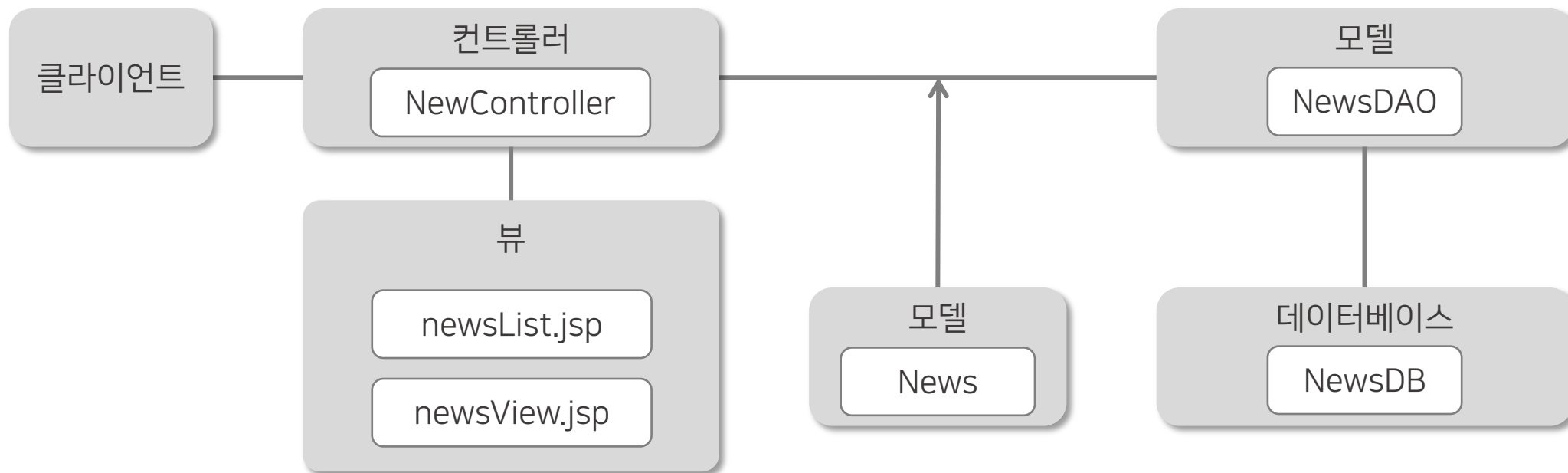






[프로젝트 실습] 뉴스 기사 관리 서비스

- 뉴스 기사를 제공하는 웹 서비스 제작
 - 부트스트랩을 활용한 화면 구현
 - DB 연동 (뉴스 DataBase에 새로운 기사를 등록, 조회, 삭제가 가능)
 - 이미지 파일 첨부 포함
- 프로젝트 구조





[프로젝트 실습] 뉴스 기사 관리 서비스 - DB 구축

-- news_db 생성

```
DROP DATABASE IF EXISTS `news_db`;
```

```
CREATE DATABASE news_db DEFAULT CHARACTER SET = 'utf8mb4' COLLATE = utf8mb4_0900_ai_ci;
```

