박 경 태
comsi.java@gmail.com

고급 자바 프로그래밍

: STS를 이용한 Spring 프로그래밍

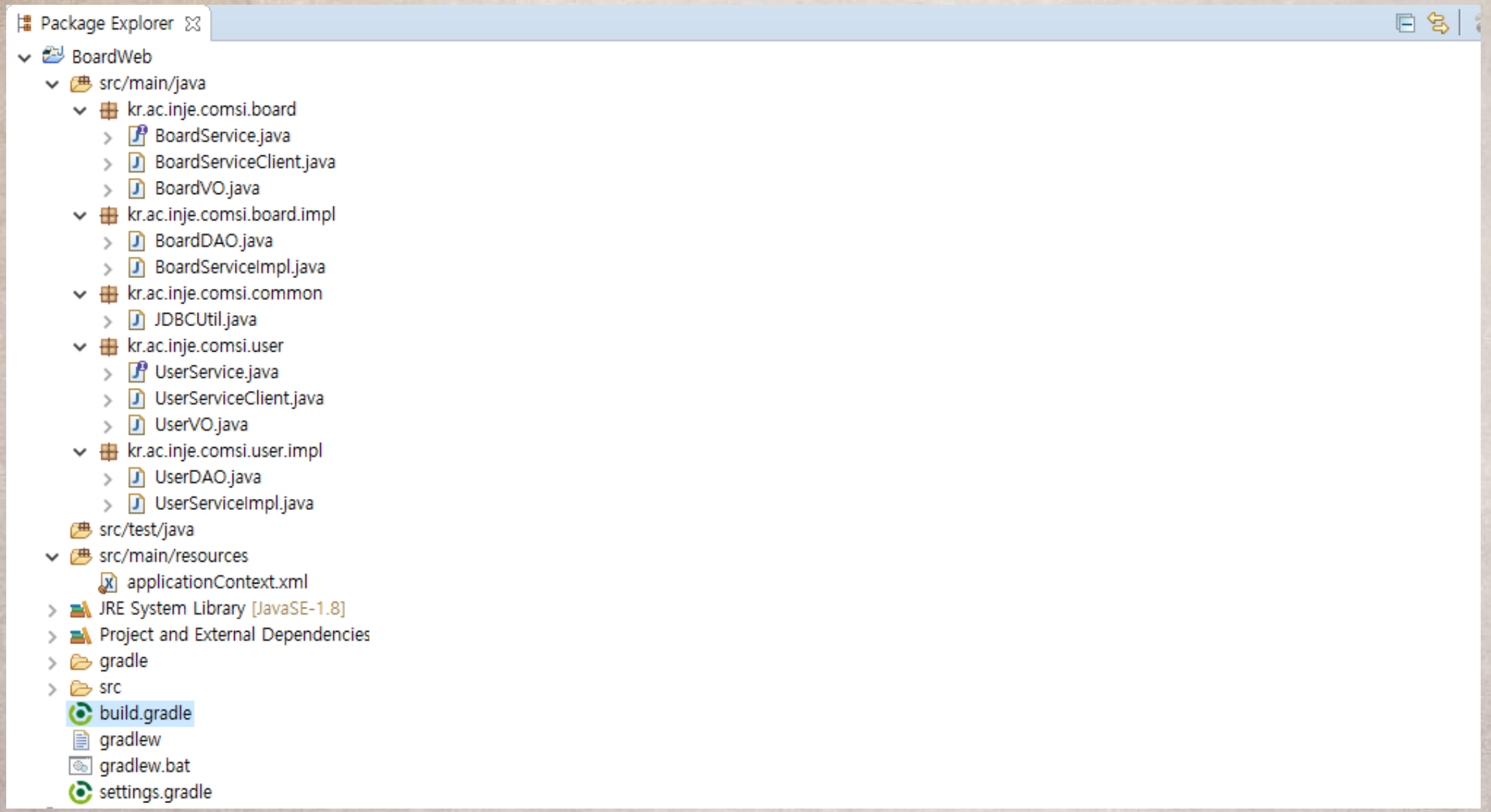
강의 내용

순서	내 용
1	• Spring IoC를 이용한 비즈니스 컴포넌트 만들기
2	• Spring AOP(Aspect Oriented Programming)를 이용한 공통 서비스 만들기 • Spring DAO(Data Access Object)를 이용한 데이터베이스 연동 및 트랜잭션 처리
3	• Spring MVC를 이용한 MVC 아키텍쳐 적용하기
4	• Spring MVC의 부가 기능 사용하기(파일 업로드, 다국어, 예외 처리 등)
5	• Spring과 MyBatis 연동하기 • Spring과 JPA 연동하기

