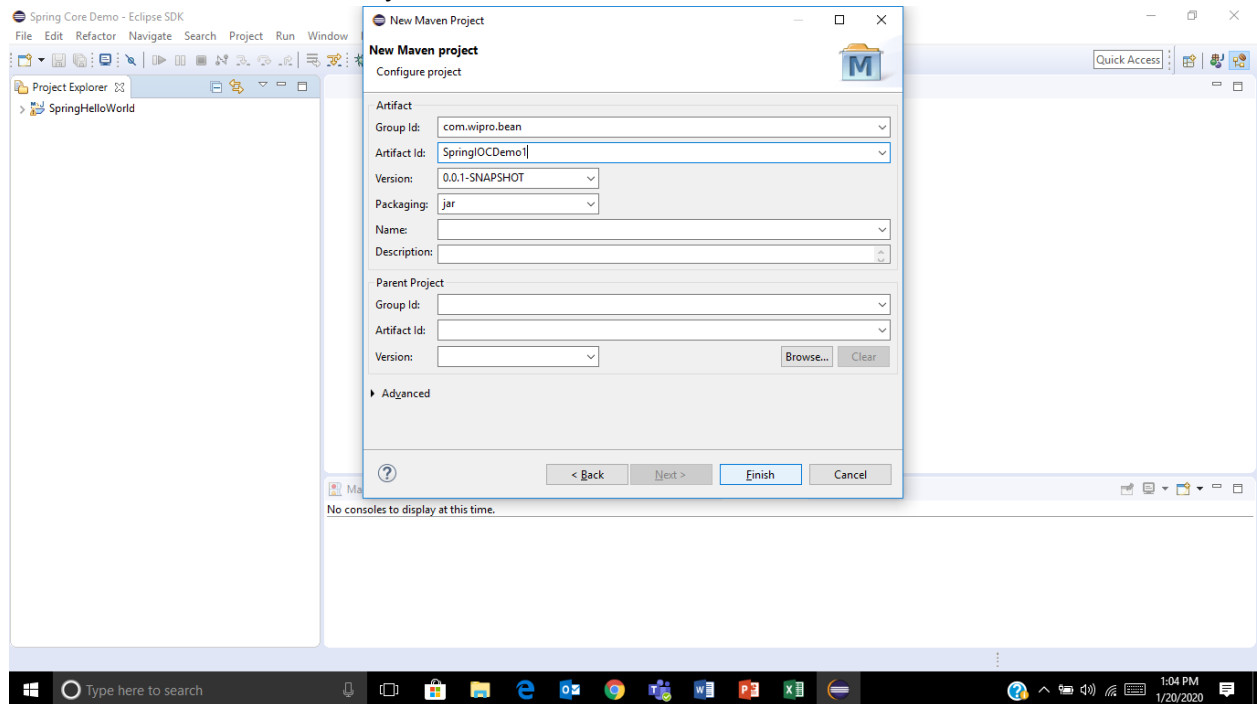


Spring IOC Demo Projects

I. Setter Based DI

a. Let us make a Maven Based Project.



b. Add the dependencies in the POM.xml file

```
<dependencies>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-core</artifactId>
        <version>5.2.1.RELEASE</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>5.2.1.RELEASE</version>
    </dependency>
</dependencies>
```

c. Under src/main/java – Create a package called com.wipro.bean

d. Under the package create 2 classes called

- i. Address.java
- ii. Employee.java

Address Class Code:

```
package com.wipro.bean;

public class Address {

    private String street;
```

```

private String city;
private String pincode;
public Address() {
    // TODO default constructor stub
}

public Address(String street, String city, String pincode) {
    this.street = street;
    this.city = city;
    this.pincode = pincode;
}

public String getStreet() {
    return street;
}
public void setStreet(String street) {
    this.street = street;
}
public String getCity() {
    return city;
}
public void setCity(String city) {
    this.city = city;
}
public String getPincode() {
    return pincode;
}
public void setPincode(String pincode) {
    this.pincode = pincode;
}
@Override
public String toString() {
    return "Address [street=" + street + ", city=" + city + ", pincode=" +
pincode + "]";
}
}

```

Employee Class Code:

```

package com.wipro.bean;

public class Employee {
    private String name;
    private String empId;
    private Address address;
    public Employee() {
        super();
        // TODO default constructor stub
    }
    public Employee(String name, String empId, Address address) {
        super();
        this.name = name;
        this.empId = empId;
    }
}

