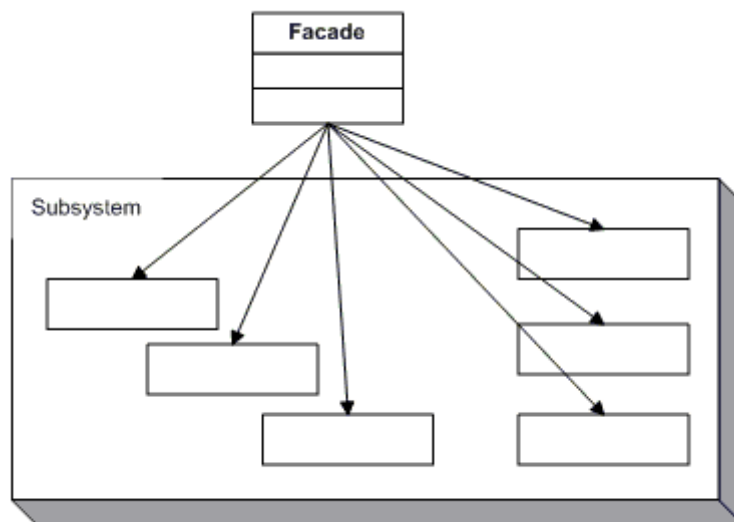


(1) 프로그램 로그 남기기

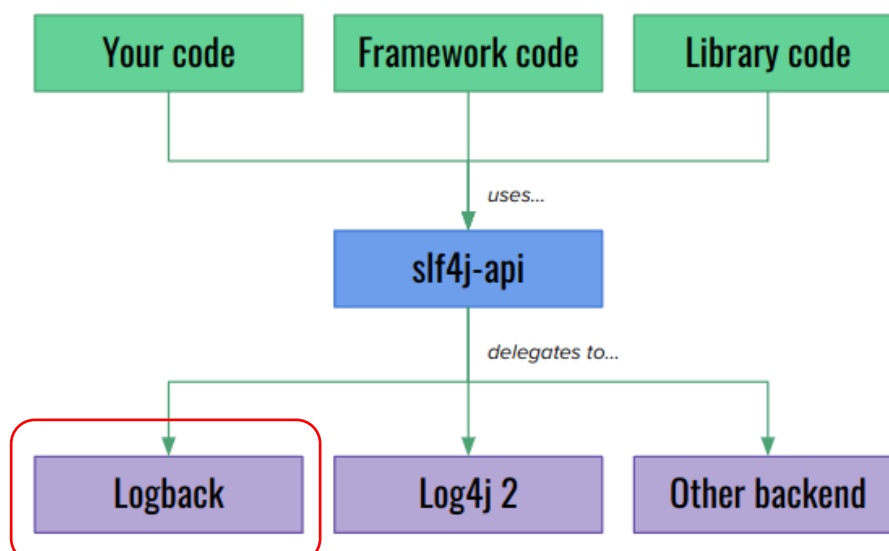
퍼사드 패턴(외관 패턴)은 소프트웨어 공학 디자인 패턴 중 하나이다. 객체 지향 프로그래밍 분야에서 자주 쓰인다.

퍼사드 패턴이란 복잡한 서브 시스템 의존성을 최소화하는 패턴이다. 서브 시스템에 있는 인터페이스들에 대한 **통합 인터페이스**를 제공하여 서브 시스템을 더 쉽게 그리고 일관성 있게 사용할 수 있도록 만드는 **더 높은 수준의 인터페이스**를 말한다.

