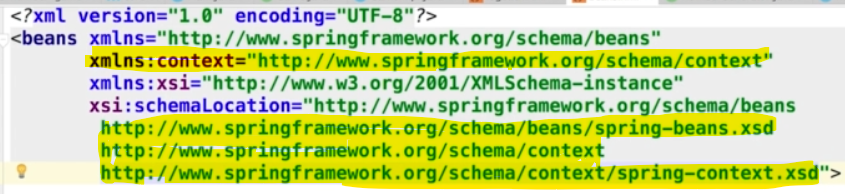
**Annotations using xml file:**

1. To use this, we need to first use this



in the beans.xml file. Since its not configured yet usually by default, we also need to manually specify it like this (Intellij ultimate version auto imported it)



**Component Scan using xml file:**

1. This will add annotation config to our app, but we can use component. Which is a generic type (check on google for more specific types of component).



Here base package we have to define the base package where we want the components. We can define the name in the beans or component.

* There is two way of getting the bean, through type only,



* Or through name and type, 

Both of these have pro and con, for example using name will be problematic since we are writing same name in two places (in beans.xml and in the class Component(“here”)). So, in case we want to change later, we have to change in two places.

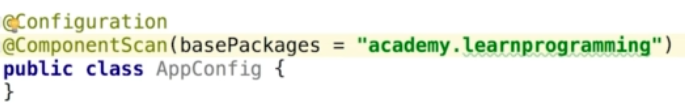
For this reason, we can use the second approach, but this will have problem in some situations as well. For example, if an interface has two or more implementations, then we will get into problems. (in this case, we need to use qualifiers)

**Using a configurations class instead of xml file:**

* **Using Component Scan Annotations**

Instead of using a beans.xml file, we can use a class to have all the configurations.

We need to add annotations for this to the class,



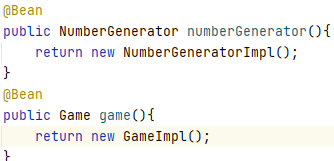
And also use a different implementation of the **ConfigurableApplicationContext** interface, namely **AnnotationConfigApplicationContext**.



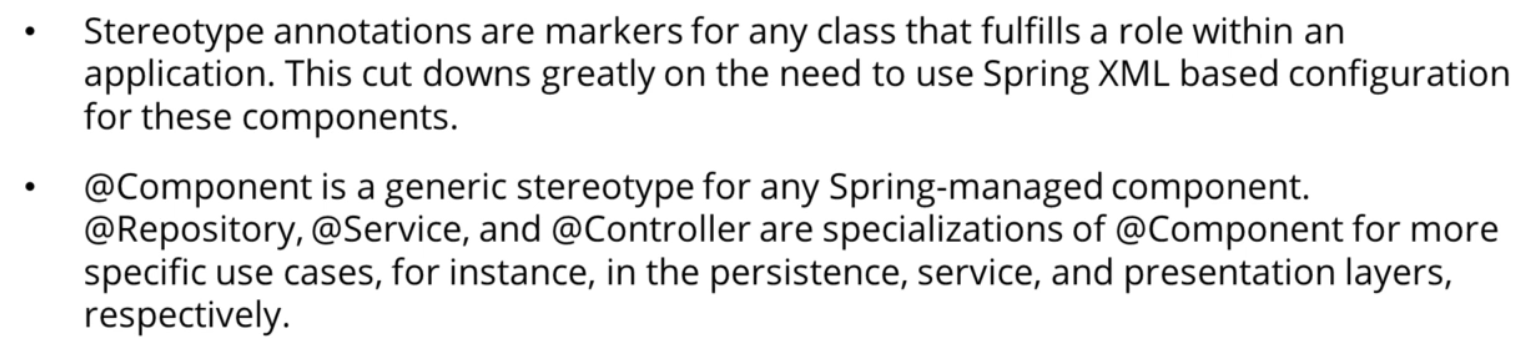
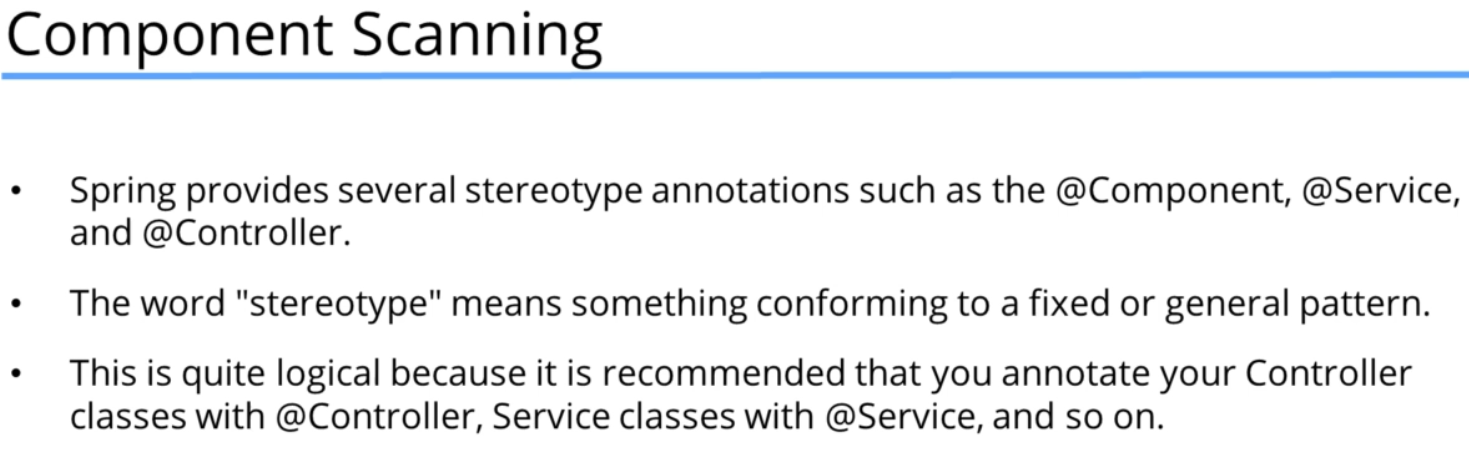
* **Using bean Annotations:**

Here we can remove the component annotations from the classes. 

In this case we can do this,



The bean annotations on both of those methods indicates that, the methods produce a bean that will be managed by the spring container. By default, the name of the bean is the same as the bean method.



`