#### Spring Framework源码阅读

##### 之Application初始化：获取ClassLoader

问题描述

Spring Boot 2.2.5

Spring Framework 5.2.6

JDK 1.8

SpringBoot提供了一个工具类SpringApplication，用于傻瓜式拉起Spring Boot应用。

在初始化阶段，有一步获取ClassLoader的源码。还没有找到应用场景，但这段逻辑写得机巧，在这篇笔记中摘录下来

Java中类加载器（ClassLoader）的作用是维护类名到它相应的二进制字节码文件的映射关系，进而加载系统资源（SystemResource）

使用ClassLoader::getSystemClassLoader方法获得类加载器（APP内部类）

在web应用中，则应获取Tomcat内含类加载器，即使用Thread::getContextThreadLoader方法

解决

@Nullable  
public static ClassLoader getDefaultClassLoader() {  
 ClassLoader cl = null;  
 try {  
 cl = Thread.*currentThread*().getContextClassLoader();  
 }  
 catch (Throwable ex) {  
 // Cannot access thread context ClassLoader - falling back...  
 }  
 if (cl == null) {  
 // No thread context class loader -> use class loader of this class.  
 cl = ClassUtils.class.getClassLoader();  
 if (cl == null) {  
 // getClassLoader() returning null indicates the bootstrap ClassLoader  
 try {  
 cl = ClassLoader.*getSystemClassLoader*();  
 }  
 catch (Throwable ex) {  
 // Cannot access system ClassLoader - oh well, maybe the caller can live with null...  
 }  
 }  
 }  
 return cl;  
}

□