

Java Spring & AWS

Sep 7, 2021 - Oct 8, 2021

Monday to Friday

9:30 AM ET - 4:30 PM ET



Agenda: INTRODUCTION TO SPRING FRAMEWORK

- Web Application using MVC
- Overview of Spring Technology
 - Challenges for Modern Applications
 - Motivation for Spring, Spring Architecture
 - The Spring Framework
- Spring Introduction
 - Managing Beans
 - Inversion of Control / IoC, Dependency Injection / DI
 - Configuration Metadata Overview, Configuring Beans (XML)

@ Copyright 2021, Summitworks Technologies Inc.

Web Application using MVC

- For creating the web application, we are using JSP for presentation logic, Bean class for representing data and DAO class for database codes.
- As we create the simple application in hibernate, we don't need to perform any extra operations in hibernate for creating web application. In such case, we are getting the value from the user using the JSP file.
- We can use MVC Pattern , we can make servlet for a controller, JSPs/JavaScript for Presentation logic , Hibernate for Business logic.

@ Copyright 2021, Summitworks Technologies Inc.



What is framework

It's a special software that is enable of developing applications based on certain architecture with the ability to generate common logics of the application.

@ Copyright 2021, Summitworks Technologies Inc.

Spring Introduction

- It was **developed by Rod Johnson in 2003**
- Can be used to develop any kind of Java JEE Application
- Its developed to simplify the development of enterprise applications in java technology
- It's a open source framework
- It provides support for hibernate , web services, struts, JSF and many other frameworks
- Spring MVC is used to develop MVC based web applications
- Spring can be used to configure declarative transaction management , remote access to your logic using RMI or web services, mailing facilities and various options in persisting your data to database.

@ Copyright 2021, Summitworks Technologies Inc.

Advantages of Spring Framework

- Predefined Templates
- Loose Coupling
- Easy to test
- Lightweight
- Fast Development

@ Copyright 2021, Summitworks Technologies Inc.

- Spring framework provides templates for JDBC, Hibernate, JPA etc. technologies. So there is no need to write too much code. It hides the basic steps of these technologies.
- Let's take the example of HibernateTemplate, you don't need to write the code for exception handling, creating connection, creating statement, committing transaction, closing connection etc. You need to write the code of executing query only. Thus, it save a lot of code.

@ Copyright 2021, Summitworks Technologies Inc.

Loose Coupling

- What is the difference between loose coupling and tight coupling?

@ Copyright 2021, Summitworks Technologies Inc.

Tight-coupling and Loose-coupling between objects

- **Tight-Coupling:-**
 1. While creating complex application in java, the logic of one class will call the logic of another class to provide same service to the clients.
 2. If one class calling another class logic then it is called collaboration.
 3. When one class is collaborating with another class then there exists tight-coupling between the two classes.
 4. If one class wants to call the logic of a second class then they first class need an object of second class it means the first class create an object of second class.

@ Copyright 2021, Summitworks Technologies Inc.



```
public class Journey {  
    Car car = new Car();  
    public void startJourney() {  
        car.move();  
    }  
}
```

```
public class Car {  
    public void move () {  
        System.out.println("Travel by Car");  
    }  
}
```

In the above code the Journey class is dependent on Car class to provide service to the end user.

It means if any change in the Car class requires the Journey class to change. For example, if the Car class move() method changes to journey() method, then you have to change the startJourney() method to call journey() instead of calling move() method.

@ Copyright 2021, Summitworks Technologies Inc.

Examples

The best example of tight coupling is **RMI** (Remote Method Invocation). (Nowadays, everywhere using web services and SOAP instead of using RMI, it has some disadvantages).

@ Copyright 2021, Summitworks Technologies Inc.

Loose Coupling:-

- Loose coupling is a design goal that seeks to reduce the inter-dependencies between components of a system with the goal of reducing the risk that changes in one component will require changes in any other component. Loose coupling is a much more generic concept intended to increase the flexibility of the system, make it more maintainable and makes the entire framework more stable.

@ Copyright 2021, Summitworks Technologies Inc.


```
public interface Vehicle {  
    void move();  
}
```

```
public class Car implements  
Vehicle {  
    @Override  
    public void move() {  
        System.out.printl  
n("Travel by Car");  
    }  
}
```

```
public class Bike implements  
Vehicle {  
    @Override  
    public void move() {  
        System.out.printl  
n("Travel by Bike");  
    }  
}
```

```
class Traveler  
{ Vehicle v;  
  public void setV(Vehicle v)  
  {  
      this.v = v;  }  
  void startJourney()  
  {  
      v.move();  }}
```

@ Copyright 2021, Summitworks Technologies Inc.