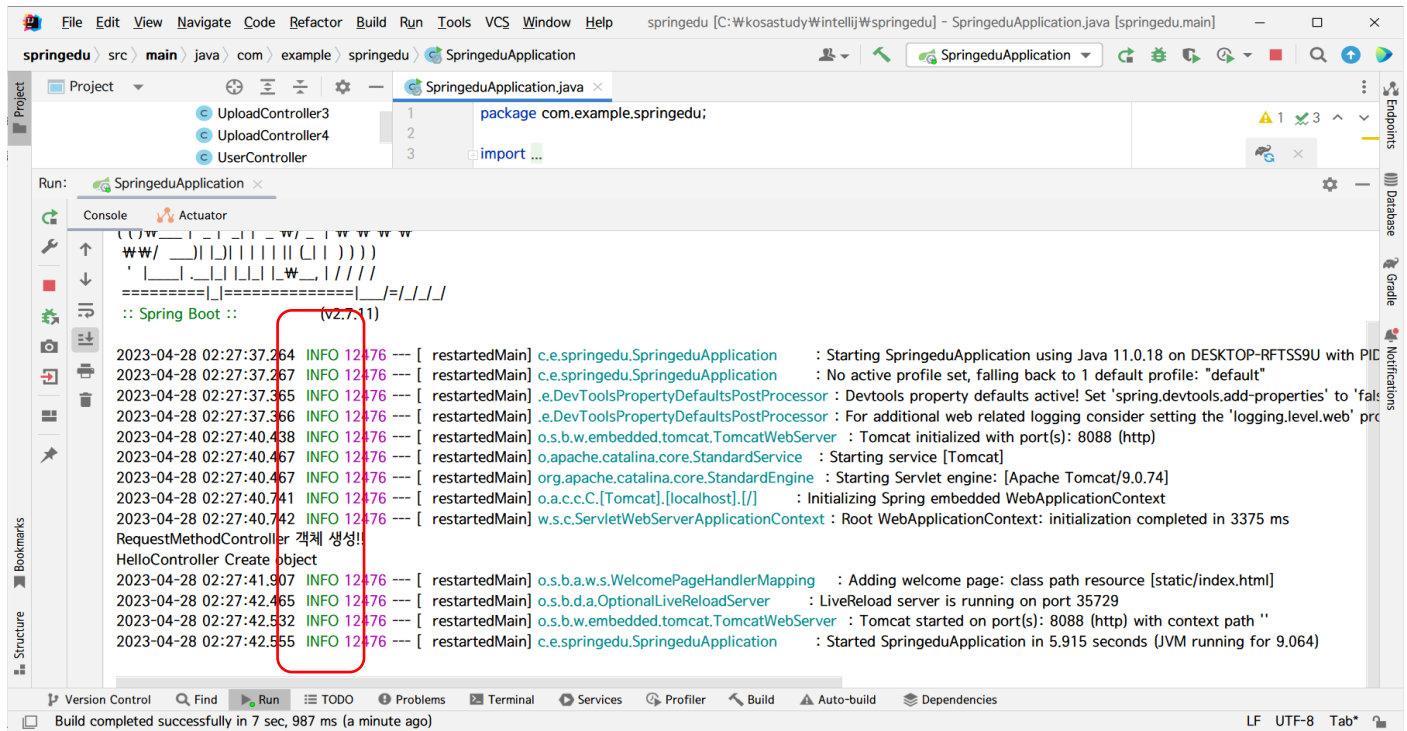


[SLF4J(Simple Logging Facade for Java)]

SLF4J는 단순한 퍼사드 패턴을 수단으로 하는 **자바 로깅 API**를 제공한다. 즉 **여러 다양한 로그 라이브러리들의 통합된 인터페이스를 제공**하는 것이 바로 SLF4J 이다.



스프링 부트에서는 **스프링 부트 로깅 라이브러리 spring-boot-starter-logging**가 함께 포함되며 이 안에 SLF4J 와 Logback 이 포함된다. 스프링은 기동될 때 INFO 레벨의 로그들이 콘솔창에 출력된다.

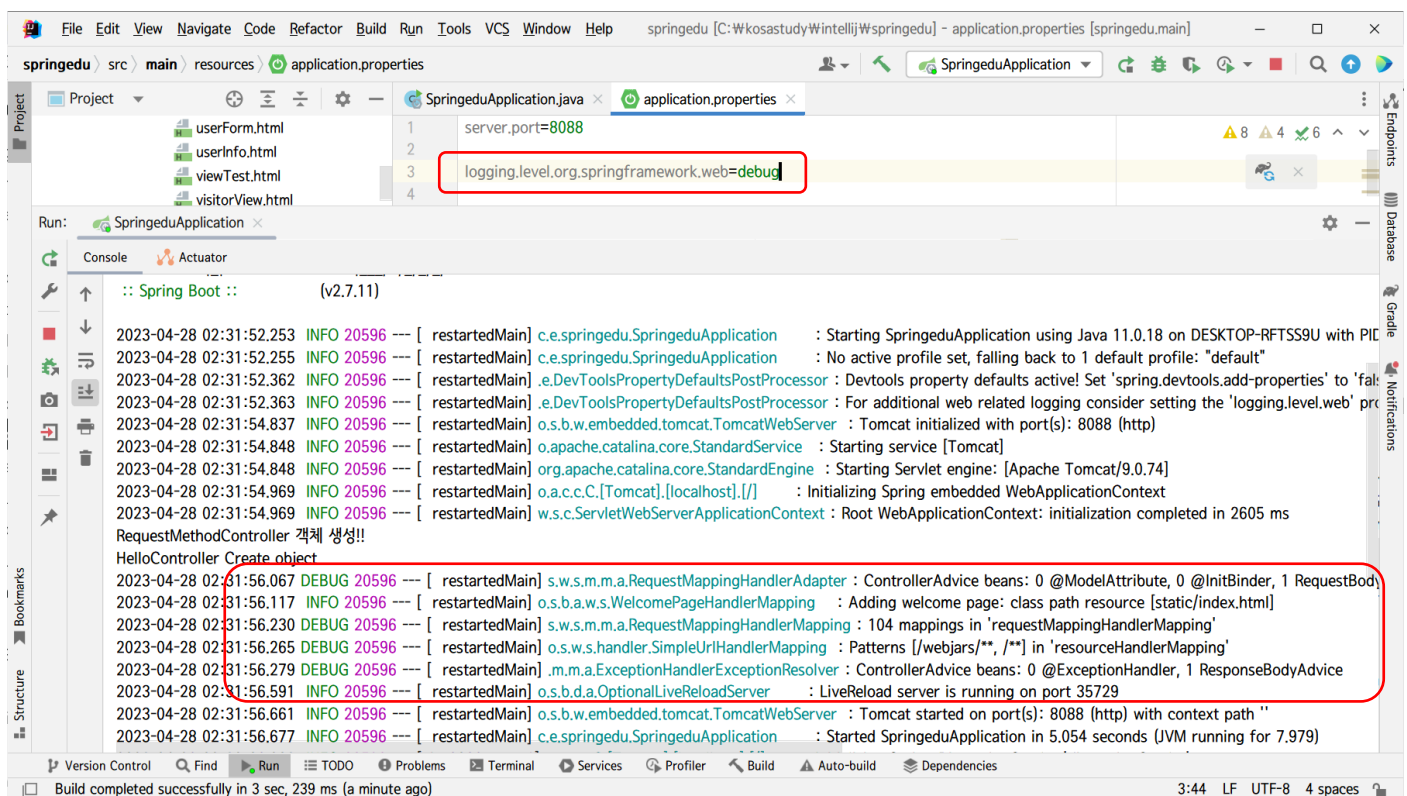


```
springedu [C:\Wkosastudy\Wintellij\Wspringedu] - SpringeduApplication.java [springedu.main]
package com.example.springedu;
import ...

:: Spring Boot :: (v2.7.11)

2023-04-28 02:27:37.264 INFO 12476 --- [ restartedMain] c.e.springedu.SpringeduApplication : Starting SpringeduApplication using Java 11.0.18 on DESKTOP-RFTSS9U with PID
2023-04-28 02:27:37.267 INFO 12476 --- [ restartedMain] c.e.springedu.SpringeduApplication : No active profile set, falling back to 1 default profile: "default"
2023-04-28 02:27:37.365 INFO 12476 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false'
2023-04-28 02:27:37.366 INFO 12476 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property
2023-04-28 02:27:40.438 INFO 12476 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8088 (http)
2023-04-28 02:27:40.467 INFO 12476 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-04-28 02:27:40.467 INFO 12476 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.74]
2023-04-28 02:27:40.741 INFO 12476 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-04-28 02:27:40.742 INFO 12476 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 3375 ms
RequestMethodController 객체 생성!!
HelloController Create object
2023-04-28 02:27:41.907 INFO 12476 --- [ restartedMain] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]
2023-04-28 02:27:42.465 INFO 12476 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-04-28 02:27:42.532 INFO 12476 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8088 (http) with context path ""
2023-04-28 02:27:42.555 INFO 12476 --- [ restartedMain] c.e.springedu.SpringeduApplication : Started SpringeduApplication in 5.915 seconds (JVM running for 9.064)
```

