Spring Framework

Content

2—Setting up new spring project

3— Important terminology used in spring framework

3—tight coupling

3—loose coupling

4—loose coupling in spring

5—how to print logs in spring framework

@Component

@Primary

@ComponentScan

Maven

Dependency Injection Types

Setting up new spring project

1. start.spring.io (keep this URL in browser)
2. Select

Project – maven project

Language – java

Spring boot – 2.4.5(select latest version should not have snapshot and m1/m2/m3)

Project metadata

Group—give package name

Project—learnSpring

Click on generate

1. Download -> extract -> copy somewhere
2. In Eclipse -> file -> Import ->(type there)existing maven -> next -> browse ->folder(where spring file has extracted) -> pom.xml(select) ->finish

Important terminology used in spring framework

Terminology

1. Tight coupling
2. Loose coupling
3. IOC Container
4. Application context
5. Component Scan
6. Dependency injection
7. Spring Beans
8. Auto wiring
9. Tight coupling

When two classes are highly dependent on each other it is called tight coupling

It occurs when a class takes many responsibilities or where a change in one class requires a change in another class

Code

In tight coupling we have used

1. Game runner class
2. Game class -> Mario, SuperContra
3. Loose coupling

* loose coupling means classes are mostly independent
* If the class A has only the knowledge about class B is what class B has exposed through its interface

Then Class A and Class B are said to be loosely coupled

* The example of loose coupling are interface

Code

In loose coupling we have used

1) Game runner class

2) Game class -> Mario, SuperContra

3) GamingConsole Interface

1. Loose coupling in spring

Spring frame work will manage all our objects

Code

We have used

1) Game runner class

2) Game class -> Mario, SuperContra

3) GamingConsole Interface

How to print logs in spring frame work

Project – learnSpring

src/main/resources

Application properties (open this file and type below line)

Logging.level.org.springframework = DEBUG

And Run

All logs will get print

@Component

* Class with @Component annotation is managed by Spring framework
* Without having to write any explicit code spring framework will scan application for classes with @Component
* Instantiate them and inject any specified dependencies into them
* Inject them whenever needed

Dependency

A java class has a dependency on another class if it uses an instance of that class we call this a class dependency

e.g game runner needs gaming console implementation

Mario game is a implementation of gaming console so Mario game is the dependency of game runner class

Component Scan

How does spring framework find component classes

It scan packages using

@ComponentScan ({"com.sau.learnSpring","com.sau.learnSpring.game.dummy"})

When we keep package name in @ComponentScan spring framework will scan @Component classes of that package also

Steps that spring framework follows

Lets Debug

1. Identify the candidate @Component class
2. Creating shared instance of singleton bean
3. Autowiring by type from bean name “GameRunner” via constructor to bean name “MarioGame”

What Spring framework do ?

* It does a component scan it identifies Component classes
* It creates instance for them
* It identifies the dependencies that are present for each of these classes
* It creates instance for those dependencies that are present for each of these classes
* It would wire the dependencies into the beans

This entire process is called dependency injection

This is also called IOC (inversion of control)

Why we called this process IOC?

* Earlier as a programmer we are responsible for creating classes, wiring classes, creating object of class
* However now we gave responsibility to Spring framework
* So instead of programmer having control of creating the objects and wiring them together the spring framework now have this control
* And that’s why this is called inversion of control(IOC)

Spring Beans

* Spring bean is an object managed by spring framework
* Spring framework creates instance of the classes which are managed by spring framework
* The creation of object, the life cycle and the destruction of the object everything is done by spring framework that why these are called spring beans

Dependency Injection

Identify Beans

Their dependencies

And wire them together

Dependency Injection is the main functionality provided by Spring IOC

The spring module is responsible for injecting dependencies through either constructor or setter method

IOC Container

The technical component the spring framework which implements the dependency injection logic is called IOC Container

So dependency injection is a concept and the dependency injection feature enabling framework is provided by IOC Container

Types of IOC Container

1. Application context
2. Bean factory

IOC Container manages the lifecycle of beans and dependencies

Basically IOC is about identifying the dependencies and wiring or injecting dependencies into the beans

Does the spring framework really add values

* In the game runner app we have very few classes
* But real world applications are much more complex
* Multiple layers are present in real world application(e.g Web, Business, Data, Database)
* Web > Business > Data > Database
* Each layer is dependent on the layer below it

e.g business layer class talk to data layer class i.e data layer class is dependency of business layer class

