SPRING 03

김규석 교수
(스프링프레임워크)

한국폴리텍대학

학습목표

- ✓ Spring에서 DAO, DTO의 이해
- ✓ Spring에서 Service의 이해
- ✓ Spring 프로젝트에서 MySQL 연동하기

DAO

✓ DAO(Data Access Object)

– DB에 접근을 하기 위한 객체

```
Statement statement = null;
ResultSet resultSet = null;
Connection connection = null;
String id = "root";
String password = "비밀번호";
String dbQuery = "show databases";

try {
    connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&useSSL=false", id, password);
    statement = connection.createStatement();

    if (statement.execute(dbQuery)) {
        resultSet = statement.getResultSet();
    }

    while (resultSet.next()) {
        String str = resultSet.getString(1);
        System.out.println(str);
    }

} catch(Exception e) {
    e.printStackTrace();
}
```

DTO

✓ DTO(Data Transfer Object)

– Controller, View 등의 계층간 데이터를 교환하기 위한 객체

```
    private String _name;
    private String _email;
    private String _phone;

    public Member(){

    }

    public String get_name() {
        return _name;
    }

    public void set_name(String _name) {
        this._name = _name;
    }

    public String get_email() {
        return _email;
    }

    public void set_email(String _email) {
        this._email = _email;
    }

    public String get_phone() {
        return _phone;
    }

    public void set_phone(String _phone) {
        this._phone = _phone;
    }
```

