

明确一点：AbstractBeanFactory\$\$Lambda 是一个匿名的实体。匿名类的父类为ObjectFactory。

ObjectFactory 只有一个方法：

```
@FunctionalInterface
public interface ObjectFactory<T> {
    T getObject() throws BeansException;
}
```

因此采用Lambda写法如下：有点绕。

```
sharedInstance = getSingleton(beanName, () -> {
    try {
        return createBean(beanName, mbd, args);
    }
    catch (BeansException ex) {
    }
});
```

匿名对象的好处： 匿名对象能回调宿主对象的Method方法， Method方法的参数Parameter必须是final的。

能够返回到宿主Method执行的某个位置，能够访问该Method函数栈的final变量。

调用匿名对象的getObject() 实质是回调了宿主对象的createBean方法。

个人感觉匿名对象就是一个函数句柄，类似c++中的函数指针。

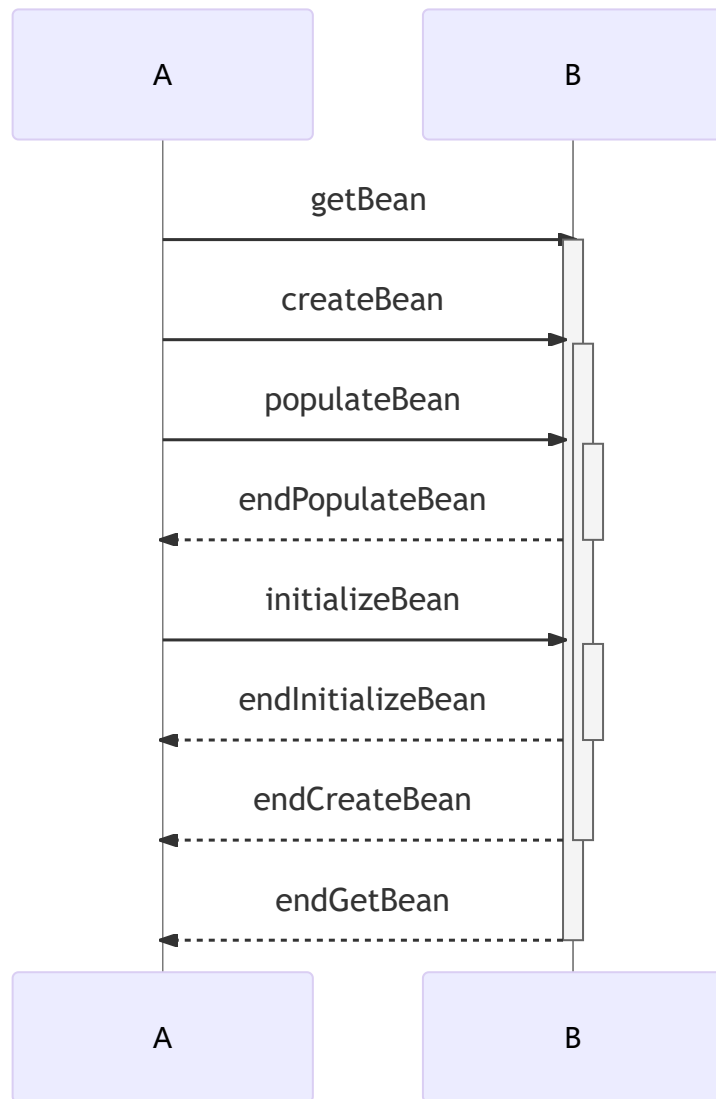
其实可以理解为Method的Local Variable 。 函数变量

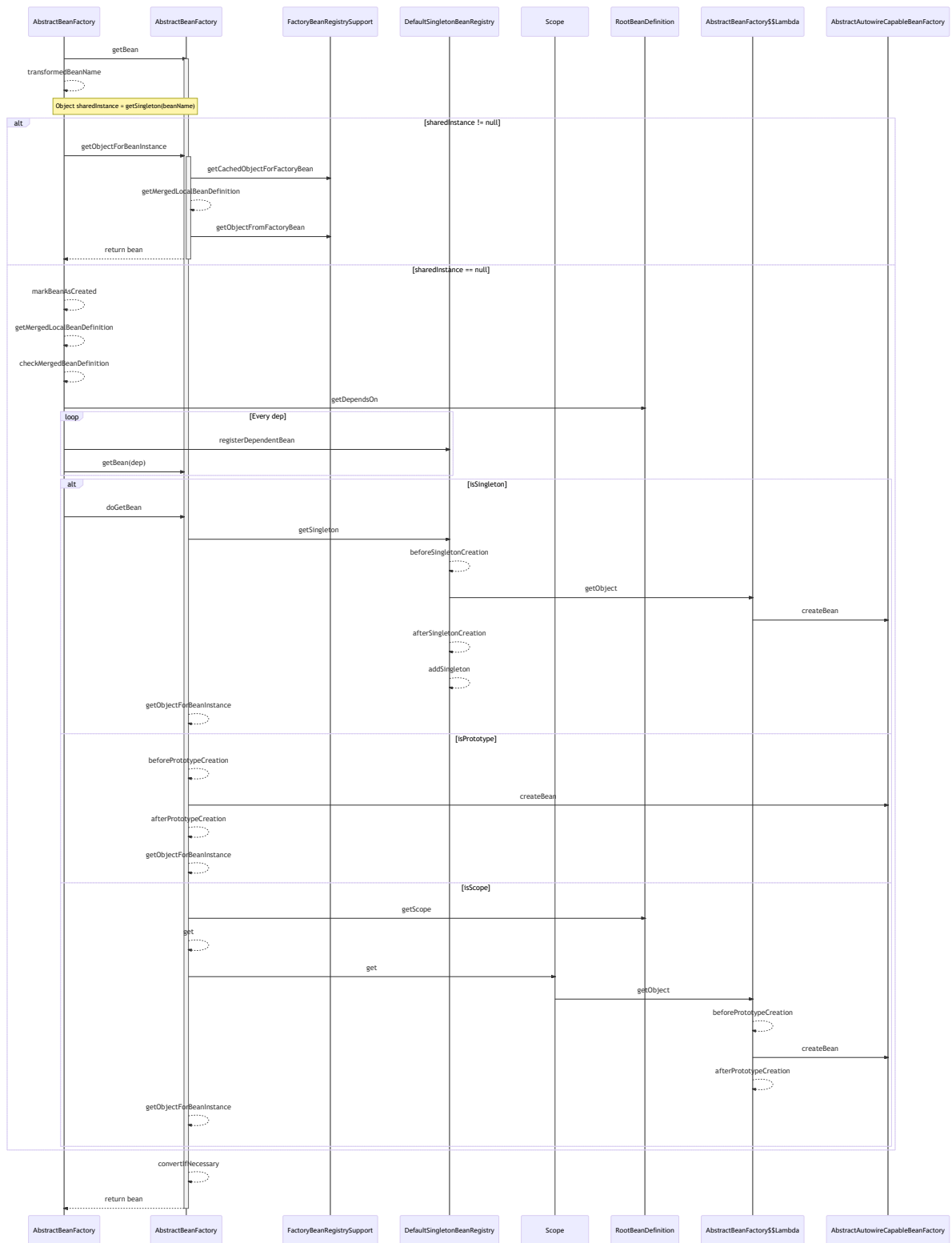
回调点会位于某个函数的某个位置上。

lambda\$doGetBean\$0 代表回调点位置doGetBean方法的第0个回调点。 换句话说，一个Method里面可同时存在多个回调点0, 1, 2, 3, 4. 回调点会按照顺序执行。

