Universidad Nacional Autónoma de México

Dirección General de Cómputo de Tecnologías de Información y Comunicación

Diplomado

Desarrollo de Sistemas con Tecnología Java

Ejercicio Tres

“BeanFactory”

Mtro. ISC. Miguel Ángel Sánchez Hernández

Tabla de contenido

1. Creación de un Proyecto Java en IntelliJ IDEA

Vamos a crear un proyecto Java con IntelliJ IDEA, los pasos son los siguientes:

1. Una vez que esté abierto el IDE, nos vamos a File→New→Project
2. Señalamos New Project
3. Indicamos los siguientes parámetros:
    - Name: spring-core-beanfactory
    - Location: Dirección de almacenamiento del proyecto, a su elección
    - Languaje: Java
    - Build system: Maven
    - JDK: 17
    - Catalog: Internal
4. Advanced Settings
    - GroupId: dgtic.core
    - ArtifactId: spring-core-beanfactory
5. Presionamos Create

2. Cambiamos el archivo pow.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<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>dgtic.core</groupId>
    <artifactId>prueba-core</artifactId>
    <version>1</version>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.4</version>
        <relativePath/>
    </parent>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-jar-plugin</artifactId>
```

```
            <version>3.1.0</version>
            <configuration>
               <archive>
                  <manifest>
                     <addClasspath>true</addClasspath>
                     <classpathPrefix>lib/</classpathPrefix>
                     <mainClass>dgtic.core.inicio.Inicio</mainClass>
                  </manifest>
               </archive>
            </configuration>
         </plugin>
      </plugins>
   </build>
</project>
```

3. Clase Persona

Creamos la clase Persona con las siguientes características.

- Señalamos en src/main/java
- Nombre de la clase: Persona
- Paquete: dgtic.core.modelo

```
package dgtic.modelo;
package dgtic.core.modelo;
public class Persona {
		private String nombre;
		private Integer edad;
		public Persona() {

		}
		public String getNombre() {
			return nombre;
		}
		public void setNombre(String nombre) {
			this.nombre = nombre;
		}
		public Integer getEdad() {
			return edad;
		}
		public void setEdad(Integer edad) {
			this.edad = edad;
		}
}
```

4. Archivo de configuración bean-configuration.xml

Para crear el archivo xml hacemos lo siguiente:

1. Señalamos la carpeta resources
2. Click derecho, New→Resource Bundle
   - Resource Bundle base name: bean-configuration.xml
   - Use XML based properties files: se señala
3. Presionamos OK.

```
<?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/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
   <!-- bean definitions here -->
   <bean id="persona" class="dgtic.core.modelo.Persona"/>
</beans>
```

5. Clase Inicio

Para crear la clase Inicio hacemos lo siguiente:

1. Señalamos en src/main/java
2. Package: dgtic.core.inicio
3. Nombre de la clase: Inicio

```
package dgtic.core.inicio;
import dgtic.core.modelo.Persona;
import org.springframework.beans.factory.support.DefaultListableBeanFactory;
import org.springframework.beans.factory.xml.XmlBeanDefinitionReader;
import org.springframework.core.io.ClassPathResource;
import org.springframework.core.io.Resource;
public class Inicio {
		public static void main(String[] args) {
				final Resource resource = new ClassPathResource("bean-configuration.xml");
				final DefaultListableBeanFactory beanFactory = new DefaultListableBeanFactory();
				final XmlBeanDefinitionReader xmlBeanDefinitionReader = new XmlBeanDefinitionReader(beanFactory);
				xmlBeanDefinitionReader.loadBeanDefinitions(resource);
				Persona per=(Persona)beanFactory.getBean("persona");
				System.out.println(per);
				System.out.println(beanFactory.isSingleton("persona"));
				System.out.println(beanFactory.getBean("persona") instanceof Persona);
				System.out.println(beanFactory.isTypeMatch("persona", Persona.class));
				System.out.println(beanFactory.getAliases("persona").length>0);
		}
}
```

Correr la aplicación