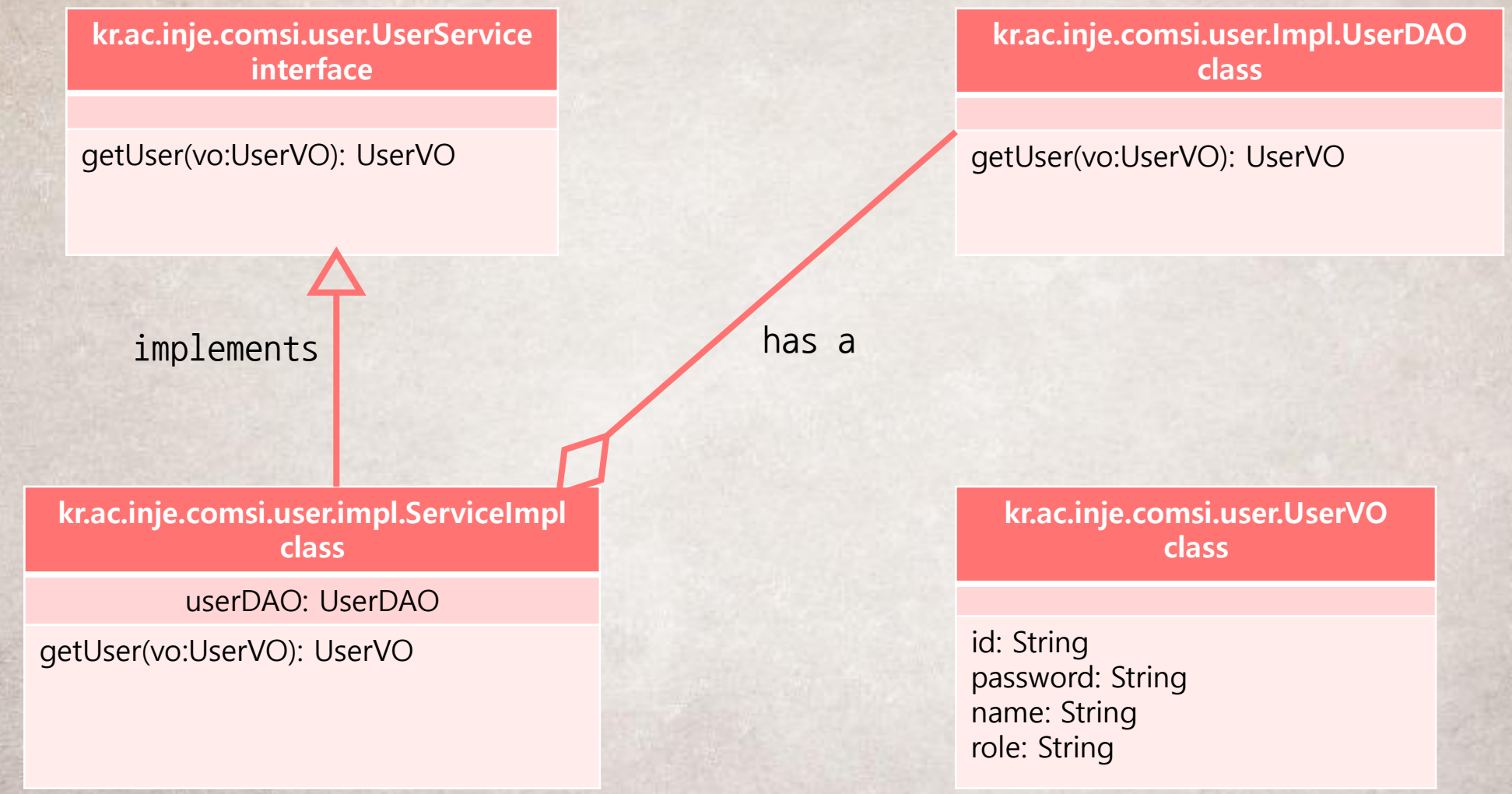
7. 비즈니스 컴포넌트 실습2

- BoardService 컴포넌트를 만들었으면 회원 정보 관리하는 UserService 컴포넌트를 추가로 개발해보자
- BoardService와 동일한 절차로 개발
- Setter 인젝션으로 의존성 주입을 처리한 후 어노테이션으로 변경하는 과정을 사용

프로젝트 구조


Package Explorer
BoardWeb
src/main/java
kr.ac.inje.comsi.board
BoardService.java
BoardServiceClient.java
BoardVO.java
kr.ac.inje.comsi.board.impl
BoardDAO.java
BoardServiceImpl.java
kr.ac.inje.comsi.common
JDBCUtil.java
kr.ac.inje.comsi.user
UserService.java
UserServiceClient.java
UserVO.java
kr.ac.inje.comsi.user.impl
UserDAO.java
UserServiceImpl.java
src/test/java
src/main/resources
applicationContext.xml
JRE System Library [JavaSE-1.8]
Project and External Dependencies
gradle
src
build.gradle
gradlew
gradlew.bat
settings.gradle