spring 会先处理DependsOn依赖。

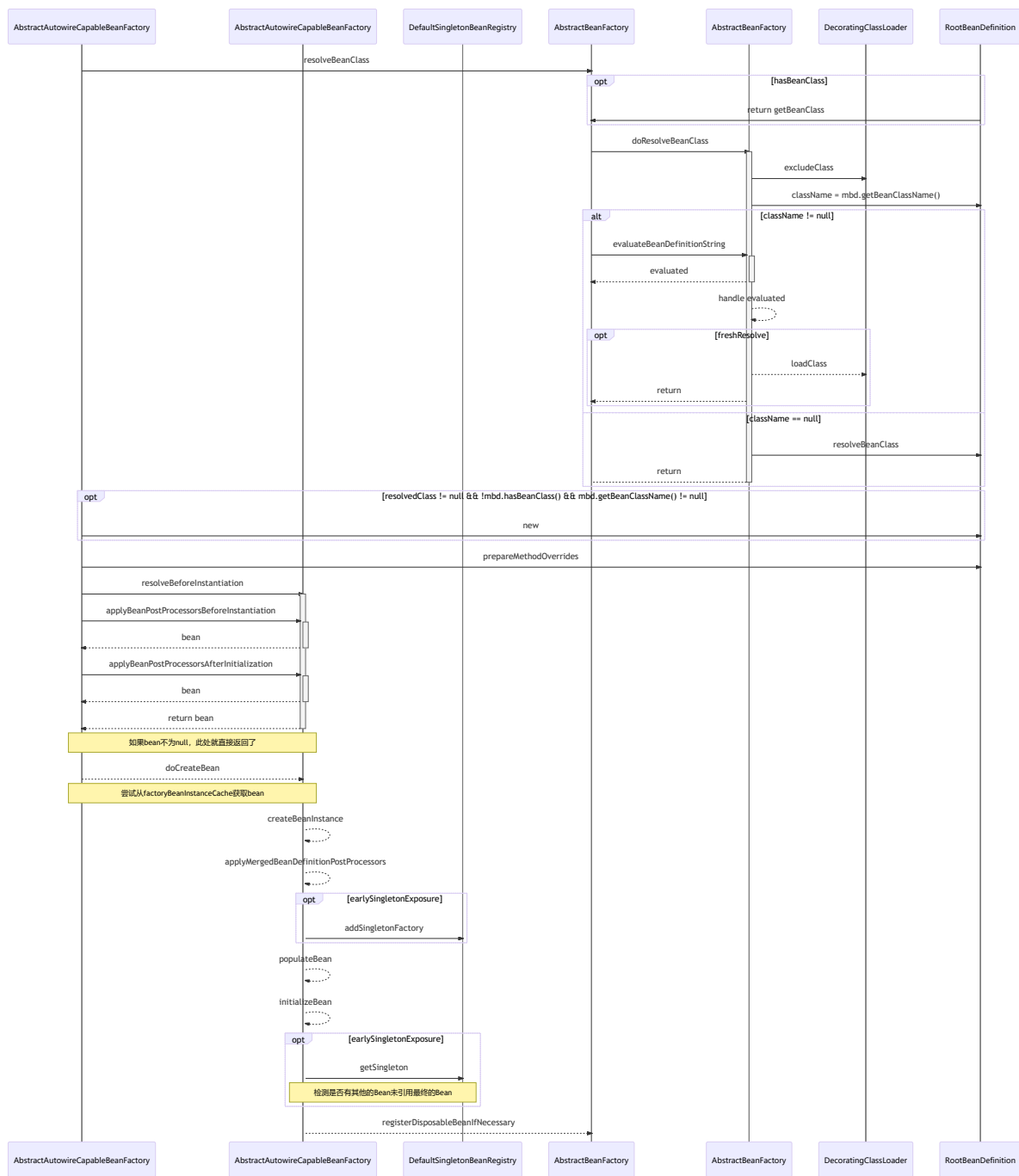
流程分析分若干阶段进行 1: getBean 2: createBean 3: PopulateBean 4: InitializeBean



