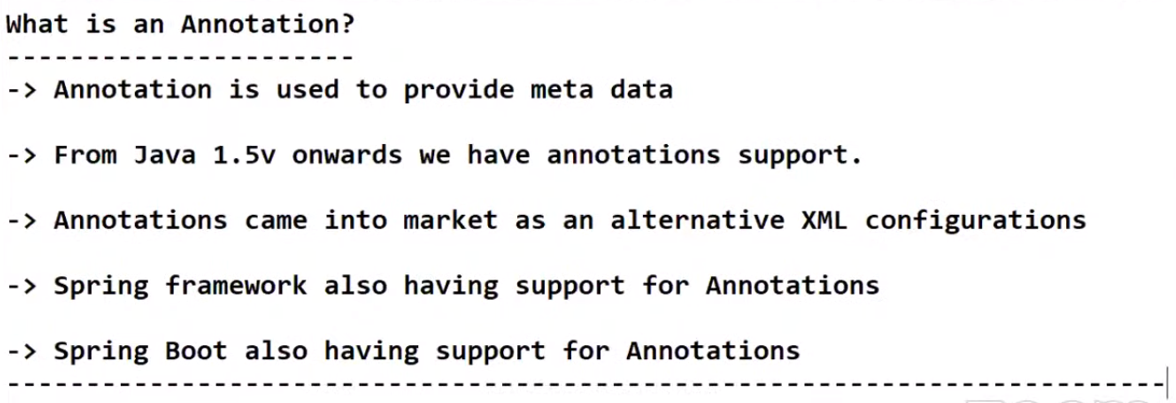
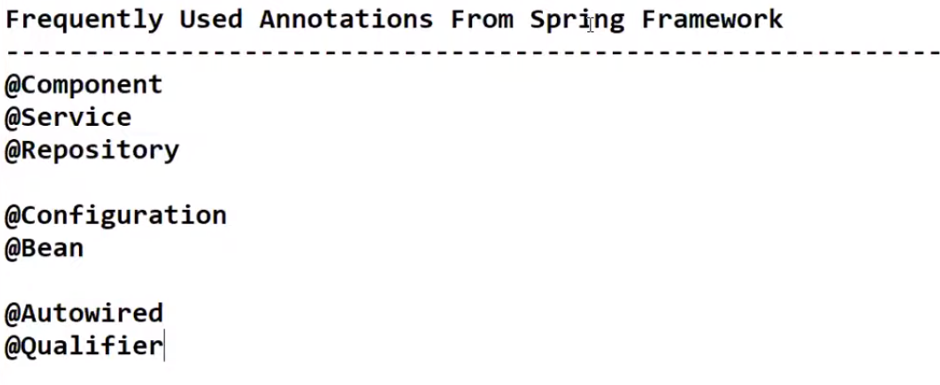
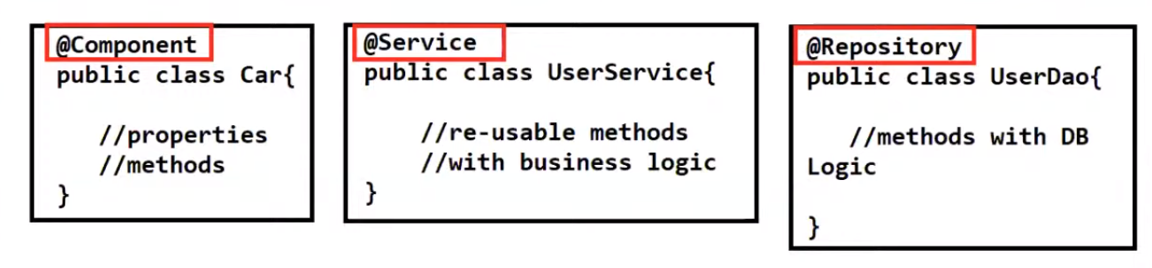
Spring Annotations



