În Spring Framework, adnotările @Component, @Autowired și @Bean sunt esențiale pentru configurarea și gestionarea *containerului de Spring* (Inversion of Control / Dependency Injection container). Iată ce face fiecare:

**🔹 @Component**

* **Rol:** Marchează o clasă ca fiind un **component Spring**, adică o **componentă gestionată de containerul Spring**.
* **Concret:** Spring va scana automat această clasă (dacă este într-un pachet scanat) și va crea o instanță a ei în contextul aplicației.
* **Exemplu:**
* @Component
* public class MyService {
* // logică
* }

**🔹 @Autowired**

* **Rol:** Realizează **injectarea automată** a unei dependențe (adică injectează o componentă gestionată de Spring într-o altă componentă).
* **Concret:** Spune containerului Spring să găsească și să injecteze automat un bean compatibil în acel câmp, constructor sau metodă.
* **Exemplu:**
* @Component
* public class MyController {
* @Autowired
* private MyService myService;
* }

**🔹 @Bean**

* **Rol:** Definește manual un **bean** (componentă Spring) într-o clasă de configurare (@Configuration).
* **Concret:** Poți folosi această adnotare pentru a crea și înregistra un bean care nu poate fi scanat automat (de ex., o bibliotecă externă).
* **Exemplu:**
* @Configuration
* public class AppConfig {
* @Bean
* public MyService myService() {
* return new MyService();
* }
* }

**Pe scurt:**

| **Adnotare** | **Ce face** | **Unde o folosești** |
| --- | --- | --- |
| @Component | Marchează o clasă ca bean Spring | Deasupra claselor proprii |
| @Autowired | Injectează automat un bean | La câmpuri, constructori sau metode |
| @Bean | Creează un bean manual în clasa de config (@Configuration) | În metode din clase de config |

Vrei și un exemplu complet, ca să vezi cum lucrează toate împreună?

Sigur! Hai să detaliem mai bine ce este un **bean** în Spring și ce face exact adnotarea @Bean.

**🔹 Ce este un *bean* în Spring?**

Un **bean** este pur și simplu un **obiect gestionat de containerul Spring**.

Spring are un mecanism numit **IoC Container** (Inversion of Control) care:

* creează automat instanțele claselor tale (adică "beans"),
* le configurează,
* le injectează acolo unde este nevoie,
* și le gestionează ciclul de viață (lifecycle).

**🔹 Ce face @Bean?**

Adnotarea @Bean este folosită într-o **clasă de configurare** (marcată cu @Configuration) pentru a **crea manual un bean**, adică să spui tu explicit:

“Spring, creează un obiect de acest tip și pune-l în container, pentru ca alte clase să-l poată folosi.”

**🔸 Când folosești @Bean?**

* Când vrei să creezi un obiect care **nu poate fi marcat cu @Component** (ex: clase dintr-o librărie externă).
* Când ai nevoie de **mai mult control asupra modului în care e creat bean-ul** (poate are constructori cu parametri speciali, setări, etc).

**🔸 Exemplu clar:**

Să zicem că ai o clasă NotificationService care NU poate fi adnotată cu @Component (poate vine dintr-o bibliotecă externă):

public class NotificationService {

private String provider;

public NotificationService(String provider) {

this.provider = provider;

}

public void send(String message) {

System.out.println("Sending via " + provider + ": " + message);

}

}

Ca să o transformi într-un bean Spring, faci asta într-o clasă de configurare:

@Configuration

public class AppConfig {

@Bean

public NotificationService notificationService() {

return new NotificationService("Email");

}

}

Acum, oriunde ai nevoie de NotificationService, Spring poate să o injecteze automat:

@Component

public class UserNotifier {

private final NotificationService notificationService;

@Autowired

public UserNotifier(NotificationService notificationService) {

this.notificationService = notificationService;

}

public void notifyUser(String msg) {

notificationService.send(msg);

}

}

**🔹 Diferența față de @Component**

| **@Bean** | **@Component** |
| --- | --- |
| Declarat manual, într-o metodă | Automat, prin adnotarea clasei |
| Se folosește pentru clase externe / personalizare | Se folosește pentru clase din codul tău |
| Ai control complet asupra modului de creare | Instanța e creată automat de Spring |

Vrei să-ți arăt și cum arată într-un proiect Spring Boot real, ca exemplu complet?