**Assignment-2: Understanding Spring Framework 6 - Autowiring and Bean Configuration**

**Objective:** The objective of this assignment is to deepen your understanding of the Spring Framework version 6, specifically focusing on the autowiring mechanism and related concepts such as autowire types, bean scopes, and lazy initialization.

**Instructions:**

1. Create a new Spring project using the XML configuration approach. You can use any Java IDE or text editor of your choice.
2. Configure the necessary dependencies in your project. Make sure to include the Spring Framework 6 dependencies in your project's build file (e.g., pom.xml for Maven or build.gradle for Gradle). You can refer to the official Spring documentation for the latest version and dependency details.
3. Create a new package in your project to organize your classes. For example, you can create a package called **com.example.springassignment**.
4. Implement the following classes and XML configuration files to demonstrate the autowiring and related concepts:

**Autowire Types:**

**Step 1: ByName** Create two classes, **Employee** and **Company**, inside the **com.example.springassignment** package.

* **Employee** class:

package com.example.springassignment;

public class Employee {

private String name;

// Implement getters and setters

}

* **Company** class:

package com.example.springassignment;

public class Company {

private Employee employee;

// Implement getters and setters

}

Create an XML configuration file called **autowire-byname.xml** and place it in the project's resources directory. Configure the autowiring by name for the **Company** bean using the **<bean>** and **<property>** elements in the XML file.

**Step 2: ByType** Create two classes, **Product** and **Order**, inside the **com.example.springassignment** package.

* **Product** class:

package com.example.springassignment;

public class Product {

private String name;

// Implement getters and setters

}

* **Order** class:

package com.example.springassignment;

public class Order {

private Product product;

// Implement getters and setters

}

Create an XML configuration file called **autowire-bytype.xml** and place it in the project's resources directory. Configure the autowiring by type for the **Order** bean using the **<bean>** and **<property>** elements in the XML file.

**Step 3: Constructor** Create two classes, **Customer** and **Invoice**, inside the **com.example.springassignment** package.

* **Customer** class:

package com.example.springassignment;

public class Customer {

private String name;

// Implement getters and setters

}

* **Invoice** class:

package com.example.springassignment;

public class Invoice {

private Customer customer;

public Invoice(Customer customer) {

this.customer = customer;

}

// Implement getters and setters

}

Create an XML configuration file called **autowire-constructor.xml** and place it in the project's resources directory. Configure the constructor autowiring for the **Invoice** bean using the **<bean>** and **<constructor-arg>** elements in the XML file.

**Bean Scopes:**

**Step 4: Singleton** Create a class called **User** inside the **com.example.springassignment** package.

* **User** class:

package com.example.springassignment;

public class User {

private String username;

// Implement getters and setters

}

Create an XML configuration file called **bean-scope-singleton.xml** and place it in the project's resources directory. Configure the singleton scope for the **User** bean using the **scope** attribute of the **<bean>** element in the XML file. Create two **User** beans in the XML file with different usernames.

**Step 5: Prototype** Create an XML configuration file called **bean-scope-prototype.xml** and place it in the project's resources directory. Configure the prototype scope for the **User** bean using the **scope** attribute of the **<bean>** element in the XML file. Create two **User** beans in the XML file with different usernames.

**Lazy Initialization:**

**Step 6: Lazy Initialization** Create a class called **ReportGenerator** inside the **com.example.springassignment** package.

* **ReportGenerator** class:

package com.example.springassignment;

public class ReportGenerator {

public ReportGenerator() {

System.out.println("ReportGenerator instance created.");

}

public void generateReport() {

System.out.println("Generating report...");

}

}

Create an XML configuration file called **lazy-init.xml** and place it in the project's resources directory. Configure the lazy initialization for the **ReportGenerator** bean using the **lazy-init** attribute of the **<bean>** element in the XML file.

**Deliverables:** Submit the following files:

1. A compressed folder containing your Spring project.
2. A document or text file explaining the purpose of each class and XML configuration file in your project.
3. Any additional instructions or comments you feel are necessary to understand your code.

**Note:** Make sure to provide a clear explanation of your code and demonstrate your understanding of the autowiring mechanism, bean scopes, and lazy initialization in the Spring Framework version 6 using XML configuration.