**What are the advantages of Java Config? What are the limitations?**

Advantages:

• Regular Java refactoring tooling will work on Java configuration classes.

• Type-checking performed by Java compiler and IDE.

• Can be located in one or a few places.

• Uses Java-code to create and configure Spring beans. Support more complex creation and configuration scenarios.

• Can be used with another Java based languages.

• Separates bean implementation from bean configuration.

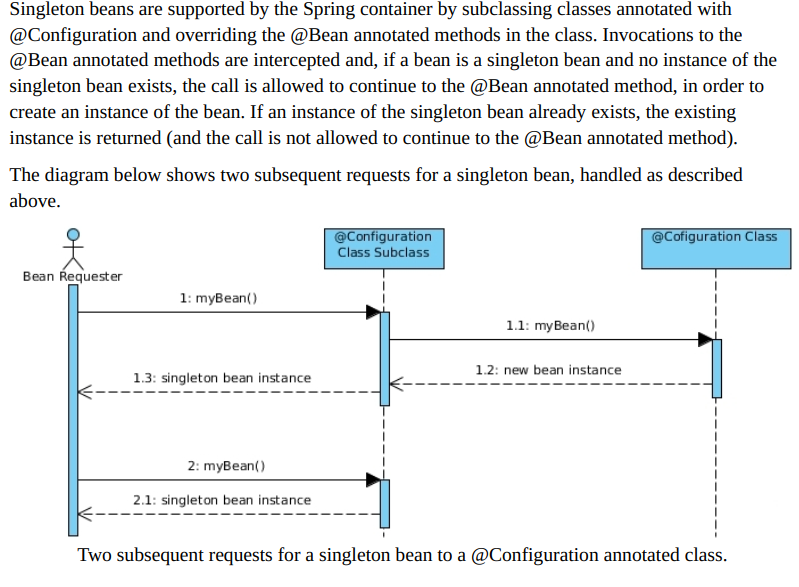
Limitations:

• Configuration classes cannot be final because of using CGLIB proxy.

**Why are you not allowed to annotate a final class with @Configuration?**

The Spring container will create a subclass of each class annotated with @Configuration when creating an application context using CGLIB. Final classes cannot be subclassed, thus classes annotated with @Configuration cannot be declared as final.

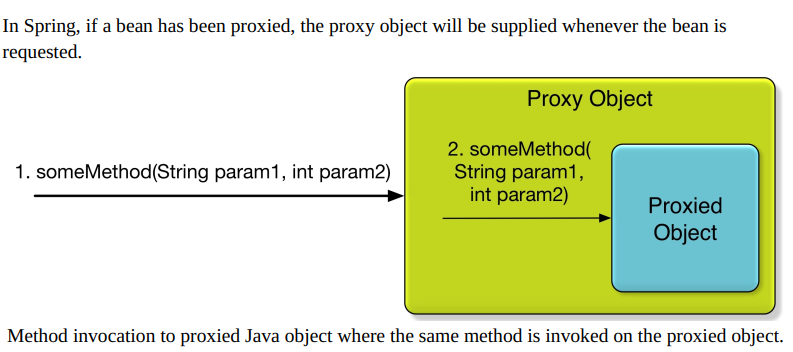
**How do @Configuration annotated classes support singleton beans?**

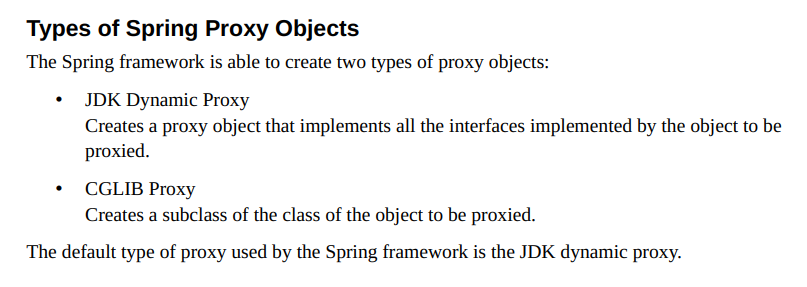


**Why can’t @Bean methods be final either?**

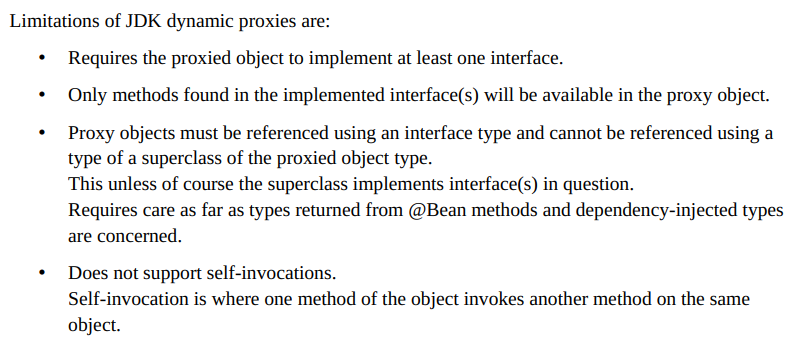
Spring container subclass classes @Configuration classes and overrides the methods annotated with the @Bean annotation, in order to intercept requests for the beans. If the bean is a singleton bean, subsequent requests for the bean will not yield new instances, but the existing instance of the bean.

