

ApplicationContextAware 的使用

作者: [xfy196](#)

原文链接: <https://ld246.com/article/1568465993273>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

ApplicationContextAware

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

```
package com.funtl.leeshop.commons.context;

import org.apache.commons.lang3.Validate;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.beans.BeansException;
import org.springframework.beans.factory.DisposableBean;
import org.springframework.context.ApplicationContext;
import org.springframework.context.ApplicationContextAware;

public class SpringContext implements ApplicationContextAware, DisposableBean {

    private static final Logger logger = LoggerFactory.getLogger(SpringContext.class);

    private static ApplicationContext applicationContext;

    /**
     * 获取存储在静态变量中的 ApplicationContext
     * @return
     */
    public static ApplicationContext getApplicationContext() {
        assertContextInjected();
        return applicationContext;
    }

    /**
     * 从静态变量 applicationContext 中获取 Bean, 自动转型成所赋值对象的类型
     * @param name
     * @param <T>
     * @return
     */
    public static <T> T getBean(String name) {
        assertContextInjected();
        return (T) applicationContext.getBean(name);
    }

    /**
     * 从静态变量 applicationContext 中获取 Bean, 自动转型成所赋值对象的类型
     * @param clazz
     * @param <T>
     * @return
     */
    public static <T> T getBean(Class<T> clazz) {
        assertContextInjected();
        return applicationContext.getBean(clazz);
    }
}
```

```

/**
 * 实现 DisposableBean 接口, 在 Context 关闭时清理静态变量
 * @throws Exception
 */
public void destroy() throws Exception {
    logger.debug("清除 SpringContext 中的 ApplicationContext: {}", applicationContext);
    applicationContext = null;
}

/**
 * 实现 ApplicationContextAware 接口, 注入 Context 到静态变量中
 * @param applicationContext
 * @throws BeansException
 */
public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {
    SpringContext.applicationContext = applicationContext;
}

/**
 * 断言 Context 已经注入
 */
private static void assertContextInjected() {
    Validate.validState(applicationContext != null, "applicationContext 属性未注入, 请在 spring-context.xml 配置中定义 SpringContext");
}
}

```

同时还需要在

[spring-context.xml](#)

配置文件装配

```
<bean id="springContext" class="com.funtl.leeshop.commons.context.SpringContext" />;
```

注意要将bean的配置放在配置的顶部

pom

需要在 pom.xml 中增加 org.apache.commons:commons-lang3 依赖

```

<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-lang3</artifactId>
  <version>3.5</version>
</dependency>

```

附：完整的pom的文件

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd

```

```
maven-4.0.0.xsd" >
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.funtl</groupId>
  <artifactId>leeshop</artifactId>
  <version>1.0.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <dependencies>
    <!-- Spring Begin -->
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-context</artifactId>
      <version>4.3.17.RELEASE</version>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-webmvc</artifactId>
      <version>4.3.17.RELEASE</version>
    </dependency>
    <!-- Spring End -->

    <!-- Servlet Begin -->
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>3.1.0</version>
      <scope>provided</scope>
    </dependency>
    <!-- Servlet End -->

    <!-- Log Begin -->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>1.7.25</version>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-log4j12</artifactId>
      <version>1.7.25</version>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>jcl-over-slf4j</artifactId>
      <version>1.7.25</version>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>jul-to-slf4j</artifactId>
      <version>1.7.25</version>
    </dependency>
    <dependency>
      <groupId>log4j</groupId>
```

```
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
<!-- Log End -->

<!-- Commons Begin -->
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.5</version>
</dependency>
<!-- Commons End -->
</dependencies>
</project>
```