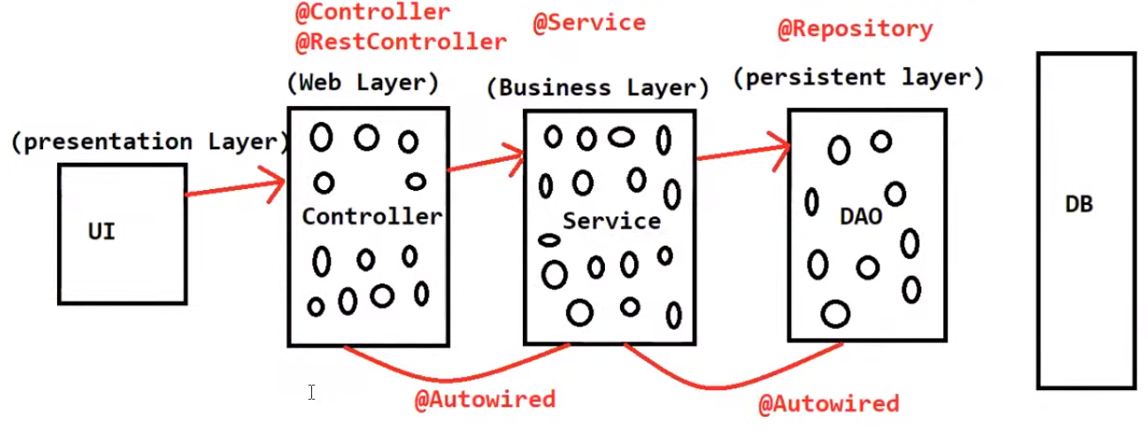


**NOTE: All spring annotations can be used in Spring-boot also.**

* Life cycle of java class is controlled by IOC container.
* IOC container and Dependency Injections are the fundamental concepts of Spring Framework.
* We can represent a Java Class as a Spring bean in 2 ways.
  + Using XML configuration.
    - *<bean id=’obj’ class=’com.packagename.BeanName/>*
  + Using Annotations.
    - *@Component, @Service , @Repository*



1. @**Component** creates ‘Car’ class as a Spring bean.
2. @**Service** or @**Component** can be used to make ‘UserService’ class as a Spring bean.
3. @**Repository** can be used to make class as a Spring bean.
4. By converting above mentioned classes with annotations, Dependency Mgmt will be taken care by IOC container.
5. @**Controller** and @**RestController** are also can be used to create a spring bean, but they are especially used for Spring Web MVC model.

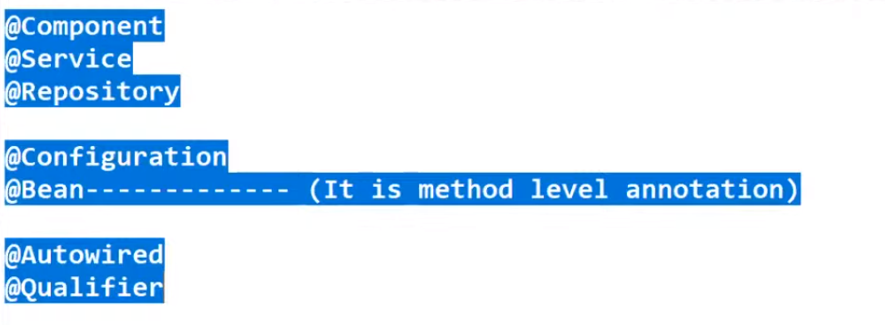


**NOTE**:

1. There will be multiple ‘Controller’, ‘Service’ and ‘DAO’ classes in real time.
2. If you are having 20 UI screens in your project, then 20 Controller classes will be created to accept the web requests from the UI screen.(May use Web Services).
3. Even in Monolithic or Micro Service Application Architecture, Layered architecture is fallowed.

**NOTE**:

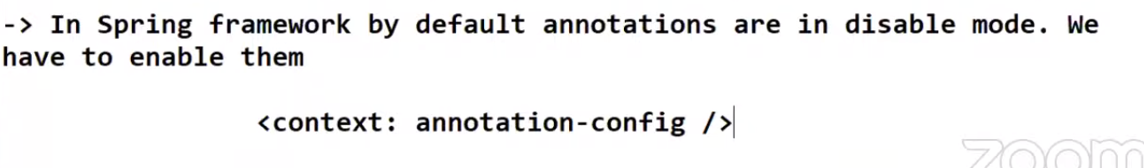
* When we use JSP or Thyme-Leaf for presentation layer, we use @**Controller**.
* When we use Angular or React for presentation layer, we use @**RestController**. This annotation supports JSON data.
* @**Service** for Business Layer
* @**Repository** for DAO Layer.
* @**Autowired** is used to inject one classes object into another class without using XML.
* We can use @**Autowired** annotation in Method (setter injection) level, Constructor level and Variable level.



@**Configuration**: To avoid XML configuration and want to behave java class as a configuration class than use this annotation.

@**Bean**: Is a method level annotation. When we want to create our method with a customized logic and want to maintain that class on our own (without using spring IOC container).

@**Qualifier**: Can be used only at Autowiring. To remove ambiguity between two or more classes are qualified for Autowiring.