-- 생성된 DB 선택

```
USE news_db;
```

-- news 테이블 생성

```
CREATE TABLE news (  
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    title VARCHAR(255) NOT NULL,  
    img VARCHAR(255) NOT NULL,  
    date DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,  
    content VARCHAR(255) NOT NULL  
);
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - 프로젝트 생성

- Dynamic Web Project 생성 [프로젝트 명: news]
- 프로젝트에서 마우스 우클릭, [Configure] [Convert to Maven Project]

Create new POM

Maven POM

This wizard creates a new POM (pom.xml) descriptor for Maven.

Project: /news

Artifact

Group Id: news

Artifact Id: news

Version: 0.0.1-SNAPSHOT

Packaging: jar

Name:

Description:

Finish Cancel



[프로젝트 실습] 뉴스 기사 관리 서비스 - 라이브러리 추가

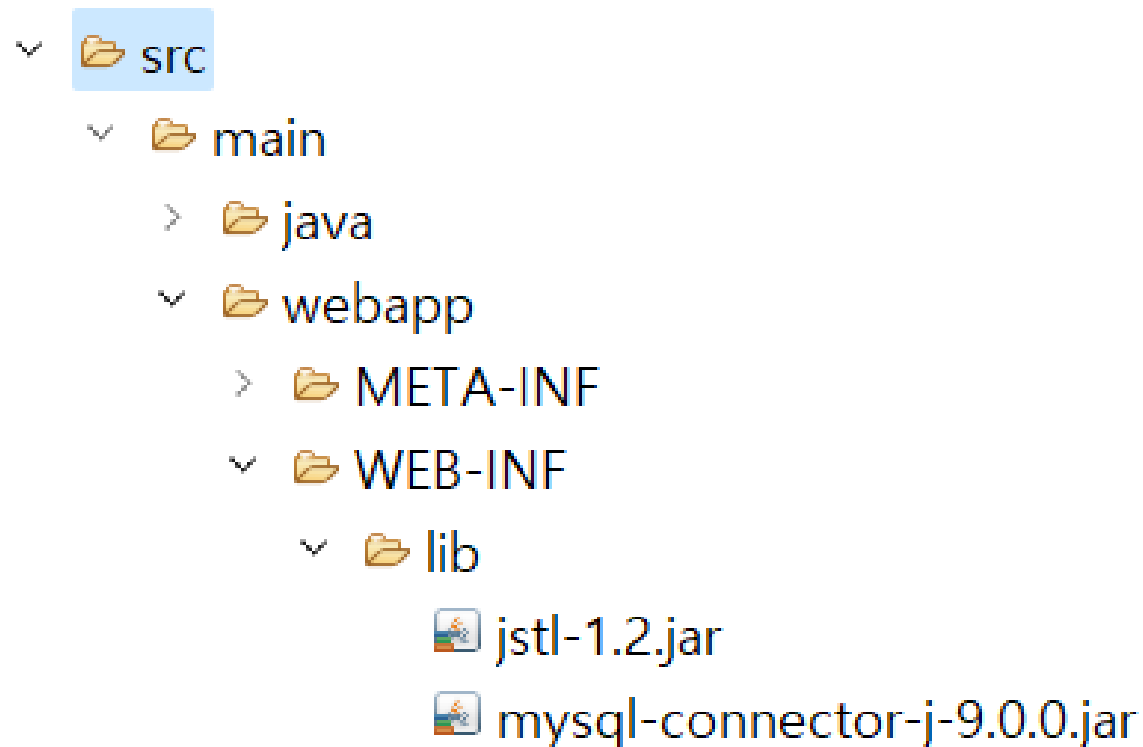
- MySQL connector J, JSTL, LOMBOK 라이브러리를 추가

```
<dependencies>
  <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
  <dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
    <version>1.18.34</version>
    <scope>provided</scope>
  </dependency>
</dependencies>
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - 라이브러리 추가

- MySQL connector J, JSTL, LOMBOK 라이브러리를 추가





[프로젝트 실습] 뉴스 기사 관리 서비스 - 패키지 구성

- ▼ Java Resources
 - ▼ src/main/java
 - com.news.controller
 - com.news.dao
 - com.news.model
 - com.news.service
 - com.news.util
- ▼ Libraries
 - > JRE System Library [JavaSE-17]
 - ▼ Maven Dependencies
 - > mysql-connector-j-9.0.0.jar - C:\WUser
 - > protobuf-java-4.26.1.jar - C:\WUsers\W/
 - > lombok-1.18.34.jar - C:\WUsers\WAdmi
 - > Server Runtime [apache-tomcat-9.0.89]
 - ▼ Web App Libraries
 - > jstl-1.2.jar - C:\WUsers\WAdministrator\W



[프로젝트 실습] 뉴스 기사 관리 서비스 - News 클래스 구현

- News 클래스는 뉴스 데이터 구조를 표현하기 위한 모델(DO 클래스)로, 테이블 구조와 일치하도록 한다.
- 여기서 date 속성은 Date 타입보다 화면 출력에 유리하도록 String 타입으로 처리하도록 한다.

```
@Data
@AllArgsConstructor
public class News {
    private int id;
    private String title, img, date, content;
}
```




[프로젝트 실습] 뉴스 기사 관리 서비스 - NewsDAO 인터페이스 생성

- NewsDAO는 News_DB에 있는 news 테이블과 연동해서, 뉴스 등록, 목록 가져오기, 삭제 기능 등을 제공한다.
- 만약 위의 기능 외에 다른 기능이 필요한 경우에는 DAO 클래스에 메소드를 추가해 구현할 수 있다.
- 인터페이스로 구현하고, 구현 객체를 따로 만들어 메소드를 완성하자.
- 또한 먼저 메소드마다 매번 연결을 만들지 않고, 커넥션 풀을 통해 미리 생성된 연결을 받아 사용할 수 있도록 구성하자.

```
public interface NewsDAO {  
    void addNews(News news) throws Exception;  
    List<News> getAll() throws Exception;  
    News getNews(int id) throws Exception;  
    void deleteNews(int id) throws Exception;  
}
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - DB 커넥션 풀 구현

```
public class ConnectionPool {
    public static class DBPool {
        private static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
        private static final String DB_URL = "jdbc:mysql://localhost:3306/news_db";
        private static final String DB_USER = "root";
        private static final String DB_PASSWORD = "1234";
        static final BasicDataSource dbcp = new BasicDataSource();
        static {
            try {
                dbcp.setDriverClassName(JDBC_DRIVER);
                dbcp.setUrl(DB_URL);
                dbcp.setUsername(DB_USER);
                dbcp.setPassword(DB_PASSWORD);
            } catch (Exception e){
                e.printStackTrace();
            }
        }
        public static Connection getDBPool() throws SQLException {
            return dbcp.getConnection();
        }
    }
}
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - NewsDAO 구현 클래스 작성