application.properties 파일에 스프링의 로그레벨과 관련한 내용을 설정할 수 있다. 다음은 logging.level.org.springframework.web=debug 를 설정하여 웹과 관련된 처리에서는 DEBUG 레벨의 로그를 남기도록 설정한 후에 스프링 부트 애플리케이션을 기동시는 화면이다. 웹과 관련해서는 INFO 외에도 DEBUG 레벨(조금 더 자세한)이 출력된 것을 볼 수 있다.



```
springedu [C:\Wkosastudy\Wintellij\Wspringedu] - application.properties [springedu.main]
server.port=8088
logging.level.org.springframework.web=debug

:: Spring Boot :: (v2.7.11)

2023-04-28 02:31:52.253 INFO 20596 --- [ restartedMain] c.e.springedu.SpringeduApplication : Starting SpringeduApplication using Java 11.0.18 on DESKTOP-RFTSS9U with PID
2023-04-28 02:31:52.255 INFO 20596 --- [ restartedMain] c.e.springedu.SpringeduApplication : No active profile set, falling back to 1 default profile: "default"
2023-04-28 02:31:52.362 INFO 20596 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : Devtools property defaults active! Set 'spring.devtools.add-properties' to 'false'
2023-04-28 02:31:52.363 INFO 20596 --- [ restartedMain] .e.DevToolsPropertyDefaultsPostProcessor : For additional web related logging consider setting the 'logging.level.web' property
2023-04-28 02:31:54.848 INFO 20596 --- [ restartedMain] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2023-04-28 02:31:54.848 INFO 20596 --- [ restartedMain] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.74]
2023-04-28 02:31:54.969 INFO 20596 --- [ restartedMain] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2023-04-28 02:31:54.969 INFO 20596 --- [ restartedMain] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2605 ms
RequestMethodController 객체 생성!!
HelloController Create object
2023-04-28 02:31:56.067 DEBUG 20596 --- [ restartedMain] s.w.s.m.m.a.RequestMappingHandlerAdapter : ControllerAdvice beans: 0 @ModelAttribute, 0 @InitBinder, 1 RequestBodyAdvice
2023-04-28 02:31:56.117 INFO 20596 --- [ restartedMain] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]
2023-04-28 02:31:56.230 DEBUG 20596 --- [ restartedMain] s.w.s.m.m.a.RequestMappingHandlerMapping : 104 mappings in 'requestMappingHandlerMapping'
2023-04-28 02:31:56.265 DEBUG 20596 --- [ restartedMain] o.s.w.s.handler.SimpleUrlHandlerMapping : Patterns [/webjars/**, /**] in 'resourceHandlerMapping'
2023-04-28 02:31:56.279 DEBUG 20596 --- [ restartedMain] .m.m.a.ExceptionHandlerExceptionResolver : ControllerAdvice beans: 0 @ExceptionHandler, 1 ResponseBodyAdvice
2023-04-28 02:31:56.591 INFO 20596 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2023-04-28 02:31:56.661 INFO 20596 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8088 (http) with context path ""
2023-04-28 02:31:56.677 INFO 20596 --- [ restartedMain] c.e.springedu.SpringeduApplication : Started SpringeduApplication in 5.054 seconds (JVM running for 7.979)
```

[로그를 남기는 컨트롤러 예제]

```
import lombok.extern.slf4j.Slf4j;
```

```
@Controller
```

```
@Slf4j
```

```
public class LogTestController1 {
```

```
    @RequestMapping("/log1")
```

```
    public ModelAndView xxx(){
```

```
        log.error("error-로그를 테스트합니다!"); // 에러가 발생했을 때 남기는 로그
```

```
        log.warn("warn-로그를 테스트합니다!"); // 경고에러가 발생했을 때 남기는 로그
```

```
        log.info("info-로그를 테스트합니다!"); // 수행 흐름에 대한 정보를 남기는 로그
```

```
        log.debug("debug-로그를 테스트합니다!"); // 개발자에게 필요한 정보를 남기는 로그
```

```
        log.trace("trace-로그를 테스트합니다!"); // 자세한 수행 흐름을 남기는 로그
```

```
        ModelAndView mav = new ModelAndView();
```

```
        mav.setViewName("logView");
```

```
        mav.addObject("msg", "톰캣콘솔창에서 확인하세요!!");
```

```
        return mav;
```

```
    }
```

