依赖注入

C-2 创建: 张林伟, 最后修改: 张林伟 01-03 11:30

问题:

代码块 Java
```java
@Service
public class BaseRatioInfoDAOImpl implements IBaseRatioInfoDAO {
    ...
}
```

代码块 Java
```java
    @Autowired
    private IBaseRatioInfoDAO baseRatioInfoDAO;
```

当 IBaseRatioInfoDAO 只有一个实现类时，@Autowired 将 baseRatioInfoDAOImpl 注入。如果 IBaseRatioInfoDAO 有多个实现类，该如何注入呢?

如果有多个类，使用 @Autowired 注入会报错，因为 **@Autowired 的注入方式是 byType 方式**，当要注入的类型在容器中存在多个时，Spring是不知道要引入哪个实现类的，所以会报错。

多个实现类可以使用 @Resource 或 @Qualifier 注入。

@Resource 默认是按照 byName 的方式注入的， 如果通过 byName 的方式匹配不到，再按 byType 的方式去匹配。

代码块 Java
```java
    @Resource(name = "baseRatioInfoDAOImpl")
    private IBaseRatioInfoDAO baseRatioInfoDAO;
```

@Qualifier 注解也是 byName的方式。配合 @Autowired 使用相当于 @Resource 。

代码块 Java
```java
    @Autowired
    @Qualifier(name = "baseRatioInfoDAOImpl")
    private IBaseRatioInfoDAO baseRatioInfoDAO;
```

附上 Stack Overflow 回答

> @Autowired can be used alone . If it is used alone , it will be wired by type . So problems arises if more than one bean of the same type are declared in the container as @Autowired does not know which beans to use to inject. As a result , use @Qualifier together with @Autowired to clarify which beans to be actually wired by specifying the bean name (wired by name)
>
> @Resource is wired by name too . So if @Autowired is used together with @Qualifier , it is the same as the @Resource.
>
> The difference are that @Autowired and @Qualifier are the spring annotation while @Resource is the standard java annotation (from JSR-250) . Besides , @Resource only supports for fields and setter injection while @Autowired supports fields , setter ,constructors and multi-argument methods injection.
>
> It is suggested to use @Resource for fields and setter injection. Stick with @Qualifier and @Autowired for constructor or a multi-argument method injection.
>
> See this:
>
> If you intend to express annotation-driven injection by name, do not primarily use @Autowired - even if is technically capable of referring to a bean name through @Qualifier values. Instead, prefer the JSR-250 @Resource annotation which is semantically defined to identify a specific target component by its unique name, with the declared type being irrelevant for the matching process.

注入Map、List类型

代码块 Java
```java
// 接口
public interface Factory {
    Object produce();
}

// 实现类1
@Order(1)
@Service("StringFactory")
```

```java
public class StringFactory implements Factory {
    @Override public Object produce() {
        return "a";
    }
}

// 实现类2
@Order(2)
@Service("IntegerFactory")
public class IntegerFactory implements Factory {
    @Override public Object produce() {
        return Integer.valueOf(1);
    }
}
```

注入

代码块 Java

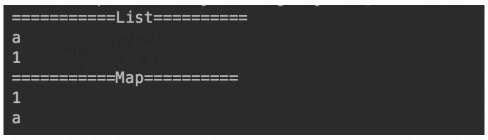
```java
    @Autowired
    private List<Factory> factories;    // Order注解只对List类型有效

    @Autowired
    private Map<String, Factory> factoryMap;    // key为bean的名称，value为bean本身
```

测试：

代码块 Java

```java
    public void testAutowiredMapList() {
        System.out.println("===========List==========");
        factories.forEach(factory -> System.out.println(factory.produce()));
        System.out.println("===========Map==========");
        factoryMap.forEach((k, v) -> System.out.println(v.produce()));
    }
```



```
===========List==========
a
1
===========Map==========
1
a
```