

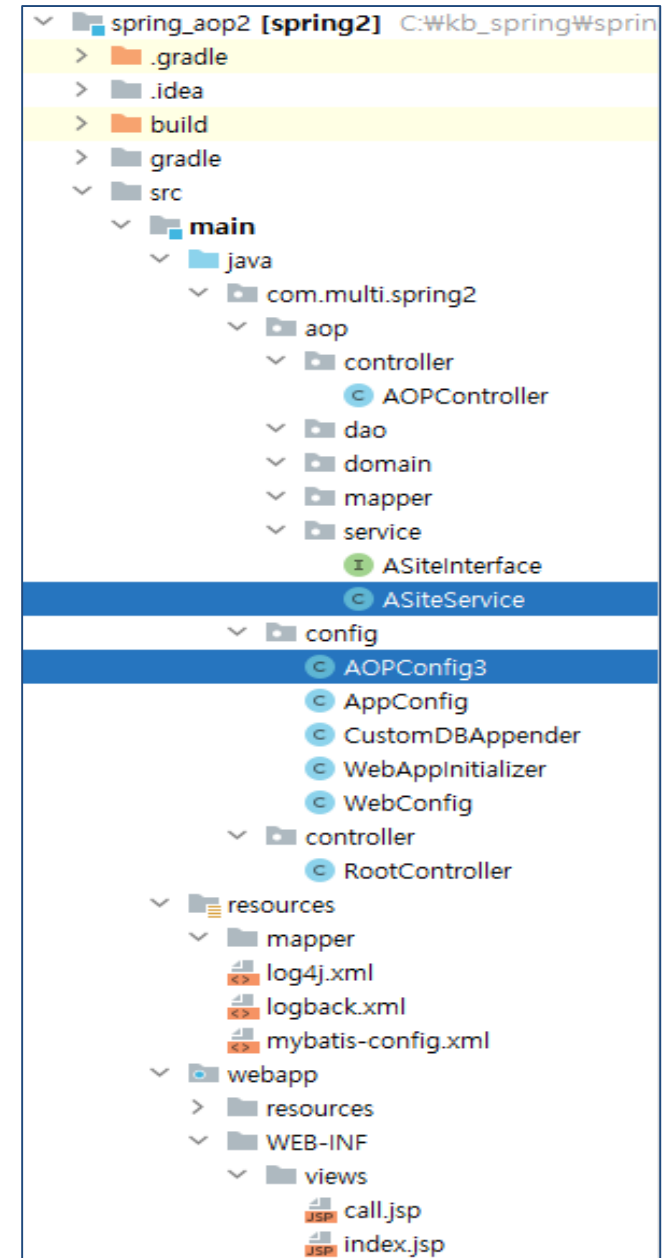
2024년 상반기 K-디지털 트레이닝

SPRING AOP

[KB] IT's Your Life

Q1 - SPRING

- 프로젝트명: **spring_aop2**, 패키지명: **com.multi.spring2**
- [spring_aop2-문제.zip](#) 프로젝트 파일을 다운로드(클릭) 받아 본인의 설정을 변경한 후, 프로젝트를 완성하시오.
- AOP를 이용하여 다음 2가지 작업의 성능을 체크할 수 있는 프로그래밍을 완성하시오.
 - **task1** : 랜덤한 값 만들어 배열에 넣은 후, 꺼내어 모두 더하여 출력하는 비즈니스 로직
 - **task2** : 문자열 내 특정 단어 빈도 계산하여 가장 많이 출현한 단어와 그 빈도수를 출력하는 비즈니스 로직
 - 성능 체크 내용 : 각 메서드의 실행 시간 출력
- ASiteService.java, AOPConfig3.java파일 내 <주석 처리된 부분>을 구현하시오.



- ASiteService.java

```
@Component
public class ASiteService implements ASiteInterface {
    @Override
    public void task1() {
        // 실제 비즈니스 로직이 여기에 위치합니다.

        int[] numbers = new int[1000];
        // 랜덤한 값 만들어 배열에 넣은 후, 꺼내어 모두 더하여 출력하는 비즈니스 로직

        /*

        코드 작성

        */

        try {
            Thread.sleep(3000); // 2초간 대기
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

- ASiteService.java

```
@Override
public void task2() {
    // 실제 비즈니스 로직이 여기에 위치합니다.
    // 큰 텍스트 문자열
    String largeText = "Lorem ipsum dolor sit amet, consectetur adipiscing elit. "
        + "Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. "
        + "Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. "
        + "Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. "
        + "Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. "
        + "Lorem ipsum dolor sit amet, consectetur adipiscing elit. ";

    // 특정 단어의 출현 빈도를 계산하기 위한 맵
    Map<String, Integer> wordCount = new HashMap<>();

    // 문자열 내 특정 단어 빈도 계산하여 가장 많이 출현한 단어와 그 빈도수를 출력하는 코드 작성

    /*

    코드 작성

    */

    try {
        Thread.sleep(1000); // 1초간 대기
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

- AOPConfig3.java

```
@Configuration
@EnableAspectJAutoProxy(proxyTargetClass = true)
public class AOPConfig3 {

    public AOPConfig3(){
        System.out.println("AppConfig3 created");
    }

    @Bean
    public AOPConfig3.AspectClass3 aspect() {
        return new AOPConfig3.AspectClass3();
    }

    @Aspect
    public static class AspectClass3 {

        @Around("execution(* com.multi.spring2.aop.service.ASiteService.task*(..))")
        public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {
            /*
            코드 작성
            */
        }
    }
}
```

- AOPController.java

```
@Controller
public class AOPController {

    ASiteInterface aSiteService;

    @Autowired
    public AOPController(ASiteInterface aSiteService) {
        this.aSiteService = aSiteService;
    }

    @RequestMapping("call.do")
    public void call() {
        aSiteService.task1();
        aSiteService.task1();
        aSiteService.task1();
        aSiteService.task1();
        aSiteService.task1();
        aSiteService.task2();
        aSiteService.task2();
        aSiteService.task2();
        aSiteService.task2();
        aSiteService.task2();
    }
}
```

수고하셨습니다!