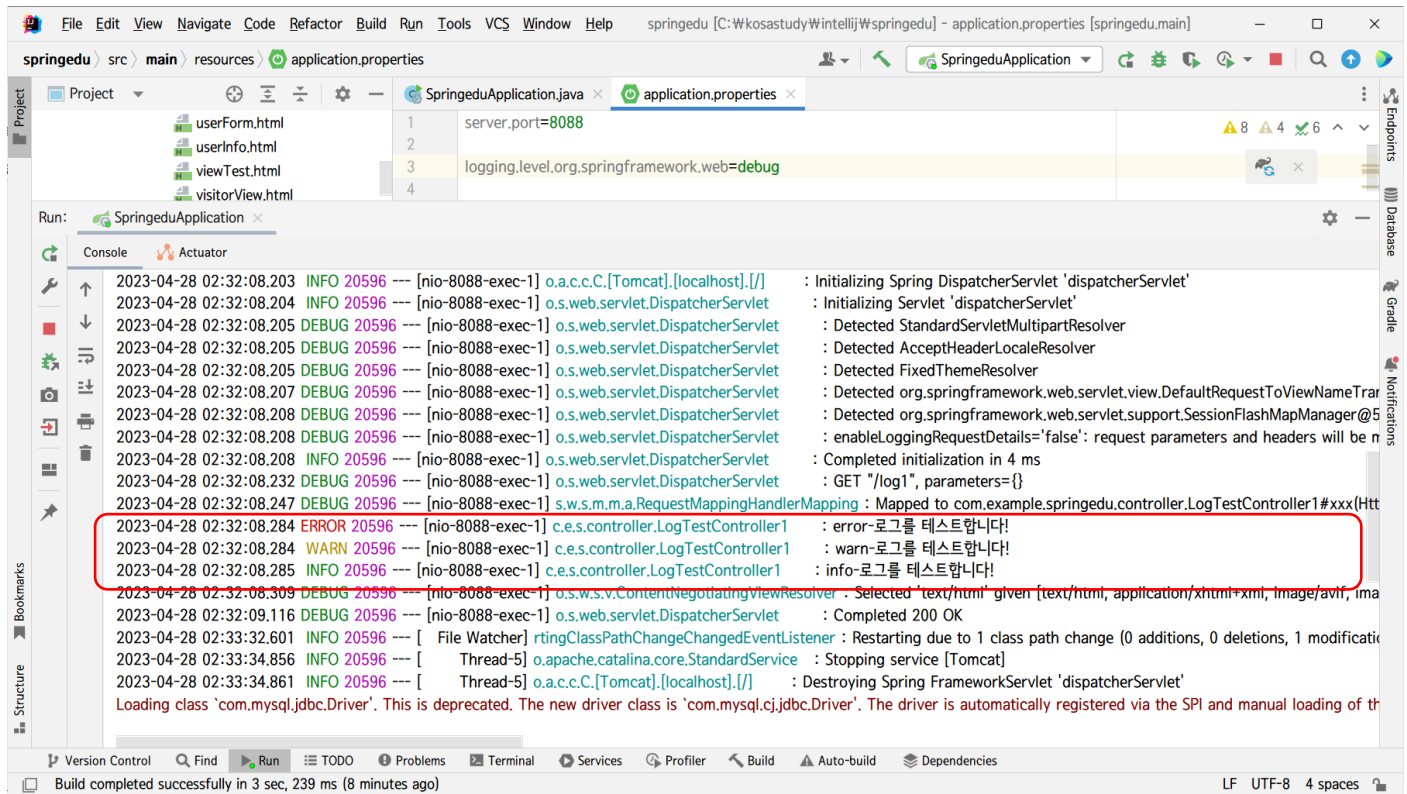
```
}
```

시스템에 설정된 로그 레벨

로그를
남길 때
사용된
메서드

	TRACE	DEBUG	INFO	WARN	ERROR	OFF
TRACE	YES	NO	NO	NO	NO	NO
DEBUG	YES	YES	NO	NO	NO	NO
INFO	YES	YES	YES	NO	NO	NO
WARN	YES	YES	YES	YES	NO	NO
ERROR	YES	YES	YES	YES	YES	NO