- NewsDAOImpl 클래스를 만들어, NewsDAO 인터페이스를 구현해보자.

```
public class NewsDAOImpl implements NewsDAO {  
    @Override  
    public void addNews(News news) throws Exception {  
        System.out.println("addNews");  
    }  
    @Override  
    public List<News> getAll() throws Exception {  
        System.out.println("getAll");  
        return null;  
    }  
    @Override  
    public News getNews(int id) throws Exception {  
        System.out.println("getNews");  
        return null;  
    }  
    @Override  
    public void deleteNews(int id) throws Exception {  
        System.out.println("deleteNews");  
    }  
}
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - NewsController 인터페이스 생성

- 사용자의 요청을 받아 모델과 연동하고, 처리된 결과를 뷰로 전달하는 컨트롤러를 구현해보자.

```
public interface NewsController {  
    void process(HttpServletRequest request, HttpServletResponse response) throws  
        ServletException, IOException;  
}
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - NewsController 인터페이스 구현

```
public class NewsListController implements NewsController {  
    @Override  
    public void process(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.getWriter().append("리스트 보기");  
    }  
}
```

```
public class NewsViewController implements NewsController {  
    @Override  
    public void process(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
        response.getWriter().append("기사 보기");  
    }  
}
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - NewsControllerServlet 구현

```
@WebServlet("/v1/*")
@MultipartConfig(maxFileSize=1024*1024*2, location="C:\\Users\\WD")
public class NewsControllerServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private Map<String, NewsController> controllerMap = new HashMap<>();
    public NewsControllerServlet() {
        controllerMap.put("/news/v1/newsList.do", new NewsListController());
        controllerMap.put("/news/v1/news.do", new NewsViewController());
    }

    @Override
    protected void service(HttpServletRequest request, HttpServletResponse response) throws
        ServletException, IOException {
        String requestURI = request.getRequestURI();
        System.out.println(requestURI);
        NewsController controller = controllerMap.get(requestURI);
        if (controller == null) {
            controller = controllerMap.get("/news/v1/newsList.do");
        }
        controller.process(request, response);
    }
}
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - 뷰 (JSP 파일) 작성

- 뉴스 목록 화면(NewsList.jsp)에서는 뉴스 목록을 보여주고, 뉴스를 선택해 세부 기사를 보거나 삭제할 수 있으며, 하단에 뉴스 등록 버튼을 두어, 기사를 등록할 수 있도록 구성한다.
- 뉴스 화면(NewsView.jsp)은 세부 기사 내용을 볼 수 있도록 구성한다.
- JSP 파일은 WEB-INF 내부에 views 폴더를 생성하고, views 폴더 하위에 NewsList.jsp와 NewsView.jsp를 생성한다.

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>뉴스 목록</title>
</head>
<body>
    <h1>뉴스 목록 표시</h1>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html;
charset=UTF-8"
pageEncoding="UTF-8"%>
<%@ taglib prefix="c"
uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>뉴스 상세 보기</title>
</head>
<body>
    <h1>뉴스 세부 내용 표시</h1>
</body>
</html>
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - 컨트롤러 수정

@Override

```
public void process(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    String view = "/WEB-INF/views/NewsList.jsp";  
    RequestDispatcher dispatcher = request.getRequestDispatcher(view);  
    dispatcher.forward(request, response);  
}
```

@Override

```
public void process(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    String view = "/WEB-INF/views/NewsView.jsp";  
    RequestDispatcher dispatcher = request.getRequestDispatcher(view);  
    dispatcher.forward(request, response);  
}
```




[프로젝트 실습] 뉴스 기사 관리 서비스 - 컨트롤러 수정

@Override

```
public void process(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    String view = "/WEB-INF/views/NewsList.jsp";  
    RequestDispatcher dispatcher = request.getRequestDispatcher(view);  
    dispatcher.forward(request, response);  
}
```

@Override

```
public void process(HttpServletRequest request, HttpServletResponse response) throws  
ServletException, IOException {  
    String view = "/WEB-INF/views/NewsView.jsp";  
    RequestDispatcher dispatcher = request.getRequestDispatcher(view);  
    dispatcher.forward(request, response);  
}
```



[프로젝트 실습] 뉴스 기사 관리 서비스 - 서비스 작성

```
public interface NewsService {  
    void addNews(HttpServletRequest request) throws Exception;  
    List<News> getAll() throws Exception;  
    News getNews(HttpServletRequest request) throws Exception;  
    void deleteNews(HttpServletRequest request) throws Exception;  
}
```