@Required Annotation-

It is used to set the particular property of class called as @Required annotation.

This is the method level annotation.

Why?

If there are 5 setter dependency in class and I want to make only one setter base injection mandatory at time. Simple dependency checking not worked, to overcome above problem we should go for @Required annotation.

Program for using @Required annotation.

Drivers.java

```
package com.demo;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import
org.springframework.beans.factory.annotation.Required;
public class Drivers {

    private String driver;
    private String url;
    private String username;
    private String password;

    @Required
    public void setDriver(String driver) {
        this.driver = driver;
    }

    @Required
    public void setUrl(String url) {
        this.url = url;
    }

    public void setUsername(String username) {
        this.username = username;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public void printCon() throws ClassNotFoundException,
SQLException {

        Class.forName(driver);
        Connection
con=DriverManager.getConnection(url,username,password);
        System.out.println(con);
    }
}
```

TestMain.java

```
package com.demo;

import java.sql.SQLException;

import org.springframework.context.ApplicationContext;
import
org.springframework.context.support.ClassPathXmlApplicationC
ontext;

public class TestMain {

    public static void main(String[] args) throws
ClassNotFoundException, SQLException {

        ApplicationContext context = new
ClassPathXmlApplicationContext("spring.xml");
        Drivers drivers = (Drivers) context.getBean("d");
        drivers.printCon();
    }
}
```

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>com.spring.test</groupId>
    <artifactId>PropertyFileDemo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <dependencies>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-core</artifactId>
            <version>${spring.version}</version>
        </dependency>

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

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.22</version>
        </dependency>

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

</project>
```

Spring.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"
   xmlns:context =
"http://www.springframework.org/schema/context"
   xsi:schemaLocation =
"http://www.springframework.org/schema/beans
   http://www.springframework.org/schema/beans/spring-beans-
3.0.xsd
   http://www.springframework.org/schema/context
   http://www.springframework.org/schema/context/spring-
context-3.0.xsd">

   <context:annotation-config/>

     <bean id="d" class="com.demo.Drivers">
          <property name="url"
value="jdbc:mysql://localhost:3306/test"></property>
          <property name="driver"
value="com.mysql.jdbc.Driver"></property>
          <property name="username" value="root"></property>
          <property name="password" value="root"></property>
     </bean>
</beans>
```

Output-

```
com.mysql.cj.jdbc.ConnectionImpl@192c3f1e
```