<https://blog.csdn.net/yuanmxiang/article/details/51120148>

**1、Spring容器简介**   
容器就是一个对于某种对象管理的综合框架，容器管理着对象的生命周期，Spring容器可以帮助你管理所有的Bean对象，专业术语称之为IOC控制反转。而在Spring中我们既可以通过xml文件配置来实现向容器注册bean，也可以通过注解@Component、@Controller、@ Service、@ Repository来向容器注册。

**2、注解使用前准备工作**   
用注解来向Spring容器注册Bean。需要在applicationContext.xml中注册

<context:component-scan base-package=”pagkage1[,pagkage2,…,pagkageN]”/>。

* 1

如：在base-package指明一个包

<context:component-scan base-package="cn.gacl.java"/>

* 1

表明cn.gacl.java包及其子包中，如果某个类的头上带有特定的注解【@Component/@Repository/@Service/@Controller】，就会将这个对象作为Bean注册进Spring容器。也可以在   
<context:component-scan base-package=” ”/>   
中指定多个包，如：

<context:component-scan base-package="cn.gacl.dao.impl,cn.gacl.service.impl,cn.gacl.action"/>

* 1

多个包逗号隔开。

**1、@Component**   
@Component   
是所有受Spring 管理组件的通用形式，@Component注解可以放在类的头上，@Component不推荐使用。

**2、@Controller**   
@Controller对应控制层的Bean，也就是Action，例如：

@Controller

@Scope("prototype")

public class UserAction extends BaseAction<User>{

……

}

* 1
* 2
* 3
* 4
* 5

使用@Controller注解标识UserAction之后，就表示要把UserAction交给Spring容器管理，在Spring容器中会存在一个名字为”userAction”的action，这个名字是根据UserAction类名来取的。

注意：如果@Controller不指定其value，则默认的bean名字为这个类的类名首字母小写，如果指定   
value{@Controller(value="UserAction")}   
或者   
{@Controller("UserAction")}，   
则使用value作为bean的名字。   
这里的UserAction还使用了@Scope注解，@Scope(“prototype”)表示将Action的范围声明为原型，可以利用容器的scope=”prototype”来保证每一个请求有一个单独的Action来处理，避免struts中Action的线程安全问题。spring 默认scope 是单例模式(scope=”singleton”)，这样只会创建一个Action对象，每次访问都是同一Action对象，数据不安全，struts2 是要求每次次访问都对应不同的Action，scope=”prototype” 可以保证当有请求的时候都创建一个Action对象。

**3、@ Service**   
@Service对应的是业务层Bean，例如：

@Service("userService")

public class UserServiceImpl implements UserService {

……… }

* 1
* 2
* 3

@Service(“userService”)注解是告诉Spring，当Spring要创建UserServiceImpl的的实例时，bean的名字必须叫做”userService”，这样当Action需要使用UserServiceImpl的的实例时,就可以由Spring创建好的”userService”，然后注入给Action：在Action只需要声明一个名字叫“userService”的变量来接收由Spring注入的”userService”即可，具体代码如下：

// 注入userService

@Resource(name = "userService")

private UserService userService;

* 1
* 2
* 3

**4、@ Repository**   
@Repository对应数据访问层Bean ，例如：

@Repository(value="userDao")

public class UserDaoImpl extends BaseDaoImpl<User> {

………

}

* 1
* 2
* 3
* 4

@Repository(value=”userDao”)注解是告诉Spring，让Spring创建一个名字叫“userDao”的UserDaoImpl实例。

当Service需要使用Spring创建的名字叫“userDao”的UserDaoImpl实例时，就可以使用@Resource(name = “userDao”)注解告诉Spring，Spring把创建好的userDao注入给Service即可。

// 注入userDao，从数据库中根据用户Id取出指定用户时需要用到

@Resource(name = "userDao")

private BaseDao<User> userDao;

* 1
* 2
* 3

以上四个都是注册bean使用的注解，与此同时Spring还提供了三个注入bean的注解，具体如下。

**1、@Resource**   
@Resource可以以 name 或 type 方式注入，默认按 byName自动注入。@Resource有两个属性是比较重要的，分是name和type。如果使用name属性，则使用byName的自动注入策略。而使用type属性时则使用byType自动注入策略。其也是JDK提供的注解。

**2、@AutoWired**   
@AutoWired只能以 type 方式注入，其是Spring提供的注解。

**3、@Qualifier**   
@Qualifier 只能以 name 方式注入

推荐使用@Resource