1. UserService 컴포넌트 구조



UserService 클래스 다이어그램

2.UserVO.java - 유저 정보 VO 클래스

UserVO.java

```java
package kr.ac.inje.comsi.user;

// VO(Value Object)
public class UserVO {
    private String id;
    private String password;
    private String name;
    private String role;

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getName() {
        return name;
    }

```

```
26      public void setName(String name) {
27          this.name = name;
28      }
29      public String getRole() {
30          return role;
31      }
32      public void setRole(String role) {
33          this.role = role;
34      }
35      @Override
36      public String toString() {
37          return "UserVO [id=" + id + ", password=" + password
38                  + ", name=" + name + ", role=" + role +"]";
39      }
40
41  }
42
43
```

3.DAO 클래스 작성 - JDBCUtil 클래스를 이용

UserServiceImpl.java | applicationContext.xml | UserServiceClient.java | UserDAO.java

```
1  package kr.ac.inje.comsi.user.impl;
2
3  import java.sql.Connection;
9
10 // DAO(Data Access Object)
11 public class UserDAO {
12
13     // JDBC 관련 변수
14     private Connection conn = null;
15     private PreparedStatement stmt = null;
16     private ResultSet rset =null;
17
18     // SQL 명령어들
19     private final String USER_GET = "select * from users where id=? and password=?";
20
21     // CRUD 기능의 메소드
22     // 회원 등록
23     public UserVO getUser(UserVO vo){
24         UserVO user = null;
25         try{
26             System.out.println("===> JDBC로 getUser() 기능 처리");
27             conn = JDBCUtil.getConnection();
28             stmt = conn.prepareStatement(USER_GET);
29             stmt.setString(1, vo.getId());
30             stmt.setString(2, vo.getPassword());
31             rset = stmt.executeQuery();
```

```
32
33              if (rset.next()){
34                  user = new UserVO();
35                  user.setId(rset.getString("ID"));
36                  user.setName(rset.getString("NAME"));
37                  user.setPassword(rset.getString("PASSWORD"));
38                  user.setRole(rset.getString("ROLE"));
39              }
40          }catch(Exception e){
41              e.printStackTrace();
42          }finally{
43              JDBCUtil.close(rset, stmt, conn);
44          }
45
46          return user;
47      }
48
49  }
50
51
```

- UserDAO 클래스는 정상적인 <bean> 등록으로 객체를 생성하고 어노테이션은 설정하지 않는다.

4. Service 인터페이스 작성

UserVO.java | UserDAO.java | UserService.java

```java
package kr.ac.inje.comsi.user;

public interface UserService {

    // CRUD 기능의 메소드
    // 회원 등록
    UserVO getUser(UserVO vo);

}
```

5. Service 구현 클래스 작성

UserVO.java | UserDAO.java | UserService.java | UserServiceImpl.java

```java
package kr.ac.inje.comsi.user.impl;

import kr.ac.inje.comsi.user.UserService;
import kr.ac.inje.comsi.user.UserVO;

public class UserServiceImpl implements UserService {

    private UserDAO userDAO;

    public void setUserDAO(UserDAO userDAO) {
        this.userDAO = userDAO;
    }

    @Override
    public UserVO getUser(UserVO vo) {
        return userDAO.getUser(vo);
    }

}
```

- UserDAO 객체를 이용해 DB 연동 처리
- UserServiceImpl 클래스에는 setter 인젝션 처리를 위한 Setter 메소드가 추가됨.

6. UserService 컴포넌트 테스트

UserVO.java | UserDAO.java | UserService.java | UserServiceImpl.java | applicationContext.xml

```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:p="http://www.springframework.org/schema/p"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="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">

    <context:component-scan base-package="kr.ac.inje.comsi"></context:component-scan>

    <bean id="userService" class="kr.ac.inje.comsi.user.impl.UserServiceImpl">
        <property name="userDAO" ref="userDAO"></property>
    </bean>

    <bean id="userDAO" class="kr.ac.inje.comsi.user.impl.UserDAO"></bean>

</beans>
```

- applicationContext.xml에 UserServiceImpl와 UserDAO 클래스를 <bean> 등록
- UserServiceImpl 클래스에서 UserDAO객체를 의존성 주입하기 위한 <property> 설정 사용

