

Spring基于java配置进行注解

一、 Configuration

1、 作用：

指定当前类是一个配置类，相当于bean.xml

2、 细节：

当配置类作为AnnotationConfigApplicationContext对象创建的参数时，该注解可以不写。

3、 例子

当有两个配置类

```
SpringConfiguration.class
```

```
@Configuration
```

```
public class SpringConfiguration {  
}
```

```
JdbcConfig.class
```

```
public class JdbcConfig{  
}
```

如果AnnotationConfigApplicationContext对象创建的参数，

//此时SpringConfiguration.class可以不用@Configuration注解，但是JdbcConfig.class需要注解

```
ApplicationContext ac = new
```

```
AnnotationConfigApplicationContext(SpringConfiguration.class);
```

//此时JdbcConfig.class也不需要注解

```
ApplicationContext ac = new
```

```
AnnotationConfigApplicationContext(SpringConfiguration.class, JdbcConfig.class);
```

二、 ComponentScan

1、 作用：

用于通过注解指定spring在创建容器时要扫描的包

@ComponentScan ({ “package1”, “package2” }) 可以同时扫描多个

包

2、属性：

basePackages:

value: 用于指定创建容器时要扫描的包。

我们使用此注解就等同于在xml中配置了：

```
<context:component-scan base-package="com.itheima">
```

```
</context:component-scan>
```

等同于：

```
@Configuration
```

```
@ComponentScan("com.itheima")
```

```
public class SpringConfiguration {  
}
```

三、Bean

1、作用：

用于把当前方法的返回值作为bean对象存入spring的ioc容器中

2、属性：

name:用于指定bean的id。当不写时，默认值是当前方法的名称

3、细节：

当我们使用注解配置方法时，如果方法有参数，spring框架会去容器中查找有没有可用的bean对象。

查找的方式和Autowired注解的作用是一样的

```
@Bean(name="runner")
```

```
@Scope("prototype")
```

```
public QueryRunner createQueryRunner(DataSource dataSource){
```

```
    return new QueryRunner(dataSource);
```

```
}
```

createQueryRunner中要使用DataSource，就要先把DataSource加入spring容器中，才能查找到

```
@Bean(name="ds2")
```

```
public DataSource createDataSource(){
```

```
    try {
```

```
        ComboPooledDataSource ds = new ComboPooledDataSource();
```

```
        ds.setDriverClass(driver);
```

```
        ds.setJdbcUrl(url);
```

```
        ds.setUser(username);
```

```
        ds.setPassword(password);
```

```
        return ds;
```

```
}catch (Exception e){  
    throw new RuntimeException(e);  
}  
}
```

四、Import

1、作用：

用于导入其他的配置类

2、 属性：

value: 用于指定其他配置类的字节码。

当我们使用Import的注解之后，有Import注解的类就是父配置类，而导入的都是子配置类

例如：

```
@Configuration  
@ComponentScan("com.itheima")  
@Import(JdbcConfig.class)  
public class SpringConfiguration {  
}
```

只需加载SpringConfiguration.class即可导入JdbcConfig.class

```
ApplicationContext ac = new  
AnnotationConfigApplicationContext(SpringConfiguration.class);
```

五、 PropertySource

1、 作用：

用于指定properties文件的位置

2、 属性：

value: 指定文件的名称和路径。

关键字: classpath, 表示类路径下

例如：

为了简化JdbcConfig，可以创建一个jdbcConfig.properties:

```
jdbc.driver=com.mysql.jdbc.Driver  
jdbc.url=jdbc:mysql://localhost:3306/easy  
jdbc.username=root  
jdbc.password=1234
```

然后：

```
@Configuration  
@PropertySource("classpath:jdbcConfig.properties")
```

```
public class SpringConfiguration {  
}
```