* There are thousands of dependencies in every application
* With spring framework instead of focusing on objects, their dependencies and wiring you can focus on business logic of your application
* Spring framework manages the life cycle of Objects >

Mark component using annotation @Component

Mark dependencies using @Autowired

Allow spring framework to do its magic

Create a simple web application

1. Open(src\pom.xml)

* In <dependencies>………</dependencies>
* Add dependency

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<groupId> is a package name

<artifactId> is a class name

For web application add ( spring-boot-starter-web)it simplifies the creation of web Application and REST API

1. In project learn spring

Src/main/java

Create package com.sau.learnSpring.sample.enterprise

Create class Controller

Controller class

**package** com.sau.learnSpring.sample.enterprise.controller;

@RestController

**public** **class** Controller {

@GetMapping("/sau") // user typed url

**public** **String** displaySum() {

**return** “sau is good girl”;

}

}

//business logic

Class BusinessService{

}

//getting data

class DataService{

}

URL to run this application on web (<https://localhost:8080/sau>)

Controller class is in the web layer this class is responsible for sending the response to the user in the right format

In interprize application we have multiple layers

DataService class it is in the data layer this class is responsible for getting data from the data base

Business service class this class is responsible for business logic

@RestController to return data back in right format (in XML format or JSON format or history

e-mail format)

On web(Chrome) > put a link (<https://localhost:8080/sau>)

8080 is the default port for spring framework

We can see on page sau is a good girl

@Primary

If you have multiple beans and you want to choose one of them then give @Primary annotation to that class

In order to provide priority to that class we can use @Primary

e.g if you have provided @Component to both Mario and superContra game but you want to run superContra then give @primary annotation in superContra

@ComponentScan

@ComponentScan ({"com.sau.learnSpring","com.sau.learnSpring.game.dummy"})

When we keep package name in @ComponentScan spring framework will scan @Component classes of that package also

Maven

Maven is Automation tool which acts as dependency manager for java application

* What happens if you manually download Spring JAR (without using maven project) ?

Spring jar needs other JARs , so we have to download all jars and its dependencies

If you have to upgrade to the new version of spring framework, then you have to go through the entire cycle again

Maven manages application dependencies (JARs needed by apps)

Maven would download spring framework and its dependencies

Important feature of maven

* Defines simple project setup that follows best practices
* Enables consistent usage across all projects
* Manages dependency update and transitive dependencies

Dependency Injection Types

1. Constructor based
2. Setter based
3. Field

Constructor based

Dependencies are set by creating the bean by using its constructor

// using constructor

GamingConsole game;

@Autowired

**public** GameRunner(GamingConsole game) {

System.***out***.println("using constructor");

**this**.game = game;

}

Setter based

Dependencies are set by calling setter methods on your beans

// using setter

@Autowired

**public** **void** setGame(GamingConsole game) {

**this**.game = game;

}

field

No setter No constructor dependency is injected by using reflection

// field

@Autowired

GamingConsole game;

Note

Which one should we use?

Spring team recommends constructor-based injection as dependencies are automatically set when an object is created.

Spring Modules

1. Core
2. Testing
3. DataAccess
4. WebServlet
5. Integration
6. WebReactive

Spring framework is divided into modules

1. Core

IOC Container, Application context, Autowired, @Component, @ComponentScan all these are part of the core

1. Testing

Mock Objects, Spring MVC Test etc

1. Data Access

Transactions, JDBC, JPA etc (when we want to talk to database spring allows us to connect these frameworks)

1. Web Servlet

Spring MVC etc

1. Web reactive

Spring WebFlux etc

1. Integration

JMS etc

Spring Projects

Spring keeps evolving (REST APT > Microservices > Cloud)

1. Spring Boot

Most popular framework to build microservices

1. Spring Cloud

Build cloud native application

1. Spring Data

Integrate with different types of data bases: NoSQL and relational

1. Spring Integration

Address challenges with Integration with different types of data bases

1. Spring Security

Secure your web application or REST APT or mocroservice

Review of spring framework

* the advantage spring framework is the you can write loosely coupled code focusing on your business logic
* It provides you with additional Flexibility because you can pick and choose which spring module to use
* And spring framework is evolving with time so your application can stay in touch with evolving trends