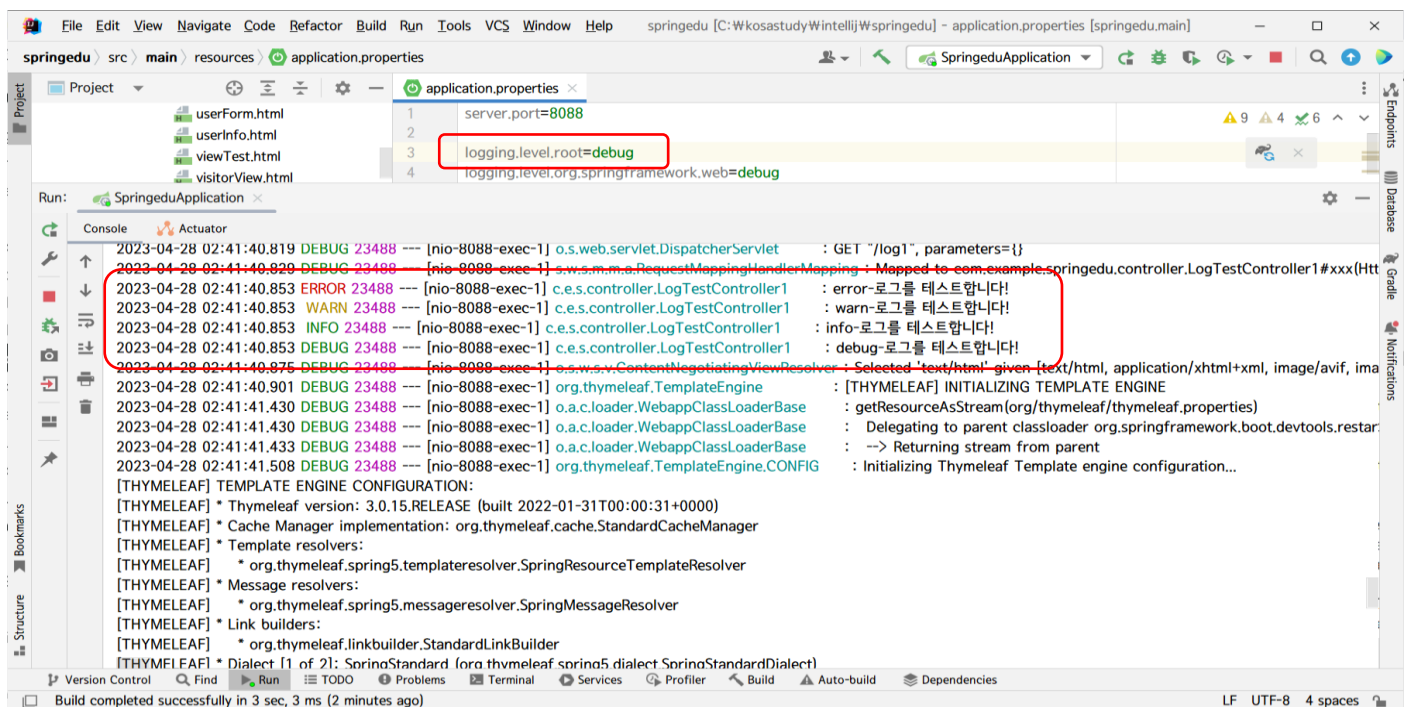
스프링의 디폴트 로그 레벨이 INFO 이므로 위에서 브라우저에서 /log1 을 요청하여 위의 컨트롤러를 실행하면 다음과 같이 info() 메서드의 로그부터 error() 메서드의 로그까지 출력되는 것을 볼 수 있다.



```
springedu [C:\Wkosastudy\Wintellij\Wspringedu] - application.properties [springedu.main]
springedu src main resources application.properties
1 server.port=8088
2
3 logging.level.org.springframework.web=debug
4

Run: SpringeduApplication
Console
2023-04-28 02:32:08.203 INFO 20596 --- [nio-8088-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2023-04-28 02:32:08.204 INFO 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2023-04-28 02:32:08.205 DEBUG 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Detected StandardServletMultipartResolver
2023-04-28 02:32:08.205 DEBUG 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Detected AcceptHeaderLocaleResolver
2023-04-28 02:32:08.205 DEBUG 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Detected FixedThemeResolver
2023-04-28 02:32:08.207 DEBUG 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Detected org.springframework.web.servlet.view.DefaultRequestToViewNameTran
2023-04-28 02:32:08.208 DEBUG 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Detected org.springframework.web.servlet.support.SessionFlashMapManager@5
2023-04-28 02:32:08.208 INFO 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : enableLoggingRequestDetails='false': request parameters and headers will be n
2023-04-28 02:32:08.232 DEBUG 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 4 ms
2023-04-28 02:32:08.247 DEBUG 20596 --- [nio-8088-exec-1] s.w.s.m.a.RequestMappingHandlerMapping : Mapped to com.example.springedu.controller.LogTestController1#xxx(Http
2023-04-28 02:32:08.284 ERROR 20596 --- [nio-8088-exec-1] c.e.s.controller.LogTestController1 : error-로그를 테스트합니다!
2023-04-28 02:32:08.284 WARN 20596 --- [nio-8088-exec-1] c.e.s.controller.LogTestController1 : warn-로그를 테스트합니다!
2023-04-28 02:32:08.285 INFO 20596 --- [nio-8088-exec-1] c.e.s.controller.LogTestController1 : info-로그를 테스트합니다!
2023-04-28 02:32:08.308 DEBUG 20596 --- [nio-8088-exec-1] o.s.w.s.v.ContentNegotiatingViewResolver : Selected 'text/html' given [text/html, application/xhtml+xml, image/avif, ima
2023-04-28 02:32:09.116 DEBUG 20596 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : Completed 200 OK
2023-04-28 02:33:32.601 INFO 20596 --- [File Watcher] rtingClassPathChangeChangedEventListener : Restarting due to 1 class path change (0 additions, 0 deletions, 1 modificati
2023-04-28 02:33:34.856 INFO 20596 --- [Thread-5] o.apache.catalina.core.StandardService : Stopping service [Tomcat]
2023-04-28 02:33:34.861 INFO 20596 --- [Thread-5] o.a.c.c.C.[Tomcat].[localhost].[/] : Destroying Spring FrameworkServlet 'dispatcherServlet'
Loading class 'com.mysql.jdbc.Driver'. This is deprecated. The new driver class is 'com.mysql.cj.jdbc.Driver'. The driver is automatically registered via the SPI and manual loading of th
```

