

Spring Framework

Date - 11/06/2023

App.Java

```
package com.springcore;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {
    public static void main(String[] args) {
        System.out.println("Hello World!");
        // Create an application context and load the configuration file
        ApplicationContext context = new ClassPathXmlApplicationContext("config.xml");
        // Retrieve the Student bean from the application context
        Student student1 = (Student) context.getBean("Student1");
        // Print the details of the Student bean
        System.out.println(student1);
    }
}
// -----
```

Student.Java

```
package com.springcore;
public class Student {
    private int studentId;
    private String studentName;
    private String studentAddress;
    public int getStudentId() {
        return studentId;
    }
    public void setStudentId(int studentId) {
        this.studentId = studentId;
    }
    public String getStudentName() {
        return studentName;
    }
    public void setStudentName(String studentName) {
        this.studentName = studentName;
    }
    public String getStudentAddress() {
        return studentAddress;
    }
    public void setStudentAddress(String studentAddress) {
        this.studentAddress = studentAddress;
    }
    public Student(int studentId, String studentName, String studentAddress) {
        super();
        this.studentId = studentId;
        this.studentName = studentName;
        this.studentAddress = studentAddress;
    }
    public Student() {
        super();
        // TODO Auto-generated constructor stub
    }
    @Override
    public String toString() {
        return "Student [studentId=" + studentId + ", studentName=" + studentName + ", studentAddress=" + studentAddress + "]\n";
    }
}
// -----
```

Config.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           https://www.springframework.org/schema/beans/spring-beans.xsd">
    <!-- This is our bean for Student-->
    <bean class="com.springcore.Student" name="Student1">
        <property name="studentId">
            <value>22344</value>
        </property>
        <property name="studentName">
            <value>Ifikar</value>
        </property>
        <property name="studentAddress">
            <value>Guwahati</value>
        </property>
    </bean>
</beans>
// -----
```

The code represents a simple Spring application that demonstrates the usage of the Spring Framework for dependency injection and object creation. It consists of three files: **App.java**, **Student.java**, and **config.xml**.

The App.java file serves as the main class for the application. It begins by printing "Hello World!" to the console. Then, it creates an ApplicationContext object using ClassPathXmlApplicationContext, which loads the Spring configuration file config.xml. This configuration file contains the bean definitions for the application. The application context is responsible for managing the beans and their dependencies.

Next, the code retrieves a bean named "Student1" from the application context using the `getBean()` method. This bean represents a Student object. The retrieved Student object is then printed to the console using its `toString()` method.

The Student.java file defines the Student class, which represents a student object. It contains private variables for the student's ID, name, and address, along with getter and setter methods to access and modify these properties. The class also provides a constructor that takes all three properties as arguments and sets them. Additionally, a default constructor is defined. The `toString()` method is overridden to provide a string representation of the Student object.

The config.xml file is the Spring configuration file written in XML format. It defines the beans for the application. In this case, it defines a bean named "Student1" of type `com.springcore.Student`. The bean has **three properties (studentId, studentName, and studentAddress)**, and their values are set using the `<property>` elements in the XML file.

Overall, this code demonstrates how to configure and use Spring beans in an application by defining them in a configuration file and retrieving them from the application context.