UserServiceClient.java - 클라이언트 프로그램 수정

UserServiceImpl.java | applicationContext.xml | UserServiceClient.java | BoardDAO.java | UserDAO.java

```java
package kr.ac.inje.comsi.user;

import org.springframework.context.support.AbstractApplicationContext;
import org.springframework.context.support.GenericXmlApplicationContext;

public class UserServiceClient {

    public static void main(String[] args) {

        // 1. Spring 컨테이너를 구동
        AbstractApplicationContext  container = new GenericXmlApplicationContext("applicationContext.xml");

        // 2. Spring 컨테이너로부터 UserServiceImpl 객체를 Lookup
        UserService userService = (UserService) container.getBean("userService");

        // 3. 로그인 기능 테스트
        UserVO vo = new UserVO();
        vo.setId("test");
        vo.setPassword("test123");

        UserVO user = userService.getUser(vo);
        if (user != null){
            System.out.println(user.getName() + "님 환영합니다.");
        }else{
            System.out.println("로그인 실패");
        }

        // 4. Spring 컨테이너 종료
        container.close();
    }

}
```

실행 결과


Console
Progress
Problems
<terminated> BoardWeb [Java Application] C:₩Program Files₩Java₩jre1.8.0_121₩bin₩javaw.exe (2017. 5. 28. 오후 11:19:11)
5월 28, 2017 11:19:11 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 28, 2017 11:19:12 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@6e2c634b: startup date [Sun May 28 23:19:12
===> JDBC로 getUser() 기능 처리
관리자님 환영합니다.
5월 28, 2017 11:19:12 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@6e2c634b: startup date [Sun May 28 23:19:12 KST

setter 인젝션을 annotation으로 변경

7. 어노테이션 적용 - Setter 인젝션 설정을 수정

UserServiceImpl.java | applicationContext.xml | UserServiceClient.java | UserDAO.java

```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:p="http://www.springframework.org/schema/p"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="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">

    <context:component-scan base-package="kr.ac.inje.comsi"></context:component-scan>
    <!--
    <bean id="userService" class="kr.ac.inje.comsi.user.impl.UserServiceImpl">
        <property name="userDAO" ref="userDAO"></property>
    </bean>

    <bean id="userDAO" class="kr.ac.inje.comsi.user.impl.UserDAO"></bean>
     -->
</beans>
```

주석처리

UserDAO.java 수정

UserServiceImpl.java | applicationContext.xml | UserServiceClient.java | UserDAO.java

```
1  package kr.ac.inje.comsi.user.impl;
2
3  import java.sql.Connection;
4  import java.sql.PreparedStatement;
5  import java.sql.ResultSet;
6
7  import org.springframework.stereotype.Repository;
8
9  import kr.ac.inje.comsi.common.JDBCUtil;
10 import kr.ac.inje.comsi.user.UserVO;
11
12 // DAO(Data Access Object)
13 @Repository("userDAO")
14 public class UserDAO {
15     // JDBC 관련 변수
16     private Connection conn = null;
17     private PreparedStatement stmt = null;
18     private ResultSet rset =null;
19
20     // SQL 명령어들
21     private final String USER_GET = "select * from users where id=? and password=?";
```

추가됨

UserDAO 객체 생성

UserServiceImpl.java 수정하여 어노테이션 추가

UserServiceImpl.java | applicationContext.xml | UserServiceClient.java | UserDAO.java

```java
package kr.ac.inje.comsi.user.impl;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import kr.ac.inje.comsi.user.UserService;
import kr.ac.inje.comsi.user.UserVO;

@Service("userService")
public class UserServiceImpl implements UserService {
    @Autowired
    private UserDAO userDAO;

    public void setUserDAO(UserDAO userDAO) {
        this.userDAO = userDAO;
    }

    @Override
    public UserVO getUser(UserVO vo) {
        return userDAO.getUser(vo);
    }

}
```