스프링의 디폴트 로그 레벨을 DEBUG 로 변경한 후에 수행을 요청한 화면이다. debug() 메서드의 로그부터 error() 메서드의 로그까지 출력되는 것을 볼 수 있다. 그런데 스프링의 디폴트 로그 레벨을 DEBUG로 하면 스프링 부트 애플리케이션 기동시 개발자 레벨의 로그 메시지가 엄청 많이 출력되는 것을 볼 수 있다.



```
springedu [C:\Wkosastudy\Wintellij\Wspringedu] - application.properties [springedu.main]
springedu src main resources application.properties
1 server.port=8088
2
3 logging.level.root=debug
4 logging.level.org.springframework.web=debug

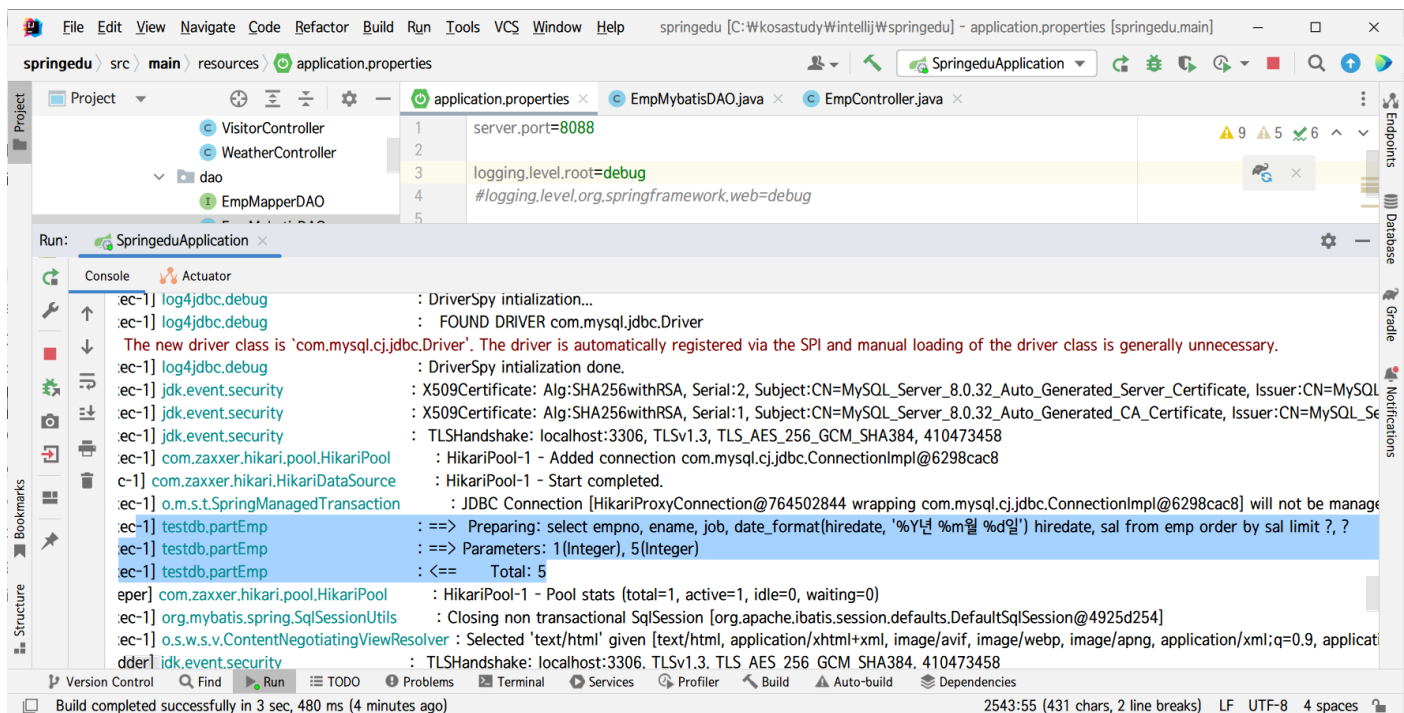
Run: SpringeduApplication
Console
2023-04-28 02:41:40.819 DEBUG 23488 --- [nio-8088-exec-1] o.s.web.servlet.DispatcherServlet : GET "/log1", parameters={}
2023-04-28 02:41:40.829 DEBUG 23488 --- [nio-8088-exec-1] s.w.s.m.a.RequestMappingHandlerMapping : Mapped to com.example.springedu.controller.LogTestController1#xxx(Http
2023-04-28 02:41:40.853 ERROR 23488 --- [nio-8088-exec-1] c.e.s.controller.LogTestController1 : error-로그를 테스트합니다!
2023-04-28 02:41:40.853 WARN 23488 --- [nio-8088-exec-1] c.e.s.controller.LogTestController1 : warn-로그를 테스트합니다!
2023-04-28 02:41:40.853 INFO 23488 --- [nio-8088-exec-1] c.e.s.controller.LogTestController1 : info-로그를 테스트합니다!
2023-04-28 02:41:40.853 DEBUG 23488 --- [nio-8088-exec-1] c.e.s.controller.LogTestController1 : debug-로그를 테스트합니다!
2023-04-28 02:41:40.875 DEBUG 23488 --- [nio-8088-exec-1] o.s.w.s.v.ContentNegotiatingViewResolver : Selected 'text/html' given [text/html, application/xhtml+xml, image/avif, ima
2023-04-28 02:41:40.901 DEBUG 23488 --- [nio-8088-exec-1] org.thymeleaf.TemplateEngine : [THYMELEAF] INITIALIZING TEMPLATE ENGINE
2023-04-28 02:41:41.430 DEBUG 23488 --- [nio-8088-exec-1] o.a.c.loader.WebappClassLoaderBase : getResourceAsStream(org/thymeleaf/thymeleaf.properties)
2023-04-28 02:41:41.430 DEBUG 23488 --- [nio-8088-exec-1] o.a.c.loader.WebappClassLoaderBase : Delegating to parent classloader org.springframework.boot.devtools.restart
2023-04-28 02:41:41.433 DEBUG 23488 --- [nio-8088-exec-1] o.a.c.loader.WebappClassLoaderBase : --> Returning stream from parent
2023-04-28 02:41:41.508 DEBUG 23488 --- [nio-8088-exec-1] org.thymeleaf.TemplateEngine.CONFIG : Initializing Thymeleaf Template engine configuration...
[THYMELEAF] * Thymeleaf version: 3.0.15.RELEASE (built 2022-01-31T00:00:31+0000)
[THYMELEAF] * Cache Manager implementation: org.thymeleaf.cache.StandardCacheManager
[THYMELEAF] * Template resolvers:
[THYMELEAF]   * org.thymeleaf.spring5.templateresolver.SpringResourceTemplateResolver
[THYMELEAF] * Message resolvers:
[THYMELEAF]   * org.thymeleaf.spring5.messageresolver.SpringMessageResolver
[THYMELEAF] * Link builders:
[THYMELEAF]   * org.thymeleaf.linkbuilder.StandardLinkBuilder
[THYMELEAF] * Dialect [1 of 2]: SprinStandard (ora thymeleaf sprin5 dialect SprinStandardDialect)
```