Example

- in above example, spring container will inject either Car object or Bike object into the Traveler by calling setter method, So if Car object is replaced with Bike then no changes are required in Traveler class, this means there is loose coupling between Traveler and Vehicle object. Actually setter method will be activated through **xml** file/annotations , will see this later for now we are good to go.

The examples of Loose coupling are Interface, JMS, Spring IOC(Dependency Injection, it can reduce the tight coupling).

@ Copyright 2021, Summitworks Technologies Inc.

Advantages

A loosely coupled will help you when your application need to change or grow. If you design with loosely coupled architecture, only a few parts of the application should be affected when requirements change. With too a tight coupled architecture, many parts will need to change and it will be difficult to identify exactly which parts will be affected. In short,

- It improves the testability.
- The benefit is that it's much easier to swap other pieces of code/modules/objects/components when the pieces are not dependent on one another.
- It's highly changeable. One module doesn't break other modules in unpredictable ways.

@ Copyright 2021, Summitworks Technologies Inc.

- In order to overcome tight coupling between objects, spring framework uses dependency injection mechanism with the help of POJO/POJI model and through dependency injection its possible to achieve loose coupling

@ Copyright 2021, Summitworks Technologies Inc.

Easy to test

- The Dependency Injection makes easier to test the application. The EJB or Struts application require server to run the application but Spring framework doesn't require server.

@ Copyright 2021, Summitworks Technologies Inc.

Lightweight

- Spring framework is lightweight because of its POJO implementation. The Spring Framework doesn't force the programmer to inherit any class or implement any interface. That is why it is said non-invasive.

@ Copyright 2021, Summitworks Technologies Inc.



IOC giving support of containers and dependency injection support

Container ?:

A basic spring container responsibilities :

- Containers provides Dynamic Object creation support, by reading data from configuration files
- Containers can support Dependency lookup and Dependency Injection.
- Containers can manage object lifecycles
- Containers can provide event handlers
- Containers can support internationalization

@ Copyright 2021, Summitworks Technologies Inc.

Fast Development

- The Dependency Injection feature of Spring Framework and its support to various frameworks makes the easy development of Java JEE application.

@ Copyright 2021, Summitworks Technologies Inc.

Inversion Of Control (IOC) and Dependency Injection

- These are the design patterns that are used to remove dependency from the programming code. They make the code easier to test and maintain. Let's understand this with the following code:
- These are the design patterns that are used to remove dependency from the programming code. They make the code easier to test and maintain. Let's understand this with the following code:

@ Copyright 2021, Summitworks Technologies Inc.

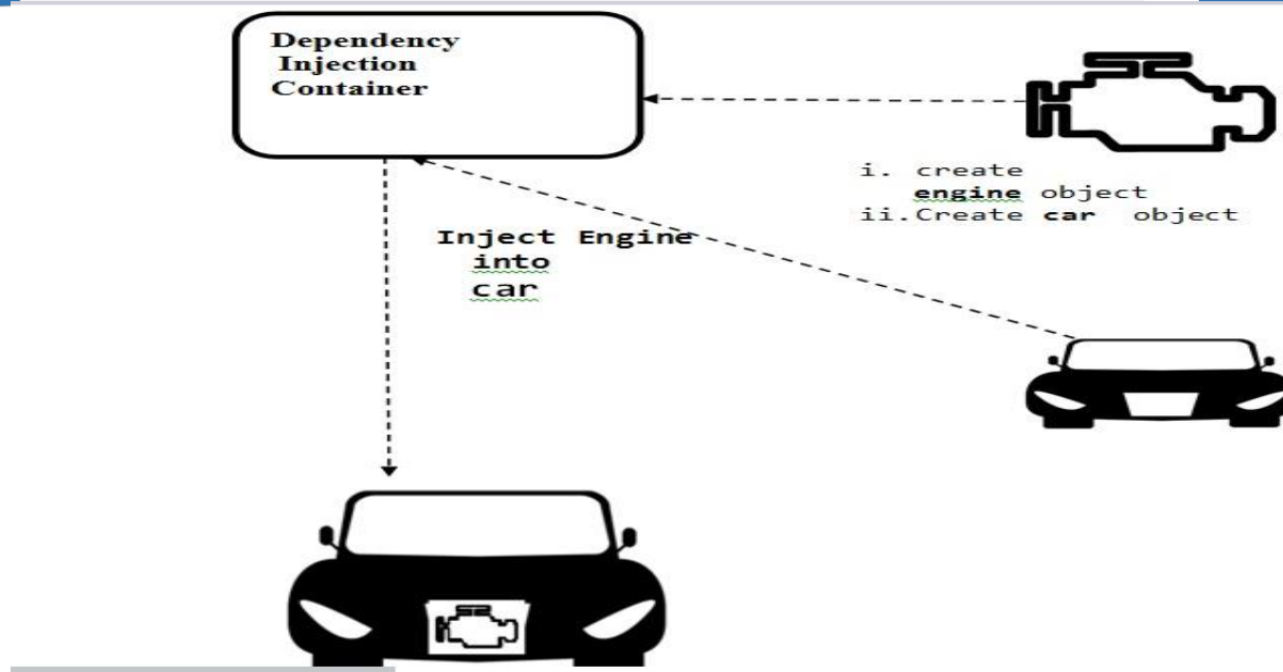
```
class Employee{
    Address address;
    Employee(){
        address=new Address();
    }
}
```