Service

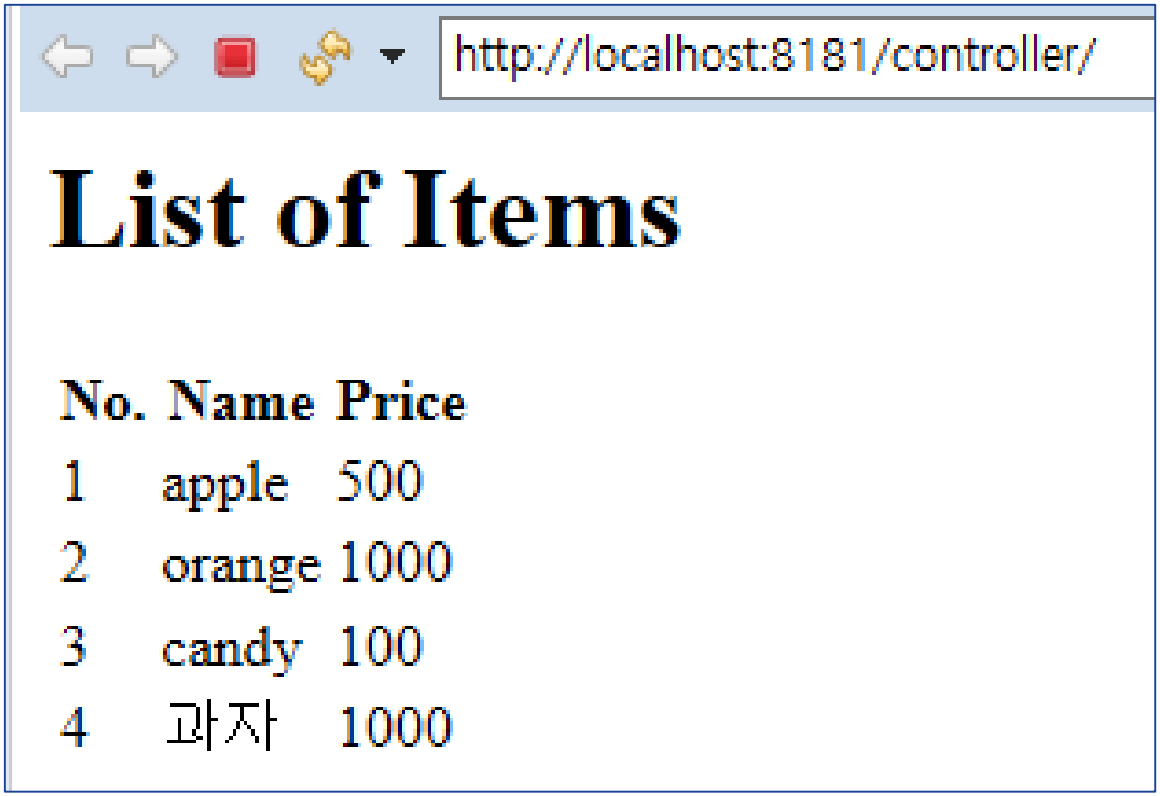
✓ Service

– DAO를 통해서 DB에 접근하고 데이터를 가공함

```
@Inject
private DAO dao;

@Override
public List<VO> selectItems() throws Exception {
    return dao.selectItem();
}

@Override
public int insertItems(Member mb) throws Exception {
    return dao.insertItem(mb);
}

@Override
public int insertItems() throws Exception {
    return dao.insertItem();
}

@Override
public String getallTheItems() throws Exception {
    List<VO> mV = dao.selectItem();
    int lengthOfList = mV.size();
    String result = "";
    for (int i = 0; i < lengthOfList; i++) {
        result = result + mV.get(i) + " ";
    }
    return result;
}
```

