

@Autowired和@Resource

资源目录下创建包要使用com/wxiang/dao的形式

使用zookeeper注册服务时，服务提供方暴露服务的注解是：

// 注意引入的包：com.alibaba.dubbo.config.annotation.Service

@Service(interfaceClass = CheckItemService.class)

服务消费方查找服务

// 查找服务

@Reference

private CheckItemService checkItemService;

当我们使用自动配置的方式装配Bean时，如果这个Bean有多个候选者，假如其中一个候选者具有@Primary注解修饰，该候选者会被选中，作为自动配置的值。

@Autowired会自动把相同类型的多个对象收集到集合中。

```
public interface IUser {  
    void say();  
}
```

@Service

```
public class User1 implements IUser{  
    @Override  
    public void say() {  
    }  
}
```

@Service

```
public class User2 implements IUser{  
    @Override  
    public void say() {  
    }  
}
```

除了@Qualifier注解之外，还能使用@Primary注解解决上面的问题。在User1上面加上@Primary注解：

@Primary

@Service

```
public class User1 implements IUser{  
    @Override  
    public void say() {  
    }  
}
```

当我们使用自动配置的方式装配Bean时，如果这个Bean有多个候选者，假如其中一个候选者具有@Primary注解修饰，该候选者会被选中，作为自动配置的值。

@Service

```
public class UserService {
```

@Autowired

```
private List<IUser> userList;
```

```

@Autowired
private Set<IUser> userSet;

@Autowired
private Map<String, IUser> userMap;

public void test() {
    System.out.println("userList:" + userList);
    System.out.println("userSet:" + userSet);
    System.out.println("userMap:" + userMap);
}
}

```

```

@RequestMapping("/u")
@RestController
public class UController {

```

```

    @Autowired
    private UserService userService;

```

```

    @RequestMapping("/test")
    public String test() {
        userService.test();
        return "success";
    }
}

```

```

userList:[com.sue.cache.service.test19.User1@2513a118, com.sue.cache.service.test19.User2@2bfb583b]
userSet:[com.sue.cache.service.test19.User1@2513a118, com.sue.cache.service.test19.User2@2bfb583b]
userMap:{user1=com.sue.cache.service.test19.User1@2513a118, user2=com.sue.cache.service.test19.User2@2bfb583b}

```

@SpringBootApplication注解，它里面内置了@ComponentScan注解的功能。

注入Filter或Listener

web应用启动的顺序是：listener->filter->servlet。

```

public class UserFilter implements Filter {

```

```

    @Autowired
    private IUser user;

```

```

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        user.say();
    }

```

```

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain
chain) throws IOException, ServletException {

    }
}

```

```

@Override
public void destroy() {
}
}

```

```

@Configuration
public class FilterConfig {

```

```

@Bean
public FilterRegistrationBean filterRegistrationBean() {
    FilterRegistrationBean bean = new FilterRegistrationBean();
    bean.setFilter(new UserFilter());
    bean.addUrlPatterns("/*");
    return bean;
}
}

```

程序启动会报错，tomcat无法正常启动。

众所周知，springmvc的启动是在DisptachServlet里面做的，而它是在listener和filter之后执行。如果我们想在listener和filter里面@Autowired某个bean，肯定是不行的，因为filter初始化的时候，此时bean还没有初始化，无法自动装配。

如果工作当中真的需要这样做，我们该如何解决这个问题呢？

```

public class UserFilter implements Filter {

    private IUser user;

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {
        ApplicationContext applicationContext =
WebApplicationContextUtils.getWebApplicationContext(filterConfig.getServletContext());

        this.user = ((IUser)(applicationContext.getBean("user1")));
        user.say();
    }

    @Override
    public void doFilter(ServletRequest request, ServletResponse response, FilterChain
chain) throws IOException, ServletException {

    }

    @Override
    public void destroy() {

    }
}

```

答案是使用WebApplicationContextUtils.getWebApplicationContext获取当前的ApplicationContext，再通过它获取到bean实例。

@Autowired和@Resource的区别

@Autowired功能虽说非常强大，但是也有些不足之处。比如：比如它跟spring强耦合了，如果换成了JFinal等其他框架，功能就会失效。而@Resource是JSR-250提供的，它是Java标准，绝大部分框架都支持。除此之外，有些场景使用@Autowired无法满足的要求，改成@Resource却能解决问题。接下来，我们重点看看@Autowired和@Resource的区别。