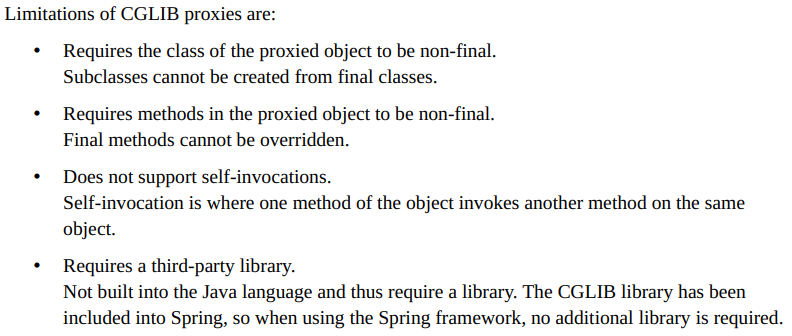
**What is a proxy object and what are the two different types of proxies Spring can create?**



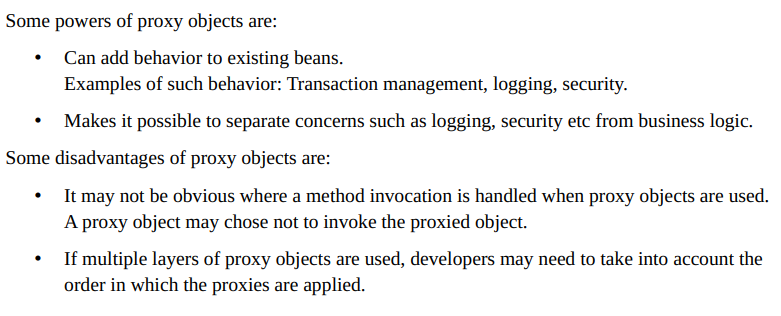


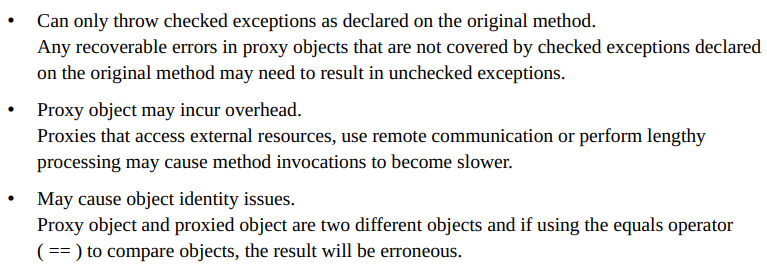
**What are the limitations of these proxies (per type)?**



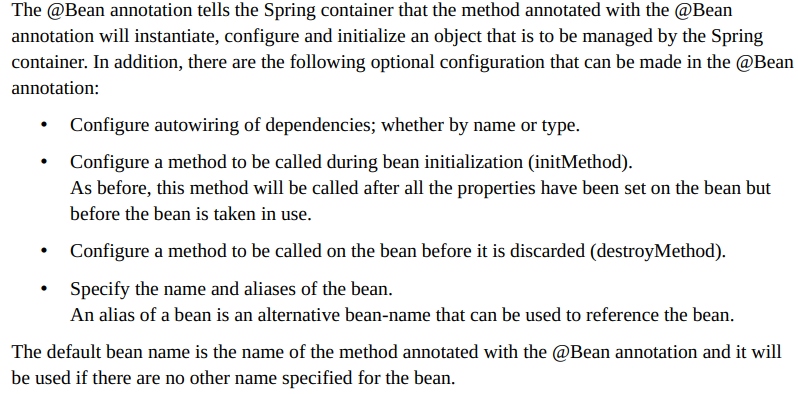


**What is the power of a proxy object and where are the disadvantages?**





**What does the @Bean annotation do?**



<https://docs.spring.io/spring/docs/5.0.3.RELEASE/spring-framework-reference/core.html#beans-java-bean-annotation>

**What is the default bean id if you only use @Bean? How can you override this?**

The default bean name, also called bean id, is the name of the @Bean

annotated method. This default id can be overridden using the name, or its

alias value, attribute of the @Bean annotation

**Why would you define a static @Bean method?**

Static methods do not get proxied. They need to be called before initializing beans, for Example in BeanFactoryPostProcessor.

**@Import and @Configuration using:**

<https://docs.spring.io/spring/docs/5.0.3.RELEASE/spring-framework-reference/core.html#beans-java-configuration-annotation>

<https://docs.spring.io/spring/docs/5.0.3.RELEASE/spring-framework-reference/core.html#beans-java-composing-configuration-classes>

**Why would you use AliasFor annotation?**

AliasForis an annotation that is used to declare aliases for annotation attributes.

<https://docs.spring.io/spring/docs/4.2.0.RC3_to_4.2.0.RELEASE/Spring%20Framework%204.2.0.RELEASE/org/springframework/core/annotation/AliasFor.html>

**Are beans lazily or eagerly instantiated by default? How do you alter this behavior?**

