

## SpringContextHolder 静态持有SpringContext的引用

转载

2018年02月24日 10:29:05

### SpringContextHolder 静态持有SpringContext的引用

```
package com.test.quartz;

import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;

/**
 *
 * 以静态变量保存Spring ApplicationContext，可在任何代码任何地方任何时候中取出ApplicaitonContext.
 * @author zhuh
 *
 */

public class SpringContextHolder implements ApplicationContextAware {

    private static ApplicationContext applicationContext;

    // 实现ApplicationContextAware接口的context注入函数，将其存入静态变量。
    public void setApplicationContext(ApplicationContext applicationContext) {
        SpringContextHolder.applicationContext = applicationContext;
    }

    // 取得存储在静态变量中的ApplicationContext。
    public static ApplicationContext getApplicationContext() {
        checkApplicationContext();
        return applicationContext;
    }

    // 从静态变量ApplicationContext中取得Bean，自动转型为所赋值对象的类型。
    @SuppressWarnings("unchecked")
    public static <T> T getBean(String name) {
        checkApplicationContext();
        return (T) applicationContext.getBean(name);
    }

    // 从静态变量ApplicationContext中取得Bean，自动转型为所赋值对象的类型。
    // 如果有多个Bean符合Class，取出第一个。
    @SuppressWarnings("unchecked")
    public static <T> T getBean(Class<T> clazz) {
        checkApplicationContext();
        return (T) applicationContext.getBeansOfType(clazz);
    }

    private static void checkApplicationContext() {
        if (applicationContext == null) {
            throw new IllegalStateException("applicaitonContext未注入,请在applicationContext
        }
    }
}
```

<!-- 用于持有ApplicationContext,可以使用SpringContextHolder.getBean('xxxx')的静态方法得到spring bean对象 -->

```
<bean class="com.xxxxx.SpringContextHolder" />
```

该工具类主要用于：那些没有归入spring框架管理的类却要调用spring容器中的bean提供的工具类。

在spring中要通过IOC依赖注入来取得对应的对象，但是该类通过实现ApplicationContextAware接口，以静态变量保存Spring ApplicationContext，可在任何代码任何地方任何时候中取出ApplicaitonContext。

如此就不能说说org.springframework.context.ApplicationContextAware这个接口了：

当一个类实现了这个接口（ApplicationContextAware）之后，这个类就可以方便获得ApplicationContext中的所有bean。换句话说，就是这个类可以直接获取spring配置文件中，所有有引用到的bean对象。

除了以上SpringContextHolder类之外，还有不需要多次加载spring配置文件就可以取得bean的类：

1.Struts2框架中，在监听器中有这么一句

```
ApplicationContext context = WebApplicationContextUtils.getWebApplicationContext(event.getServletContext)
```

之后可以用

```
scheduleService = (IScheduleService)context.getBean("scheduleService");
```

取到对象，请问context都可以取到什么信息，这些信息的来源在哪？是XML里配置了呢，还是固定的一部分信息呢？

2、这个 application封装的是web.xml 内部的信息

而你的web.xml里面有spring的配置文件，所有，里面还包含spring的信息

同样包含struts2的filter信息

总之就是和web.xml有关系的所有信息

3、在web.xml里有这么一段

```
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>/WEB-INF/applicationContext*.xml</param-value>
</context-param>
```

那么在取信息的时候，也会把applicationContext.xml里的信息取出来

使用方式

```
/**
 * Hello world!
 */
public class App {
    public static void main(String[] args) {

        System.out.println("加载spring");
        ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext(new String[]
        {
            "context.xml"
        });
        context.start();

        SpringContextHolder springContextHolder = new SpringContextHolder();
        springContextHolder.setApplicationContext(context);
        //SpringContextHolder.getBean("queryDealService");
        System.out.println("Hello World!");
    }
}
```



发表你的评论