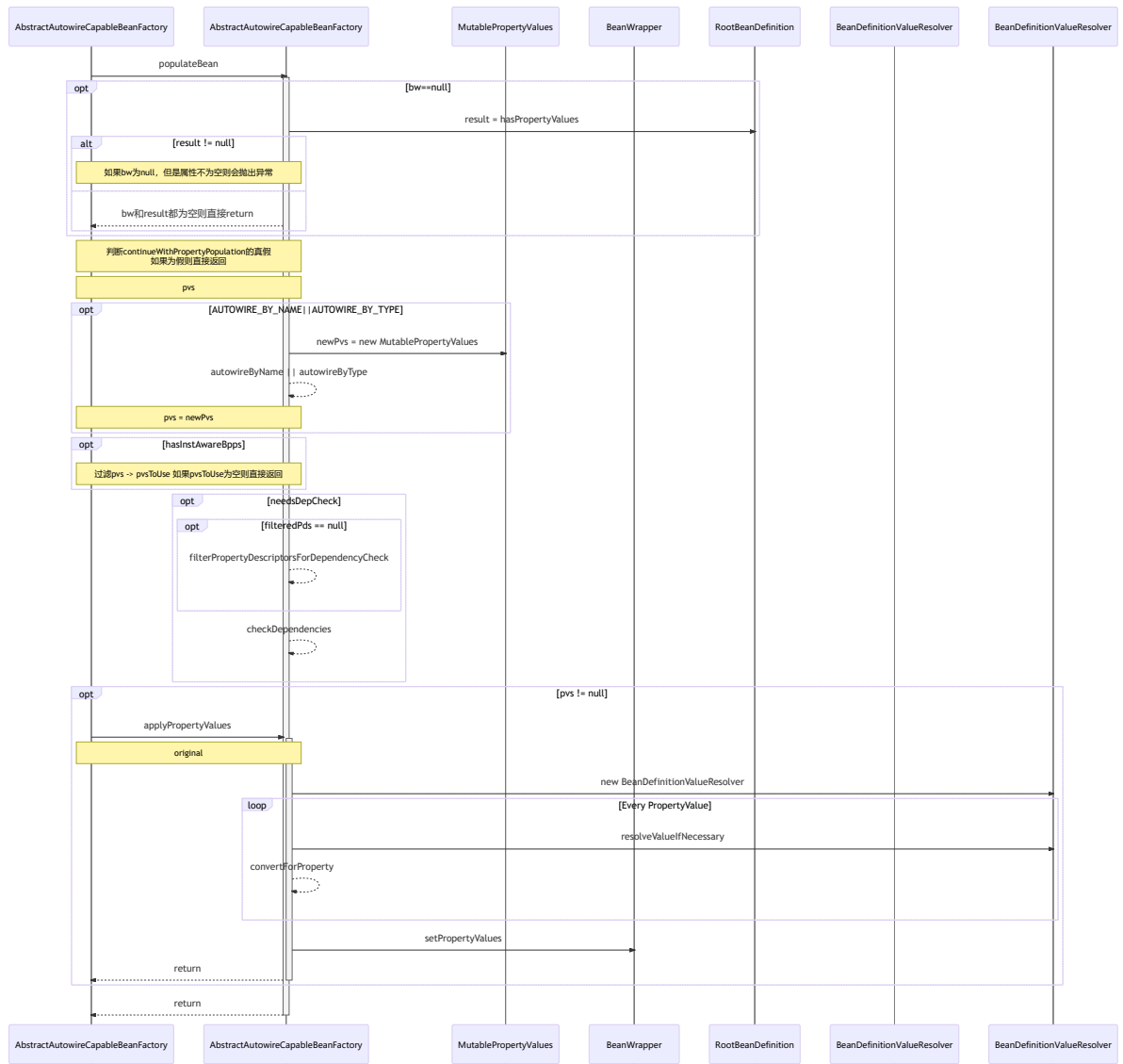


createBean流程分析。其中populateBean和initializeBean两个过程单独分析

maybe opt 可以理解为卫语句。。。



populateBean 流程分析



initializeBean

