**COMPONENT SCAN**

**Was ist ein Component?**

Eine Komponent ist verantwortlich für irgendeine Operation.

Das sind die Objects oder “Beans” die von Spring verwaltet werden.

**Wann sollte @Component genutzt werden?**

Wenn immer Klassen verwendet werden, welche keine Services, Controller, Repositories sind und diese Automatisch verknüft verwendet werden möchten.

**Component Scan:**

Beim Starten der Spring App wird das root package und alle darunterliegenden packages nach components durchsucht. Diese components werden dann von Spring verwaltet und befinden sich im Application Context.

**Implizit bei Spring Boot:** Spring Boot vereint die Annotations @Configuration, @ComponentScan, und @EnableAutoConfiguration in der @SpringBootApplication Annotation. Mit dieser Standardeinstellung scannt Spring Boot automatisch nach Komponenten mit den Stereotypen @Component, @Controller, @Service und @Repository im aktuellen Paket (das die @SpringBoot-Hauptklasse enthält) und seinen Unterpaketen.

Eine **explizite Scan Anweisung** erfolgt nur, wenn eine variierende Paketstruktur im Projekt angelegt wurde oder allgemein kein SpringBoot genutzt wird. Dann werden @ComponentScan und @Configuration genutzt um Spring anzuweisen in dem entsprechenden Package, Klassen zu scannen, die mit einer Stereotyp-Annotation annotiert sind.

Beispiel   
(birds package gehört nicht zum root package und wird daher auch nicht per default gescannt. BirdsExplicitScan müsste mit @Configuration markiert und in die SpringBoot Application importiert werden):

|- packagename.componentscan (Main Package)

|- SpringComponentScanningApplication.java

|- packagename.birds

|- BirdsExplicitScan.java

|- Eagle.java

|- Sparrow.java

**Verwendung von Filtern durch Nutzung von @ComponentScan mit Attributen**

Beim Scannen können Filter genutzt werden, welche von Spring zur Verfügung gestellt werden

Durch die Verwendung von Filtern kann die Menge der in Frage kommenden Komponenten eingeschränkt werden.

Beispiele für Attribute:

1. includeFilters(definiert welche Klassen beim Scan inkludiert werden sollen)
2. excludeFilters(definiert welche Klassen beim Scan nicht berücksichitg werden sollen)

Filtertypen:

ANNOTATION: Passt nur auf Klassen mit einer bestimmten Stereotyp-Annotation.

ASPECTJ: Passt auf Klassen mit einem AspectJ-pattern

ASSIGNABLE\_TYPE: Passt auf Klassen, die diese Klasse oder Interface erweitern/implementieren.

REGEX: Findet Klassen, die einen regulären Ausdruck für Package Namen verwenden.

Quellen zum Nachlesen:

<https://www.baeldung.com/spring-component-scanning>

[Component Scanning with Spring Boot - Reflectoring](https://reflectoring.io/spring-component-scanning/)

[Spring Component Scan - DZone Java](https://dzone.com/articles/spring-component-scan)