이름이 userService인 Service객체 생성

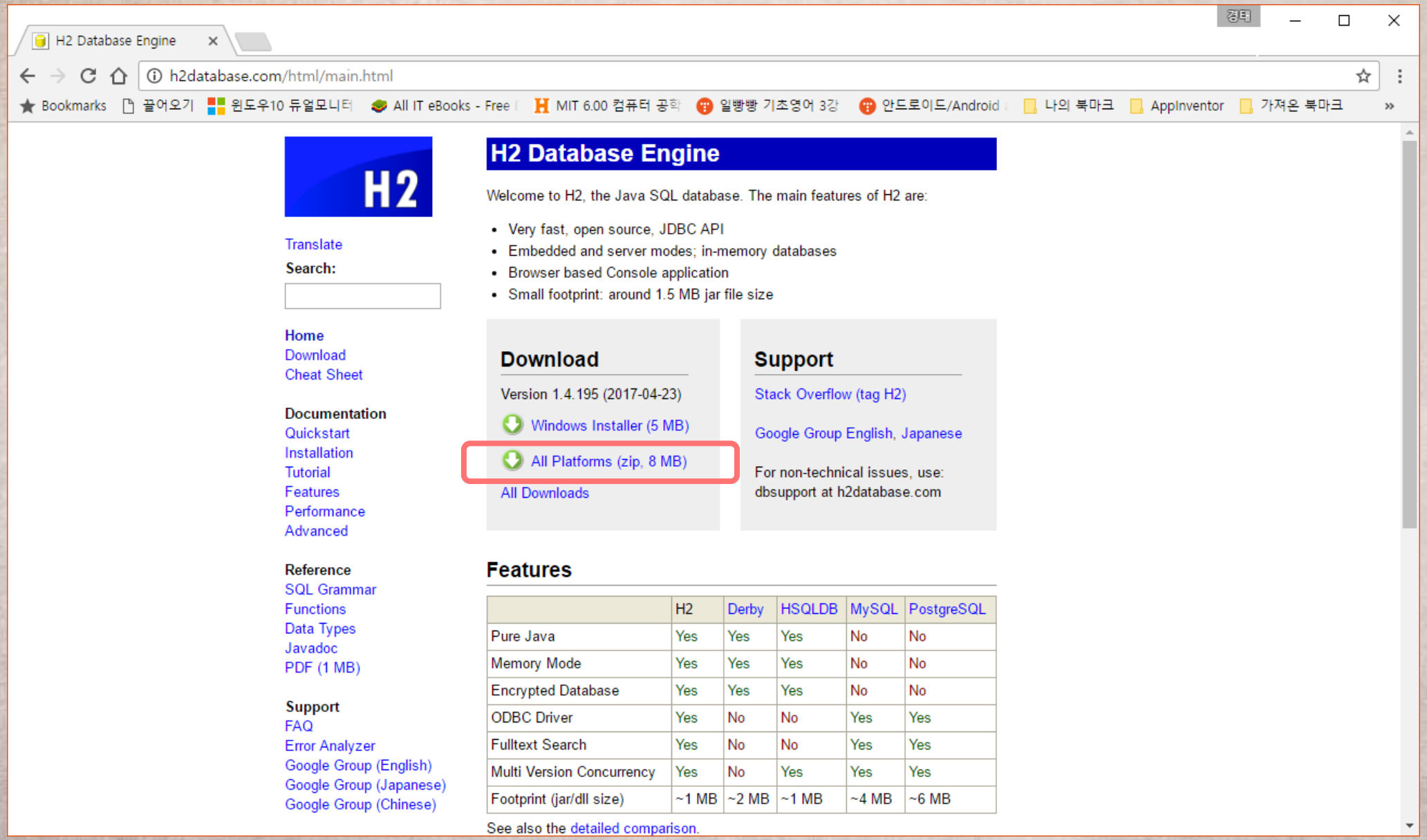
UserDAO 객체 주입

실행 결과

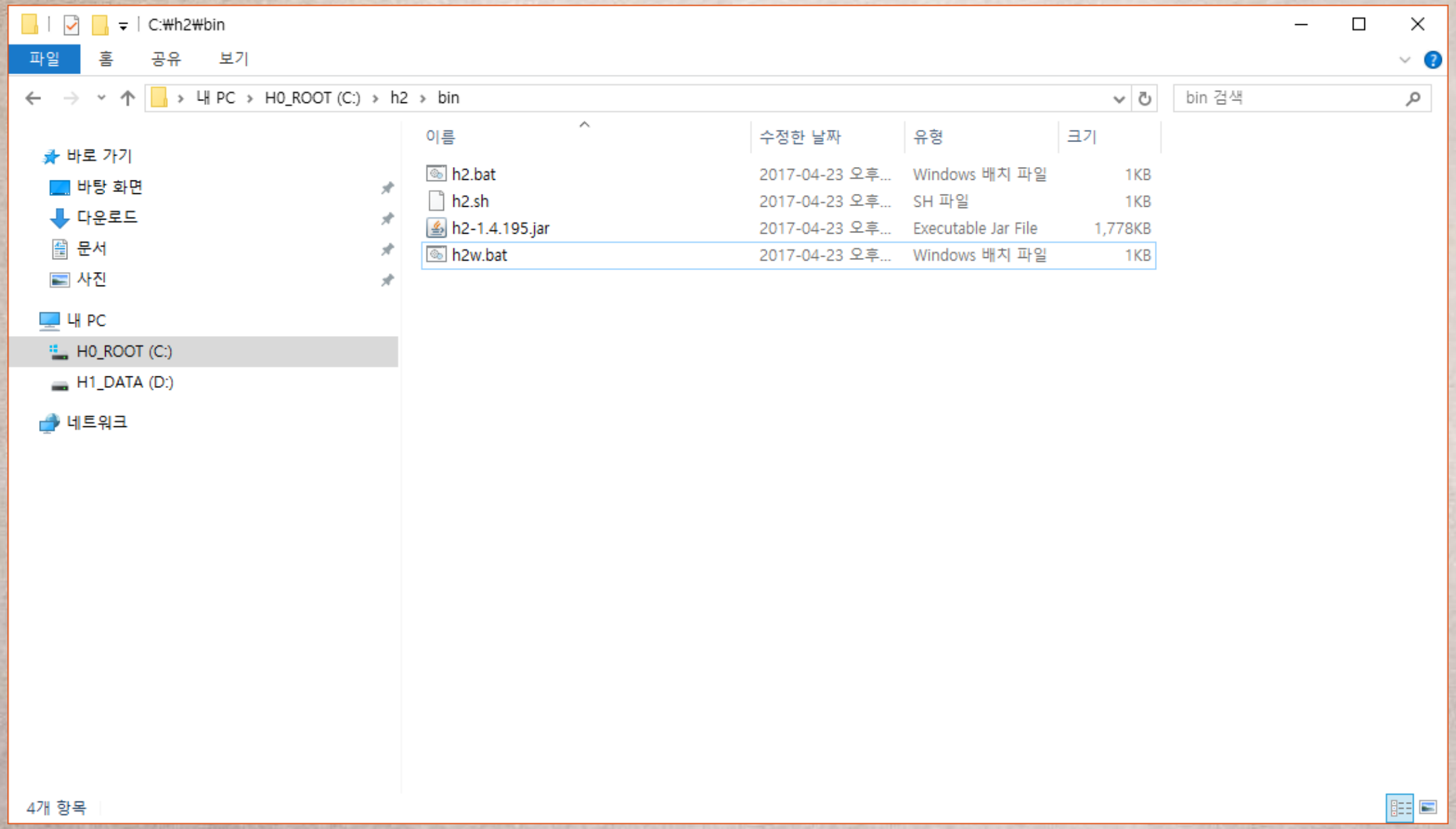

Console
Progress
Problems
<terminated> BoardWeb [Java Application] C:₩Program Files₩Java₩jre1.8.0_121₩bin₩javaw.exe (2017. 5. 28. 오후 11:19:11)
5월 28, 2017 11:19:11 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader loadBeanDefinitions
정보: Loading XML bean definitions from class path resource [applicationContext.xml]
5월 28, 2017 11:19:12 오후 org.springframework.context.support.GenericXmlApplicationContext prepareRefresh
정보: Refreshing org.springframework.context.support.GenericXmlApplicationContext@6e2c634b: startup date [Sun May 28 23:19:12
===> JDBC로 getUser() 기능 처리
관리자님 환영합니다.
5월 28, 2017 11:19:12 오후 org.springframework.context.support.GenericXmlApplicationContext doClose
정보: Closing org.springframework.context.support.GenericXmlApplicationContext@6e2c634b: startup date [Sun May 28 23:19:12 KST