Logback 은 5개의 로그레벨을 지원하며 **ERROR, WARN, INFO, DEBUG, TRACE** 이다. 설정 파일에 대상별(자바에서는 패키지 또는 클래스)로 레벨 지정이 가능하고 그 등급 이상의 로그만 저장하는 방식이다.

LEVEL : TRACE > DEBUG > INFO > WARN > ERROR
저 고

개발 서버는 DEBUG 레벨을 설정하여 로그를 남기고 운영 서버는 INFO 레벨로 로그를 남기도록 설정하는 것이 일반적이다.

- Mybatis 에서 수행되는 SQL 문의 로그 남기기



SQL 로그를 조금 더 보기 좋게 출력하려면 log4jdbc 라는 SQL 로깅을 위한 JDBC 프락시 드라이버를 사용한다. 약간의 환경 설정이 필요하다.

(1) **build.gradle** 파일의 dependencies 블록에 다음 행을 추가한다.

```
implementation 'org.bgee.log4jdbc-log4j2:log4jdbc-log4j2-jdbc4.1:1.16'
```

(2) `application.properties` 에 작성되어 있는 JDBC 드라이버 정보와 URL 정보를 다음 내용으로 변경한다.

```
spring.datasource.url: jdbc:log4jdbc:mysql://localhost:3306/edudb?characterEncoding=UTF-8&serverTimezone=UTC
spring.datasource.driver-class-name=net.sf.log4jdbc.sql.jdbcapi.DriverSpy
```


(3) 다음 내용으로 작성된 `log4jdbc.log4j2.properties` 파일을 생성한다.

```
log4jdbc.spylogdelegator.name=net.sf.log4jdbc.log.slf4j.Slf4jSpyLogDelegator
log4jdbc.dump.sql.maxlinelength=0
```

(4) 로그 설정 상세내용

`jdbc.sqlonly` - SQL 문을 보여준다.

`jdbc.sqltiming` - SQL 문과 이 SQL 문을 수행하는 시간(ms)을 같이 보여준다.

`jdbc.audit` - ResultSet 을 제외한 모든 JDBC 호출 정보를 로그로 보여준다.

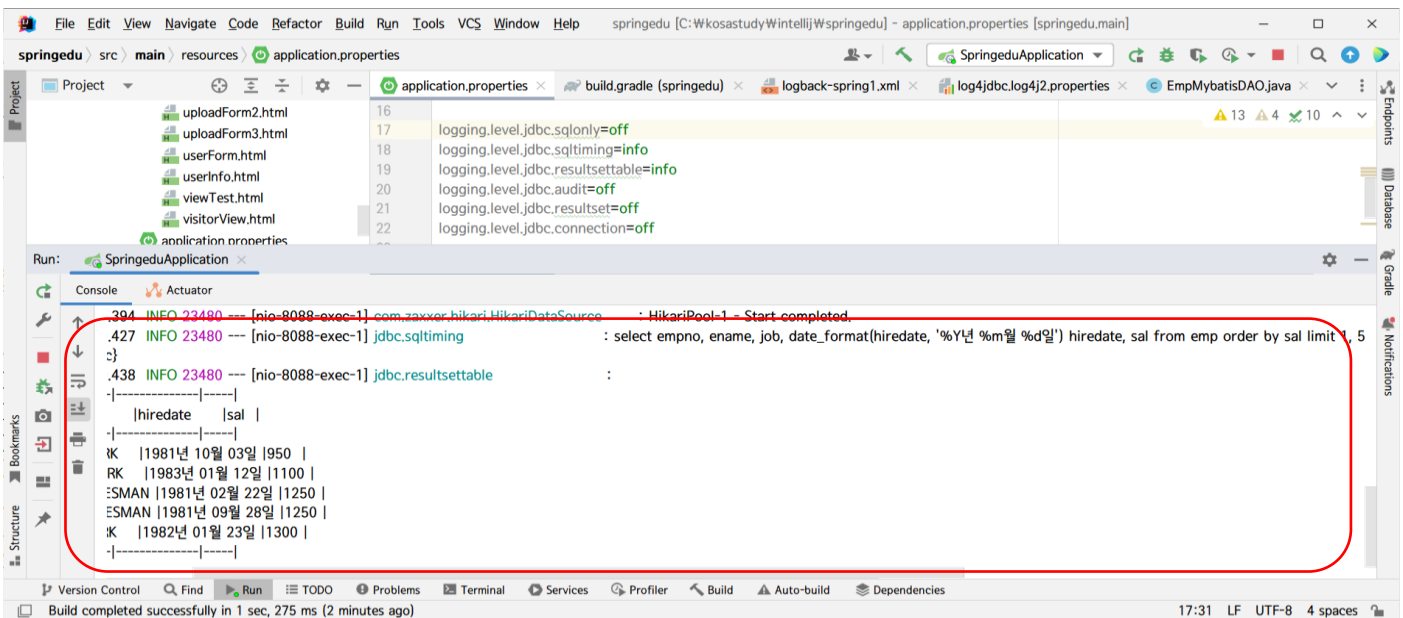
상당히 많은 양의 로그가 나오기 때문에 권장하지 않음.

