1. pom.xml

```null 1.8 // @PostMapping @GetMapping 을 위해 5.0.7 로 버전 변경 5.0.7.RELEASE 1.6.10 1.6.6 .... 중략 // MYSQL 커넥터 , 히카리 커넥터 풀(DB소스) , jdbc , mybatis, log4jdbc 연결 mysql mysql-connector-java 8.0.22

```

```
<!-- HikariCP -->
<dependency>
        <groupId>com.zaxxer</groupId>
        <artifactId>HikariCP</artifactId>
        <version>3.4.5</version>
</dependency>

<!-- spring-jdbc -->
<dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>${org.springframework-version}</version>
</dependency>

<!-- mybatis -->
<dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis</artifactId>
        <version>3.5.6</version>
</dependency>

<!-- mybatis-spring -->
<dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis-spring</artifactId>
        <version>2.0.6</version>
</dependency>

<!-- log4jdbc-log4j2-jdbc4 / 콘솔에 현재 실행되는 sql 문장 출력 -->
<dependency>
        <groupId>org.bgee.log4jdbc-log4j2</groupId>
        <artifactId>log4jdbc-log4j2-jdbc4</artifactId>
```

```
                <version>1.16</version>
        </dependency>

        <dependency>
                <groupId>org.bgee.log4jdbc-log4j2</groupId>
                <artifactId>log4jdbc-log4j2-jdbc4.1</artifactId>
                <version>1.16</version>
        </dependency>
```


2. web.xml
```null
 // 한글을 위해
        <filter>
                <filter-name>encodingFilter</filter-name>
                <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
                <init-param>
                        <param-name>encoding</param-name>
                        <param-value>utf-8</param-value>
                </init-param>
        </filter>

        <filter-mapping>
                <filter-name>encodingFilter</filter-name>
                <url-pattern>/*</url-pattern>
        </filter-mapping>
```

3. root-context ```null

```
<!-- SqlSessionFactoryBean 등록 -->
<bean id="sqlSessionFactoryBean"
        class="org.mybatis.spring.SqlSessionFactoryBean"
        p:dataSource-ref="datasource"
        p:mapperLocations="classpath:com/bit/kim/mapper/*.xml">
</bean>

<!-- SqlSessionTemplate 등록 -->
```

```
<bean id="sqlSessionTemplate"
        class="org.mybatis.spring.SqlSessionTemplate">
        <constructor-arg ref="sqlSessionFactoryBean" />
</bean>
```

```
<br>
### 4. servlet-context
```null
 //  기본적으로 스캔 하는 것 제외하고 패키지를 생성한 경우에 추가
        <context:component-scan
                base-package="com.bit.kim" />
        <context:component-scan base-package="service" />
        <context:component-scan base-package="dao" />
        <context:component-scan base-package="dto" />
```

5. log4j ```null // 콘솔에서 db 에서 넘어오는 데이터를 읽기 위해 // warm -> info ```
6. log4jdbc.log4j2.properties ```null // 없으면 파일 생성 log4j 와 같은 위치
log4jdbc.spylogdelegator.name=net.sf.log4jdbc.log.slf4j.Slf4jSpyLogDelegator ```
7. Controller , dao , service , dto , mapper 1. 컨트롤러 ```null @Controller public class HomeController {

```
    @Autowired
private listservice service;

@RequestMapping(value = "/")
public String home(Model model) {

        List<listdto> list = service.selectlist();

        model.addAttribute("list2", list);
        System.out.println(list);

        return "home";
}
}
```

```
<br>
2. 서비스,서비스임플
```
```

```null
// service.java
public interface listservice {

	public List<listdto> selectlist();

	public void insertname(String name);

}

// serviceimpl.java
@Service
public class listserviceimpl implements listservice {

	@Inject
	private listdao dao;

	public List<listdto> selectlist() {

		return dao.selectlist();
	}
}
```

3. 다오,다오임플

```
// dao.java
public interface listdao {

	public List<listdto> selectlist();

}

// daoimpl.java
@Repository
public class listdaoimpl implements listdao {

	@Autowired
	private SqlSessionTemplate templated;

	private static String namespace = "dao.listdao";
```

```
        public List<listdto> selectlist() {
                return templated.selectList(namespace + ".selectlist");
        };

}
```

4. 맵퍼 ```null ```

---

8. 주의사항

1. 라이브러리 및 기타 설정을 잘 추가하자
2. 경로에 항상 주의하자.
3. MVC 구조를 충분히 이해 하고 사용하자.
4. 인터페이스를 안써도 되지만 사용하자.
5. 각 라이브러리와 설정을 를 왜 사용하는지 이해하자