```

```

        this.address = address;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getEmpId() {
        return empId;
    }
    public void setEmpId(String empId) {
        this.empId = empId;
    }
    public Address getAddress() {
        return address;
    }
    public void setAddress(Address address) {
        this.address = address;
    }
    @Override
    public String toString() {
        return "Employee [name=" + name + ", empId=" + empId + ",
address=" + address + "]\n";
    }
}

```

- e. Create the beans.xml configuration file under the src/main/resources

```

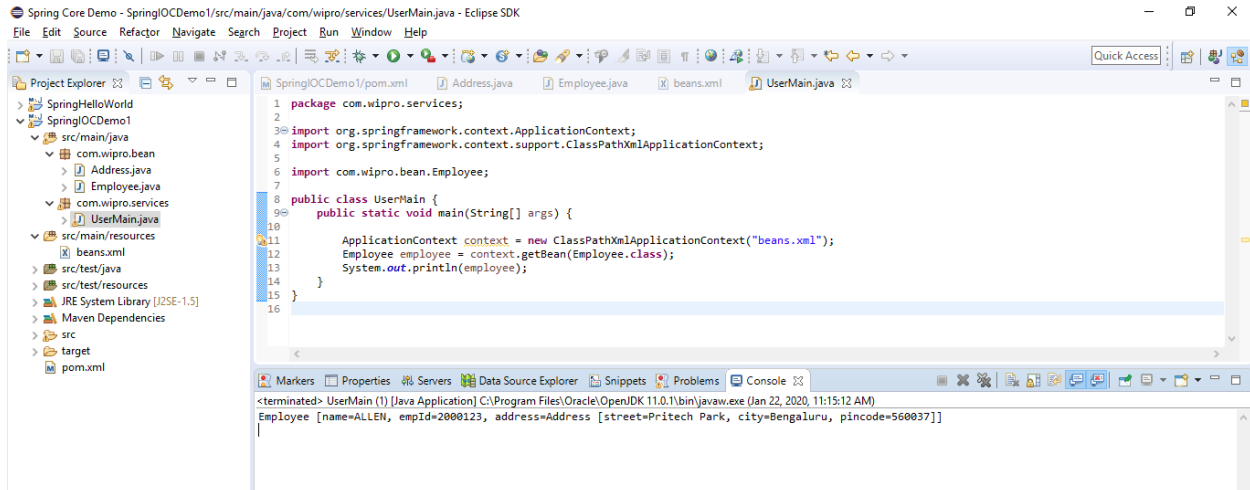
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <bean id="addressBean" class="com.wipro.bean.Address">
        <property name="street" value="Pritech Park"></property>
        <property name="city" value="Bengaluru"></property>
        <property name="pincode" value="560037"></property>
    </bean>

    <bean id="employeeBean" class="com.wipro.bean.Employee">
        <property name="name" value="ALLEN"></property>
        <property name="empId" value="2000123"></property>
        <property name="address" ref="addressBean"></property>
    </bean>
</beans>

```

- f. Create the UserMain class under src/main/java – package com.wipro.services



II. Constructor Based DI

On to the same application above Let us change the beans.xml and UserMain to see difference in Constructor based DI.

1. Create xml configuration file named beans-consdi.xml under src/main/resources

```
<?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.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd">

    <bean id="addressBean" class="com.wipro.bean.Address">
        <!-- Using Constructor for passing the values -->
        <constructor-arg index="0" type="java.lang.String" value="Pritech
Park"/></constructor-arg>
        <constructor-arg index="1" type="java.lang.String"
value="Bengaluru"/></constructor-arg>
        <constructor-arg index="2" type="java.lang.String"
value="560037"/></constructor-arg>
    </bean>

    <bean id="employeeBean" class="com.wipro.bean.Employee">
        <constructor-arg index="0" type="java.lang.String"
value="ALLEN"/></constructor-arg>
        <constructor-arg index="1" type="java.lang.String"
value="2000123"/></constructor-arg>
        <constructor-arg index="2" type="com.wipro.bean.Address"
ref="addressBean"/></constructor-arg>
    </bean>
```

</beans>

2. Create UserMainConstructorDI class to test the constructor based DI.