In such case, there is dependency between the Employee and Address (tight coupling). In the Inversion of Control scenario, we do this something like this:

```
class Employee{
    Address address;
    Employee(Address address){
        this.address=address;
    }
}
```

Thus, IOC makes the code loosely coupled. In such case, there is no need to modify the code if our logic is moved to new environment. In Spring framework, IOC container is responsible to inject the dependency. We provide metadata to the IOC container either by XML file or annotation.

@ Copyright 2021, Summitworks Technologies Inc.



@ Copyright 2021, Summitworks Technologies Inc.

Advantage of Dependency Injection

- makes the code loosely coupled so easy to maintain
- makes the code easy to test

@ Copyright 2021, Summitworks Technologies Inc.

Spring is having 3 containers those 3 containers are

- Core container(BeanFactory)
- J2ee Container(ApplicationContext)
- Web container(WebApplicationContext)

@ Copyright 2021, Summitworks Technologies Inc.

Modules

- Core
- Data Access / Integration: Spring JDBC and Spring ORM.
- Web: Model View Controller Framework
- AOP

@ Copyright 2021, Summitworks Technologies Inc.

Spring Core Container

- The Spring Core container contains core, beans, context and expression language (EL) modules.

@ Copyright 2021, Summitworks Technologies Inc.

- This group comprises of JDBC, ORM, OXM, JMS and Transaction modules. These modules basically provide support to interact with the database.

@ Copyright 2021, Summitworks Technologies Inc.

web

- This group comprises of Web, Web-Servlet, Web-Struts and Web-Portlet. These modules provide support to create web application.

@ Copyright 2021, Summitworks Technologies Inc.

There are 2 type of dependency injections

- By using setter methods
- By using parameterized constructors

@ Copyright 2021, Summitworks Technologies Inc.



Files to create for simple example

- create the class
- create the xml file to provide the values
- create the test class
- Load the spring jar files
- Run the test class

@ Copyright 2021, Summitworks Technologies Inc.

Create the test class

- The **Resource** object represents the information of applicationContext.xml file. The Resource is the interface and the **ClassPathResource** is the implementation class of the Resource interface. The **BeanFactory** is responsible to return the bean. The **XmlBeanFactory** is the implementation class of the BeanFactory. There are many methods in the BeanFactory interface. One method is **getBean()**, which returns the object of the associated class.

@ Copyright 2021, Summitworks Technologies Inc.

Java bean with Objects and collections

- We need to change the xml file if you have java bean objects as a instance variables and collection object

@ Copyright 2021, Summitworks Technologies Inc.

The IoC container is responsible to instantiate, configure and assemble the objects. The IoC container gets informations from the XML file and works accordingly. The main tasks performed by IoC container are:

- to instantiate the application class
- to configure the object
- to assemble the dependencies between the objects

There are two types of IoC containers. They are:

- **BeanFactory**
- **ApplicationContext**

@ Copyright 2021, Summitworks Technologies Inc.

Any Queries?

@ Copyright 2021, Summitworks Technologies Inc.