

אפשרויות עם בנאים:

בכל מחלקה עם @Component אפשר להגדיר כמה בנאים שרוצים

- בנאי אחד עם פרמטרים: מקבל autowired אוטומטית (spring תשתמש בו ותזריק)
-
- בנאי אחד ללא פרמטרים ועוד בנאים עם פרמטרים אבל ללא autowired - ספרינג תשתמש בבנאי שלא מקבל פרטמרים.
-
- בנאי אחד ללא פרמטרים ובנאי עם פרמטרים ועם autowired - ספרינג תשתמש בבנאי שמקבל פרמטרים ותזריק. (מותר בנוסף לתת autowired על השדות)
-
- מספר בנאים שמקבלים פרמטרים אבל רק אחד עם autowired - ספרינג תשתמש בבנאים עם autowired ותזריק.
-
- מספר בנאים שמקבלים פרמטרים אבל ללא autowired - ספרינג תזרוק שגיאה שאין בנאי ברירת מחדל.
-
- אסור יותר מבנאי אחד עם autowired.

אסטרטגיה לביצוע Autowired

- אם יש התאמה אחת בלבד (לפי type) אז מוזרק
- אם יש יותר מאחת אז לפי Primary אם יש
- אם יש כמה אפשרויות ואין Primary אז לפי Qualifier
- לפי שם השדה

Annotations

@ComponentScan

שמים על class למטרת קונפיגורציה, כדי להגדיר סריקת מחלקות כדי לאתר הגדרות של beans

@Configuration

שמים על class למטרת קונפיגורציה, כדי להגדיר שה class יכול להכיל bean methods

@PropertySource

שמים על class למטרת קונפיגורציה, כדי להגדיר מיקום לקובץ properties

@Component

שמים על מחלקה כדי להגדיר bean מהסוג של המחלקה

@Bean

שמים על מתודה כדי להגדיר bean מהסוג שהמתודה מחזירה (בתוך class שמכיל אנוטציית Configuration)

@Scope("prototype")

@Scope("singleton")

קונפיגורציה להגדרת bean – האם יהיה סינגלטון או פרוטוטיפ

@Autowired

הוראה להזריק bean שנדרש ל bean הנוכחי:

1. CTOR

2. field

3. setter

@Qualifier

מצורף בהקשר שלAutowired כשיש יותר מהתאמה אחת ורוצים לציין איזה להזריק

@Primary

מצרפים להגדרה של bean כדי לקבוע שהוא ההתאמה העיקרית במקרה שיש כמה אפשרויות.

@Value

מצרפים לשדה או פרמטר בבנאי או מתודה כדי להזריק ערך מתוך קובץ properties

@Lazy

שמים על הגדרה של bean שהוגדר כ singleton על מנת לשנות את ההתנהגות ברירת מחדל של טעינה Eager. כלומר ה bean לא ייוצר עד שמישהו מבקש אותו. לא רלוונטי ל prototype (תמיד lazy)