	xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
	xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
	<modelVersion>4.0.0</modelVersion>
	<groupId>spring</groupId>
	<artifactId>SpringApplication</artifactId>
	<version>0.0.1-SNAPSHOT</version>
	<dependencies>
		<dependency>
			<groupId>org.springframework</groupId>
			<artifactId>spring-core</artifactId>
			<version>${spring.version}</version>
		</dependency>
```

```xml
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>${spring.version}</version>
        </dependency>

    </dependencies>
    <properties>
        <spring.version>3.2.3.RELEASE</spring.version>
    </properties>
</project>
```

Output-

```
Student data>>Student [firstname=ram, age=25, city=pune]
```