`jdbc.resultset` - ResultSet 을 포함한 모든 JDBC 호출 정보를 로그로 보여준다.

audit 처럼 많은 로그가 나오기 때문에 권장하지 않음.

`jdbc.resultsettable` - SQL 의 결과로 조회된 데이터를 table 형태로 로그를 보여준다.

`jdbc.connection` - DB 연결, 연결 해제와 관련된 로그를 보여준다



스프링의 디폴트 로그 레벨은 INFO 로 두고 위에서 제시한 Mybatis 로그 설정만 해도 개발에 많은 도움이 될 것으로 보인다.

(2) 매퍼파일 작성시 #{xxx} 와 \${xxx} 의 사용

#{xxx} → 'xxx'

\${xxx} → xxx

#{xxx}을 사용하면 자동으로 타입에 맞춰서 단일 인용부호가 붙어서 처리되므로 SQL 문에 사용되는 동적 **데이터 값** 설정에 적당하다. 그런데 만일 테이블명 또는 컬럼명을 동적으로 설정하려는 경우엔 자동으로 부여되는 단일 인용부호가 필요없으므로 이 때는 \${xxx}를 사용한다.

예1)

```
select name, schoolname, addr from ${tablename} where ${colname} = #{colvalue}
```

예2)

```
select id, name, writedate, memo from visitor where memo like '%||#{key}||%'
```

```
select id, name, writedate, memo from visitor where memo like '%${key}%'
```

☆ 주의할 사항은 \${xxx}의 경우에는 xxx 가 반드시 전달되는 객체의 프로퍼티명이거나 HashMap 객체의 키값이어야 한다.

(3) SELECT 명령의 수행 결과를 HashMap 객체로 받아오기

```
Map<키타입, XxxVO> map = session.selectMap("매퍼의태그명", "키로사용할프로퍼티명");
```

(4) INSERT 할 데이터를 HashMap 객체로 주기

```
<insert id="태그명" parameterType="hashmap">
```

```
HashMap<String, String> map = new HashMap<>();
```

```
map.put("name", ".....");
```

```
map.put("writedate", ".....");
```

```
map.put("memo", ".....");
```

```
session.insert(statement, map);
```

(5) 매퍼 파일 대신 애노테이션으로 처리하기

Mybatis 3.0 부터 XML 로 작성하는 매퍼파일 대신 자바 클래스 또는 인터페이스에 작성되는 애노테이션으로 대신할 수 있다. 매퍼를 애노테이션으로 처리하려는 경우에는 스프링 부트 애플리케이션 기동 프로그램(SpringedApplication.java) @MapperScan 에 대한 정보를 작성해야 한다.

```
@MapperScan(value={"com.example.springedu.dao"})
```

구현 방법은

애노테이션과 SQL 이 정의된 인터페이스(매퍼 인터페이스)

애노테이션이 정의된 인터페이스+SQL이 정의된 매퍼파일

애노테이션이 정의된 인터페이스+SQL과 애노테이션이 정의된 매퍼 클래스

등 다양하다. 우리는 매퍼 인터페이스 방법으로 테스트 해본다. 다른 구현 방법들은

<https://mybatis.org/mybatis-3/java-api.html> 을 참고한다.

```
import com.example.springedu.domain.EmpVO;
```

```
import com.example.springedu.domain.PageDTO;
```

```
import org.apache.ibatis.annotations.Mapper;
```

```
import org.apache.ibatis.annotations.Select;
```

```
import java.util.List;
```

```
@Mapper
```

```
public interface EmpMapperDAO {
```

```
    @Select("select count(*) from emp")
```

```
    public int getAllDataNum();
```

```
    @Select("select empno, ename, job, date_format(hiredate, '%Y년 %m월 %d일') hiredate, sal from emp")
```

```
    public List<EmpVO> listAll();
```

```
    @Select("select empno, ename, job, date_format(hiredate, '%Y년 %m월 %d일') hiredate, sal from emp order by sal limit #{startNum}, #{countNum}")
```

```
    public List<EmpVO> listPart(PageDTO vo);
```

```
}
```