

# Spring Framework

.....

# מה זה Spring Framework?

- Spring Framework הוא קבוצה של פרויקטים פתוחים

- Spring מאפשר :

- להקל על פיתוח יישומים Java באמצעות הפשטת והפחתת התלות בין רכיבי הקוד

- גישה למסד נתונים Spring מספק גישה נוחה למסדי נתונים אוטומציה של יצירת אובייקטים וחיבור למסד נתונים ( לדוגמה הוא חוסך שאילתות sql במקומן משתמשים בפונקציות שמורות ( find() findAll() )

- Inversion of Control (IoC) כל הנושא של הזרקת תלויות (עם ה @Autowired)

- Aspect-Oriented Programming (AOP) הפרדת לוגיקה מצליחה מהלוגיקה העיקרית. לדוגמא:

```
@Aspect
@Component
public class LoggingAspect {

    @Before("execution(* com.example.service.*(..))")
    public void logBefore(JoinPoint joinPoint) {
        System.out.println("Entering method: " + joinPoint.getSignature().getName());
    }
}
```

```
@Service
public class MyService {

    public String doSomething() {
        System.out.println("Doing something...");
        return "Done";
    }
}
```

# Annotation (אנוטציה)

## מהי אנוטציה?

אנוטציה היא תווית מיוחדת שמוצמדת למרכיבים שונים בקוד כמו מחלקות, שיטות, משתנים, פרמטרים ועוד. היא מוסיפה מידע לקוד ומאפשרת לספרינג להבין איך לנהוג עם הקוד שלך היא מתחילה בסימן @ ואחריו שם האנוטציה.

בספרינג משתמשים בה בעיקר ל:

1. הזרקת תלויות (@Autowired-Dependency Injection)
2. סימון קומפוננטות (@Service-Component Scanning @Repository)
3. טיפול בהטיות (Aspect-Oriented Programming – AOP) @Before @After @Aspect
4. הגדרת תצורת קונפיגורציה @Bean @Configuration

## Annotation שימושיים:

- @Repository - מסמן מחלקה כ-Repository המהווה שכבת גישה לנתונים.
- @Service - מסמן מחלקה כשירות, המייצגת את הלוגיקה העסקית
- @Controller - מסמן מחלקה כ-Controller האחראית על טיפול בבקשות HTTP.
- @RequestMapping - מגדיר את המיפוי של הבקשות ל-Endpoints
- @GetMapping, @PostMapping, @PutMapping, @DeleteMapping - מסמנים סוגי בקשות HTTP
- @PathVariable - מאפשר לחלץ נתונים מה URL של הבקשה
- @RequestBody - מאפשר לקרוא נתונים מהגוף של הבקשה ולהמיר אותם לאובייקטים של Java
- @Value - מאפשר קביעת ערכים מתוך קובץ properties

•  
@Configuration - מסמן שהמחלקה היא מחלקת קונפיגורציה.

@Bean - מסמן שיטה שמחזירה אובייקט ש-Spring צריך ליצור ולנהל.

דוגמת קוד:

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;

@Configuration
public class AppConfig {

    @Bean
    public MyService myService() {
        return new MyServiceImpl();
    }
}
```

## דוגמאות לשימוש ב:

@PostMapping, @PutMapping, @GetMapping, @PathVariable, @RequestBody, @RestController, @RequestMapping, @Autowired

```
package com.example.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;
import com.example.service.UserService;
import com.example.model.User;

import java.util.List;

