@Retention(RetentionPolicy.***RUNTIME***)  
@Target(ElementType.***TYPE***)  
@Documented  
@Repeatable(ComponentScans.**class**)  
**public** @**interface** ComponentScan {  
@AliasFor(**"basePackages"**)  
 String[] value() **default** {};  
@AliasFor(**"value"**)  
 String[] basePackages() **default** {};  
Class<?>[] basePackageClasses() **default** {};  
Class<? **extends** BeanNameGenerator> nameGenerator() **default** BeanNameGenerator.**class**;  
Class<? **extends** ScopeMetadataResolver> scopeResolver() **default** AnnotationScopeMetadataResolver.**class**;  
ScopedProxyMode scopedProxy() **default** ScopedProxyMode.***DEFAULT***;  
String resourcePattern() **default** ClassPathScanningCandidateComponentProvider.***DEFAULT\_RESOURCE\_PATTERN***;  
**boolean** useDefaultFilters() **default true**;  
Filter[] includeFilters() **default** {};  
Filter[] excludeFilters() **default** {};  
**boolean** lazyInit() **default false**;  
@Retention(RetentionPolicy.***RUNTIME***)  
 @Target({})  
 @**interface** Filter {  
FilterType type() **default** FilterType.***ANNOTATION***;  
@AliasFor(**"classes"**)  
 Class<?>[] value() **default** {};  
@AliasFor(**"value"**)  
 Class<?>[] classes() **default** {};  
String[] pattern() **default** {};  
  
 }  
  
}

* 包扫描
* includeFilters，包扫描时，包含哪些组件。
* excludeFilters，包扫描时，排除哪些组件。
* useDefaultFilters，使用默认过滤器。使用includeFilters过滤时，需禁用默认过滤器。
* @Repeatable(ComponentScans.**class**)，这是一个重复注解，可以多写几次。也可以用ComponentScans写多个ComponentScan注解。

**public enum** FilterType {  
*按照注解过滤* ***ANNOTATION***,  
*按照给定类型过滤* ***ASSIGNABLE\_TYPE***,  
*按照ASPECTJ表达式过滤* ***ASPECTJ***,  
*按照正则表达式过滤* ***REGEX***,  
*按照自定义规则过滤，需要实现org.springframework.core.type.filter.TypeFilter接口。* ***CUSTOM***}