**背景：**

Springboot版本：2.1.6

相关类：

**ServiceA**：通过@Autowired filed注入 msgSender。

**ServiceB**：通过@Autowired filed注入 msgSender。

**BeanDef**：@Configuration，通过@Bean的方式声明Bean msgSender。通过@Autowired filed注入 ServiceB。

msgSender是引入的第三方工具类，通过@Bean的方式声明为Bean。

**现象：**

项目启动抛出异常：

Requested bean is currently in creation: Is there an unresolvable circular reference?

**原因分析：**

查看报错的代码：

**protected void beforeSingletonCreation(String beanName) {**

**if(!this.inCreationCheckExclusions.contains(beanName)&& !this.singletonsCurrentlyInCreation.add(beanName)) {**

**throw new BeanCurrentlyInCreationException(beanName);**

**}**

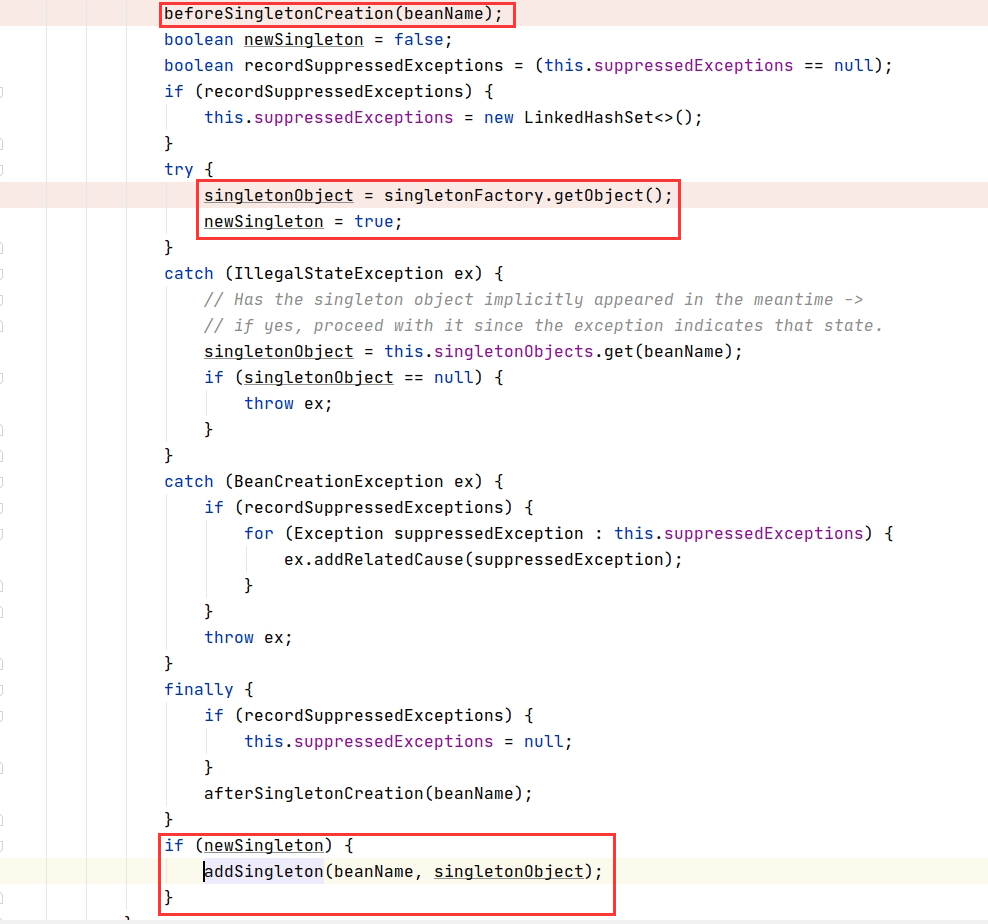
**}**

分析代码得知：

singletonsCurrentlyInCreation：一个保存当前正在创建单例Bean的Set。

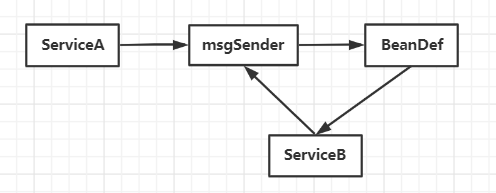
在这个set执行add操作时抛出了异常，对应的beanName为msgSender。

Spring在创建bean时会先将类的工厂方法暴露出来，存放在二级缓存singletonFactories中。当发现循环依赖时，从二级缓存中取出工厂方法并生成earlySingletonObject存放在三级缓存earlySingletonObjects中。



通过查看上述代码得知，spring在创建/获取bean进行依赖注入前需要调用**beforeSingletonCreation**方法。

分析几个类之间的依赖关系：



1. 创建ServiceA，将ServiceA放入singletonsCurrentlyInCreation中，并将其工厂方法放入二级缓存singletonFactories中。

发现需要注入msgSender。

1. **创建msgSender，msgSender是在BeanDef中声明的，因此需要先创建BeanDef，并且无法通过暴露工厂方法放入二级缓存singletonFactories中。**

**但singletonsCurrentlyInCreation中依然添加了msgSender。**

1. 创建BeanDef，将BeanDef放入singletonsCurrentlyInCreation中，并将其工厂方法放入二级缓存singletonFactories中。

发现需要注入ServiceB。

1. 创建ServiceB，将ServiceB放入singletonsCurrentlyInCreation中，并将其工厂方法放入二级缓存singletonFactories中。

发现需要注入msgSender。

1. **在获取msgSender时，由于无法从一级及二级缓存中获取到引用，因此还是走创建Bean的逻辑。**需要先调用**beforeSingletonCreation**方法，在此方法中，singletonsCurrentlyInCreation.add()方法添加msgSender时，返回false，因此抛出异常。

由此得出异常原因：涉及到@Bean声明的Bean，特殊情况下的循环依赖导致验证单例Bean(msgSender)是否正在创建时抛出异常。

在此情形中，只有在ServiceA先加载时才会出现循环依赖的异常，因此在最近一次开发中才发现这个问题。

**解决方案：**

对于定义Bean的configuration类，应保持其纯粹性，不应依赖其它bean，防止出现类似的循环依赖问题。