@RestController
@RequestMapping("/api/users")
public class UserController {

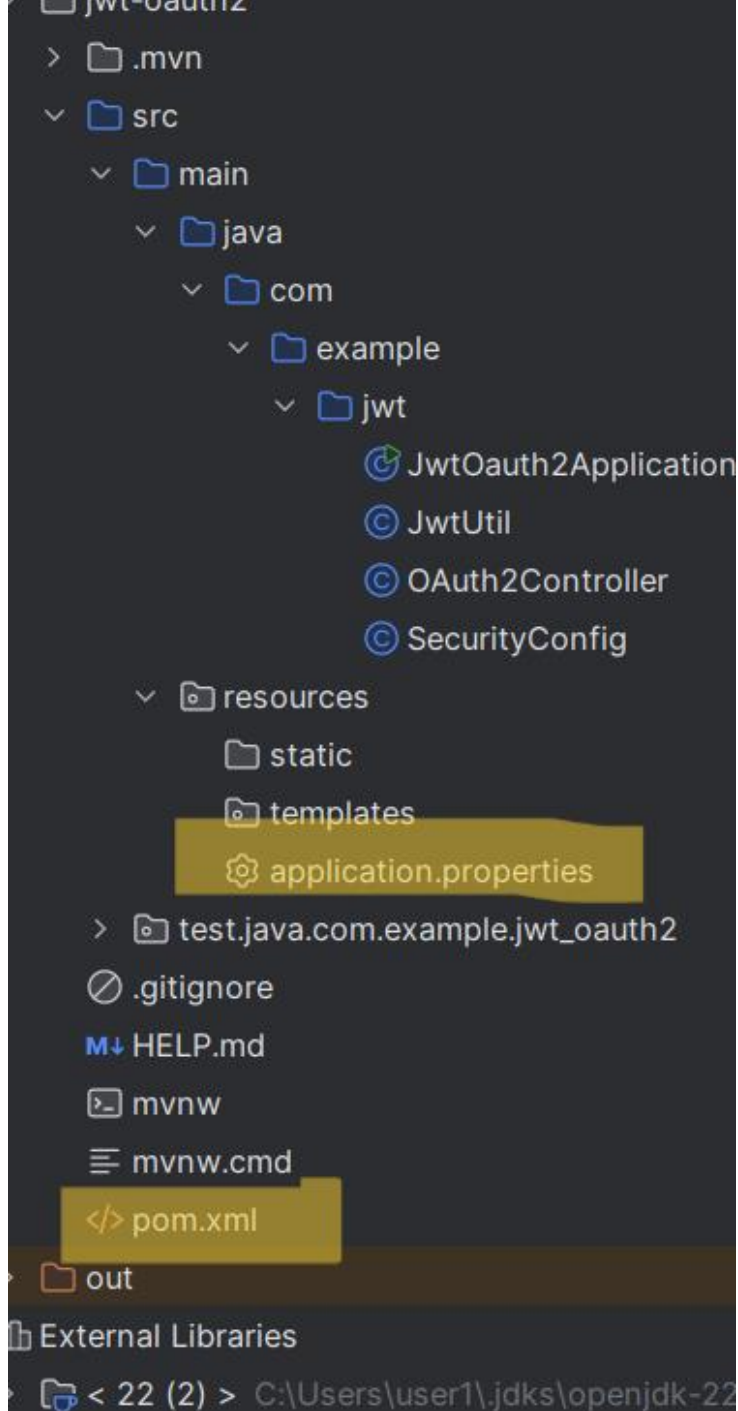
    private final UserService userService;

    @Autowired
    public UserController(UserService userService) {
        this.userService = userService;
    }
}
```

```
@GetMapping("/{id}")
public User getUserById(@PathVariable Long id) {
    return userService.getUserById(id);
}

@PostMapping
public User createUser(@RequestBody User user) {
    return userService.saveUser(user);
}

@PutMapping("/{id}")
public User updateUser(@PathVariable Long id, @RequestBody User userDetails) {
    User user = userService.getUserById(id);
    if (user != null) {
        user.setName(userDetails.getName());
        user.setEmail(userDetails.getEmail());
        return userService.saveUser(user);
    }
    return null;
}
```



## קובץ pom.xml: (הוא חופף ברעיון שלו לקובץ package.json בריאקט)

קובץ של ניהול התלויות של Maven הוא מכיל מידע על הפרוייקט ועל התלויות הדרושות לו כך ש Maven תוכל להוריד ולנהל אותם באופן אוטומטי לדוגמא תלות בספריית h2(DB):

```
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
```

## קובץ application.properties:

משמש לקביעת תצורה (configuration) של האפליקציה ב Spring  
הקובץ מכיל הגדרות שונות כמו הגדרות מסד נתונים, הגדרות שרת, תכונות שונות של האפליקציה ועוד

```
# הגדרות מסד נתונים
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=password

# הגדרות שרת
server.port=8080

# הגדרות אפליקציה
spring.application.name=MySpringApp
```

בדוגמא הזו קובץ ה  
:application.properties

1. קובע את פרטי ההתחברות למסד הנתונים.
2. מגדיר את הפורט שעליו האפליקציה תפעל
3. מגדיר את שם האפליקציה