**Springboot**

**Spring**

**Benefits**

Inversion of control – Hand over control of things such as garbage collection to the spring framework.

Dependency Injection – Allowing loose coupling between classes etc, so that it can be possible to test each separately , even when, in reality, two classes etc are always connected.

Aspect Oriented Programming (AOP) – Factoring out cross-cutting concerns, i.e, some tasks are carried out in every module and these are separated out.

1. **Ways of Doing Dependency Injection**

* Constructor-based
* XML Configuration
* Annotation based
* Java Configuration

1. **XML Configuration**

This is done in the pom.xml file as well as a arbitrarily created file in the “resources” directory under “src/main”.

In the pom.xml, we define the spring framework as the dependency.

<dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>6.0.11</version>  
 </dependency>  
</dependencies>

In the resources folder, we can create a file called “spring.xml”. This will serve as the spring container.

Here:

* We define the xml version.
* We import the spring namespace in the xml. Depends on the spring version and comes from the documentation.

*https://docs.spring.io/spring-framework/docs/4.2.x/spring-framework-reference/html/beans.html#beans-factory-xml-import*

* We define the beans. (The classes/objects in the projects).
* We reload maven. A button appears in the file to do this.

<?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 id="coach" class="com.springdemo.Coach"></bean>  
  
</beans>

To access the beans in the container (above), there are two interfaces:

* Bean factory
* Application context – extends bean factory and therefore has a lot more features. More commonly used

Within the main class file, we import the “ApplicationContext” library from the “com.springframework.context” package. We then instantiate an ApplicationContext object. It provides many options to get the spring context. We choose the XML based configuration one and pass to it the file “spring.xml” file.

From this AppilcationContext object we can get whatever bean we want using the getBean() method.

import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class Main {  
 public static void main(String[] args) {  
  
 ApplicationContext context = new ClassPathXmlApplicationContext("spring.xml");  
  
 Coach coach = context.getBean(Coach.class);