데이터베이스 구축

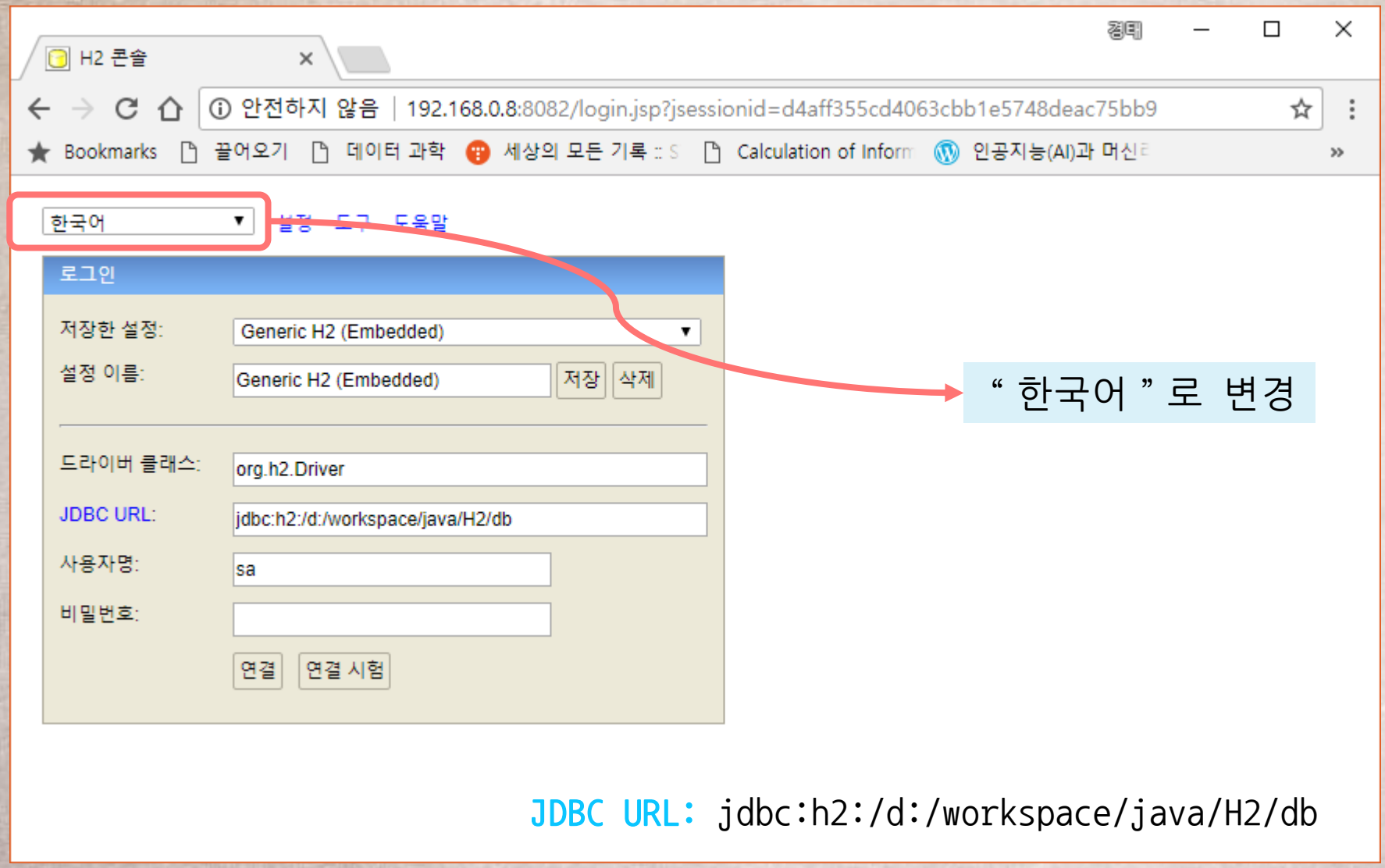
H2-http://h2database.com



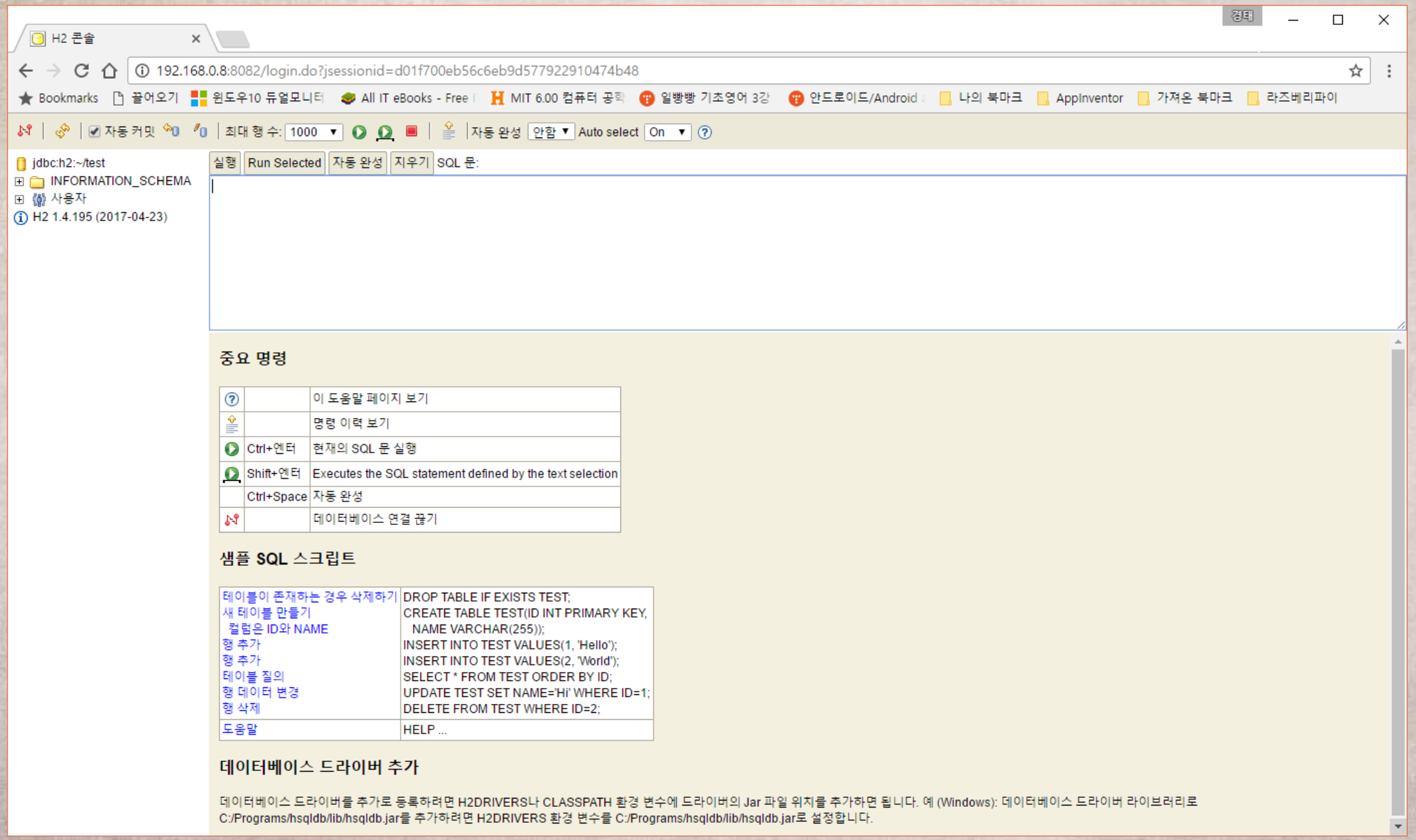
H2Database 실행 - h2→bin>h2w.bat 실행



H2 서버 - 구동


H2 콘솔
안전하지 않음 | 192.168.0.8:8082/login.jsp?jsessionid=d4aff355cd4063cbb1e5748deac75bb9
Bookmarks
끌어오기
데이터 과학
세상의 모든 기록
Calculation of Inform
인공지능(AI)과 머신
한국어
로그인
저장한 설정:
Generic H2 (Embedded)
설정 이름:
Generic H2 (Embedded)
저장
삭제
드라이버 클래스:
org.h2.Driver
JDBC URL:
jdbc:h2:/d:/workspace/java/H2/db
사용자명:
sa
비밀번호:
연결
연결 시험
" 한국어 " 로 변경
JDBC URL: jdbc:h2:/d:/workspace/java/H2/db

H2 서버 - 연결


H2 콘솔
192.168.0.8:8082/login.do?jsessionid=d01f700eb56c6eb9d577922910474b48
Bookmarks
끌어오기
윈도우10 듀얼모니터
All IT eBooks - Free
MIT 6.00 컴퓨터 공학
일빵빵 기초영어 3강
안드로이드/Android
나의 북마크
AppInventor
가져온 북마크
라즈베리파이
자동 커밋
최대 행 수: 1000
자동 완성
안함
Auto select
On
jdbc:h2:~/test
INFORMATION_SCHEMA
사용자
H2 1.4.195 (2017-04-23)
실행
Run Selected
자동 완성
지우기
SQL 문:
중요 명령
이 도움말 페이지 보기
명령 이력 보기
Ctrl+엔터
현재의 SQL 문 실행
Shift+엔터
Executes the SQL statement defined by the text selection
Ctrl+Space
자동 완성
데이터베이스 연결 끊기
샘플 SQL 스크립트
테이블이 존재하는 경우 삭제하기
DROP TABLE IF EXISTS TEST;
새 테이블 만들기
컬럼은 ID와 NAME