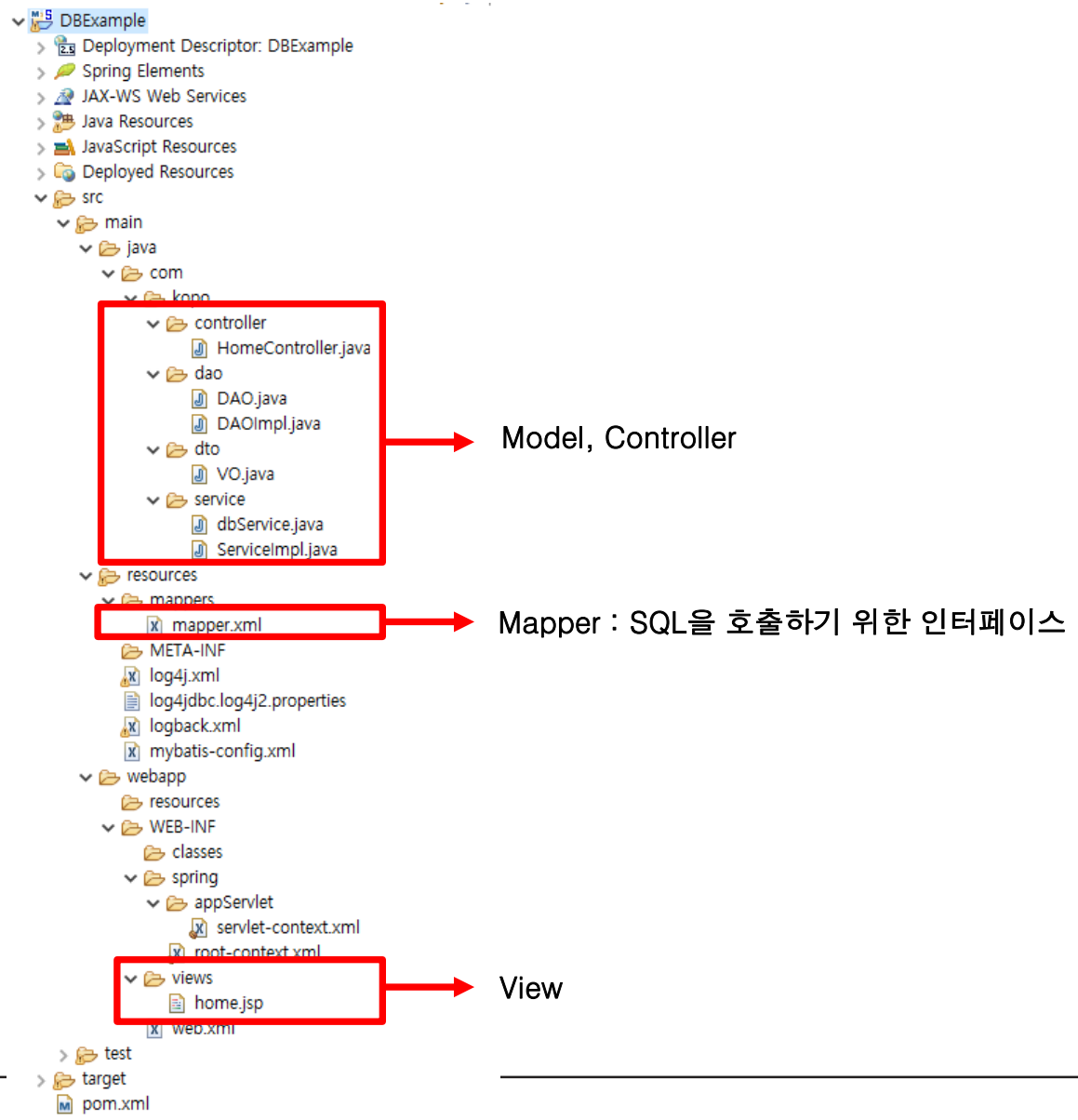
Spring 프로젝트에서 MySQL 연동하기

✓ 첨부된 DBExample 프로젝트를 Import하여 실행하기



Spring 프로젝트에서 MySQL 연동하기

✓ 프로젝트 파일 구조



Spring 프로젝트에서 MySQL 연동하기

- ✓ root-context.xml
- ✓ bean 생성을 위해 아래 내용 수정 / 추가

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:jdbc="http://www.springframework.org/schema/jdbc"
    xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"
    xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.3.xsd
        http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring-1.2.xsd
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd">

    <!-- Root Context: defines shared resources visible to all other web components -->

    <!-- MySQL dataSource -->
    <bean id="dataSource"
        class="org.springframework.jdbc.datasource.DriverManagerDataSource">
        <property name="driverClassName" value="net.sf.log4jdbc.sql.jdbcapi.DriverSpy"></property>
        <property name="url"
            value="jdbc:log4jdbc:mysql://localhost:3306/joinasamember?useSSL=false&amp;serverTimezone=UTC"></property>
        <property name="username" value="root"></property>
        <property name="password" value="root"></property>
    </bean>

    <!-- mybatis SqlSessionFactoryBean -->
    <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"></property>
        <property name="configLocation" value="classpath:/mybatis-config.xml"></property>
        <property name="mapperLocations" value="classpath:mappers/**/*mapper.xml"></property>
    </bean>

        <!-- mybatis     -->
    <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate"
        destroy-method="clearCache">
        <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"></constructor-arg>
    </bean>

    <!-- bean 등록 -->
    <context:component-scan base-package="com.kopo.dao"></context:component-scan>
    <context:component-scan base-package="com.kopo.service"></context:component-scan>
```

Spring 프로젝트에서 MySQL 연동하기

- ✓ servlet-context.xml
- ✓ 아래와 같이 설정 수정하기

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc.xsd
        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context.xsd">

    <!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -->

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources in the ${webappRoot}/resources directory -->
    <resources mapping="/resources/**" location="/resources/" />

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF/views directory -->
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <context:component-scan base-package="com.kopo.controller" />

</beans:beans>
```

Spring 프로젝트에서 MySQL 연동하기

✓ 실습 #1

– 인터페이스 id를 ‘select’로 변경하여 동작하게 하기

```
<mapper namespace="com.kopo.mapper.mapper">
    <select id="list" resultType="VO">
        SELECT ID, NAME, PRICE FROM ITEMS
    </select>
```

Spring 프로젝트에서 MySQL 연동하기

✓ 실습 #2

– 속성값을 'select1'으로 변경하여 동작하게 하기

```
@RequestMapping(value = "/", method = RequestMethod.GET)
public String home(Locale locale, Model model) throws Exception{

    logger.info("home");

    List<VO> myMembersList = service.selectItems();

    model.addAttribute("select1", myMembersList);
```

Spring 프로젝트에서 MySQL 연동하기

✓ 실습 #3

– Annotation을 수정하여 URL 변경해보기

```
@RequestMapping(value = "/", method = RequestMethod.GET)
public String home(Locale locale, Model model) throws Exception{

    logger.info("home");

    List<VO> myMembersList = service.selectItems();

    model.addAttribute("select1", myMembersList);
```

Spring 프로젝트에서 MySQL 연동하기

✓ 실습 #4

- Annotation을 수정하여 Parameter를 받아보기

```
@RequestMapping(value = "/", method = RequestMethod.GET)
public String home(Locale locale, Model model) throws Exception{

    logger.info("home");

    List<VO> myMembersList = service.selectItems();

    model.addAttribute("select1", myMembersList);
```

과제 #1

- “DB Example의 전체 구조를 자세히 이해하기”

지금부터....

- ✓ 프로젝트 기획서 v0.8 작성