@Autowired默认按byType自动装配，而@Resource默认byName自动装配。

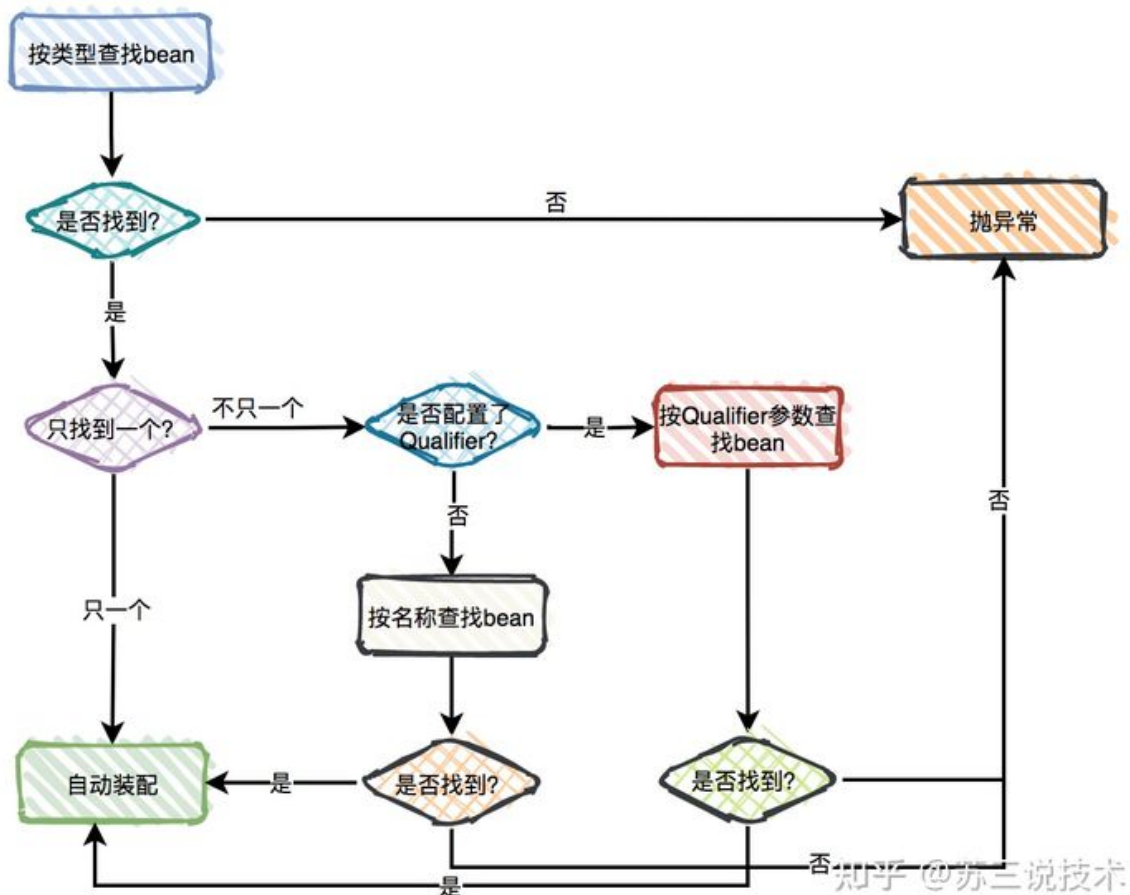
@Autowired只包含一个参数：required，表示是否开启自动准入，默认是true。而@Resource包含七个参数，其中最重要的两个参数是：name 和 type。

@Autowired如果要使用byName，需要使用@Qualifier一起配合。而@Resource如果指定了name，则用byName自动装配，如果指定了type，则用byType自动装配。

@Autowired能够用在：构造器、方法、参数、成员变量和注解上，而@Resource能用在：类、成员变量和方法上。

@Autowired是spring定义的注解，而@Resource是JSR-250定义的注解。

@Autowired的装配顺序如下：



@Resource的装配顺序如下：

如果同时指定了name和type：