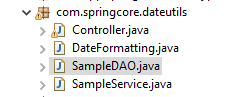


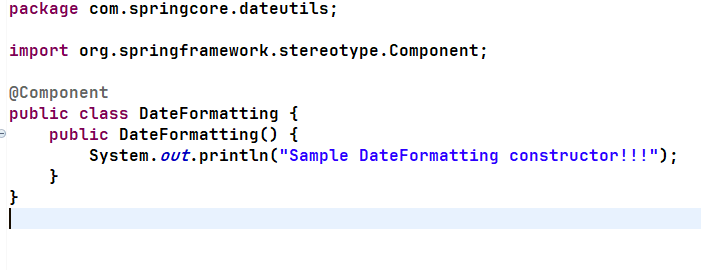
We have to tell IOC container to which package(s) it needs to scan, in order to create object(s)

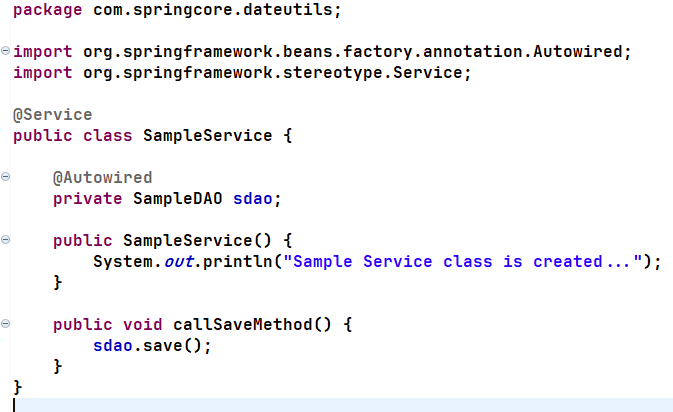
Also add, **<context: component-scan base-package=”com.package”/>**

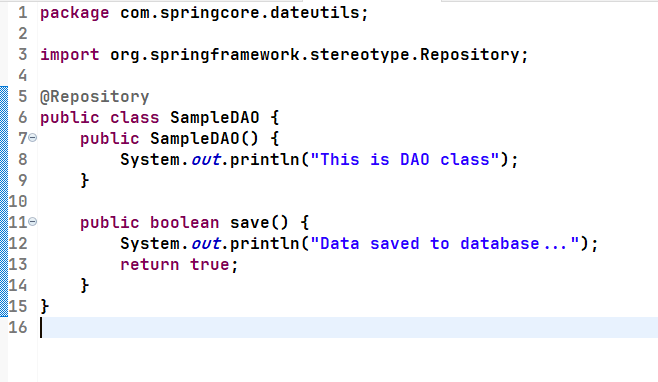
**NOTE**: But in spring boot automatically annotations are enabled.

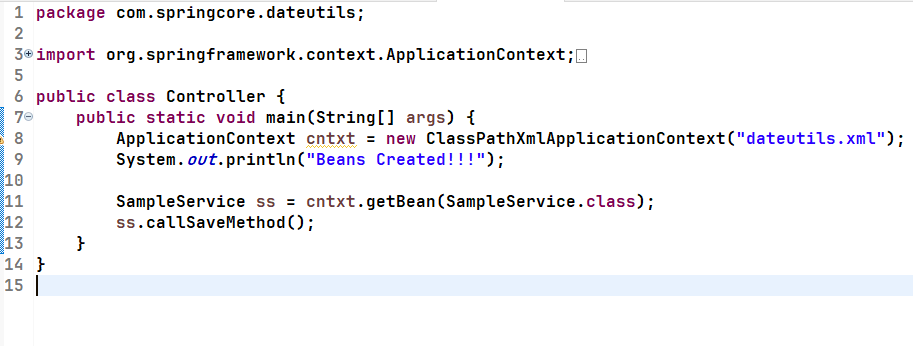
Example Applications:











dateutils.xml:

**<?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"**

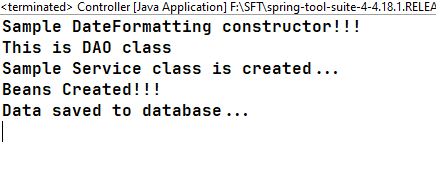
**xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd**

**http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd">**

**<context:annotation-config />**

**<context:component-scan base-package="com.springcore.dateutils" />**

**</beans>**



To practice Spring Programming:

https://codeanywhere.com/languages/spring#:~:text=You%20can%20code%2C%20learn%2C%20build,online%20browser%20based%20Cloud%20IDE