CREATE TABLE TEST(ID INT PRIMARY KEY,
NAME VARCHAR(255));
행 추가
INSERT INTO TEST VALUES(1, 'Hello');
행 추가
INSERT INTO TEST VALUES(2, 'World');
테이블 질의
SELECT * FROM TEST ORDER BY ID;
행 데이터 변경
UPDATE TEST SET NAME='Hi' WHERE ID=1;
행 삭제
DELETE FROM TEST WHERE ID=2;
도움말
HELP ...
데이터베이스 드라이버 추가
데이터베이스 드라이버를 추가로 등록하려면 H2DRIVERS나 CLASSPATH 환경 변수에 드라이버의 Jar 파일 위치를 추가하면 됩니다. 예 (Windows): 데이터베이스 드라이버 라이브러리로 C:/Programs/hsqldb/lib/hsqldb.jar를 추가하려면 H2DRIVERS 환경 변수를 C:/Programs/hsqldb/lib/hsqldb.jar로 설정합니다.

H2 서버 - 테이블 생성 및 데이터 입력코드(sql문)

USER 테이블

```
CREATE TABLE USERS(
        ID VARCHAR2(8) PRIMARY KEY,
        PASSWORD VARCHAR2(8),
        NAME VARCHAR2(20),
        ROLE VARCHAR2(5)
);

INSERT INTO USERS VALUES('test','test123','관리자','Admin');
INSERT INTO USERS VALUES('user1','user1','홍길동','user');
```

BOARD 테이블

```
CREATE TABLE BOARD(
        SEQ NUMBER(5) PRIMARY KEY,
        TITLE VARCHAR2(200),
        WRITER VARCHAR2(20),
        CONTENT VARCHAR2(2000),
        REGDATE DATE DEFAULT SYSDATE,
        CNT NUMBER(5) DEFAULT 0
);

INSERT INTO BOARD(SEQ, TITLE, WRITER, CONTENT) VALUES(1,'가입인사','관리자','잘 부탁드립니다.....');
```

H2 서버 - 입력 데이터 확인