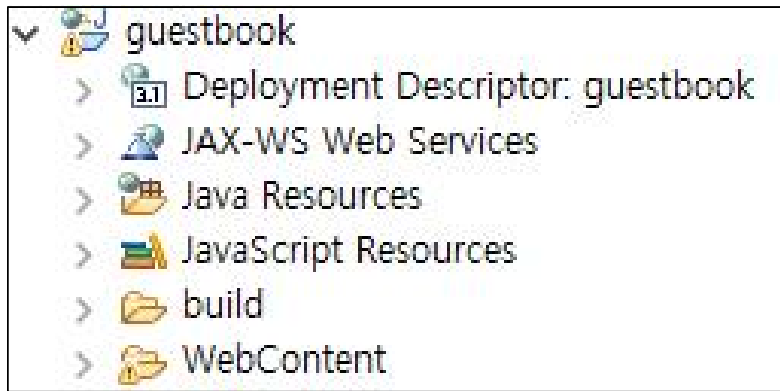


■ ‘guestbook’ 프로젝트 생성



- com.jbm.guestbook.xxx로 패키지 생성

■ guestbook 테이블 생성

Column Name	Data Type	Nullable	Default	Primary Key
NO	NUMBER(8,0)	No	-	1
WRITER	VARCHAR2(100)	No	-	-
CONTENTS	VARCHAR2(420)	No	-	-
REGDATE	TIMESTAMP(6)	No	-	-
IP	VARCHAR2(15)	No	-	-
1 - 5				

■ guestbook_seq 시퀀스 생성

■ WEB-INF → lib 폴더에 ojdbc / mybatis 라이브러리 가져다 놓기



- 우리가 Java SE에서 수행할 때는 무조건 자기 컴퓨터에서 작동되기 때문에 class path만 등록해도 되지만, 웹 프로그래밍은 내 컴퓨터가 아닌 다른 컴퓨터의 웹서버에서 작동되므로 직접 필요한 라이브러리를 넣어야 합니다.

■ META-INF폴더 → context.xml 에 커넥션풀 설정



```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
    <Resource name="jdbc/oracle"
        auth="Container"
        driverClassName="oracle.jdbc.OracleDriver"
        type="javax.sql.DataSource"
        url="jdbc:oracle:thin:@localhost:1521:xe"
        username="test" password="1111"
        maxIdle="20" maxActive="20" maxWait="-1"
    />
</Context>
```

■ Guest V0생성

```
import java.sql.Timestamp;

public class Guest {

    private int no;
    private String writer,contents,ip;
    private Timestamp regdate;

    public Guest() {
        // TODO Auto-generated constructor stub
    }

    public Guest(String writer,
                  String contents,
                  String ip) {
        this.writer = writer;
        this.contents = contents;
        this.ip = ip;
    }
    // getter / setter 생략
}
```

■ SqlSessionUtil 제작

```
package com.mhd.guestbook.util;

import java.io.Reader;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;

public class SqlSessionUtil {

    private static SqlSessionFactory factory;

    public static SqlSession getSession() {
        return factory.openSession(true);
    }

    static {
        try {
            Reader reader =
Resources.getResourceAsReader("com/jbm/guestbook/config/mybatis-config.xml");

            factory = new SqlSessionFactoryBuilder().build(reader);

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

    } //static

} //SqlSessionUtil class end
```

■ mybatis-config 파일 생성

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org/DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <typeAliases>
    <typeAlias type="com.jbm.guestbook.vo.Guest" alias="Guest"/>
  </typeAliases>

  <!-- 환경설정(DB연결) -->
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC"/>
      <dataSource type="JNDI">
        <property name="data_source"
          value="java:comp/env/jdbc/oracle"/>
      </dataSource>
    </environment>
  </environments>

  <!-- mapper파일들에 대한 설정 -->
  <mappers>
    <mapper resource="com/jbm/guestbook/map/guestbook.xml"/>
  </mappers>

</configuration>
```

■ mapper 파일 생성

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="guestbook">

    <select id="selectList" resultType="Guest">
        SELECT no,writer,contents,regdate,ip
        FROM guestbook
        ORDER BY no DESC
    </select>

    <select id="selectOne" parameterType="int" resultType="Guest">
        SELECT no,writer,contents,regdate,ip
        FROM guestbook
        WHERE no=#{no}
    </select>

    <insert id="insert" parameterType="Guest">
        INSERT INTO guestbook(no,writer,contents,regdate,ip)
        VALUES(guestbook_seq.nextval,#{writer},
                #{contents},systimestamp,#{ip})
    </insert>

    <delete id="delete" parameterType="int">
        DELETE guestbook
        WHERE no = #{no}
    </delete>

    <update id="update" parameterType="Guest">
        UPDATE guestbook SET contents = #{contents}
        WHERE no=#{no}
    </update>

</mapper>
```

■ GuestbookDAO 생성

```
package com.mhd.guestbook.dao;

import java.util.List;

import org.apache.ibatis.session.SqlSession;

import com.mhd.guestbook.util.SqlSessionUtil;
import com.mhd.guestbook.vo.Guest;

public class GuestbookDAO {

    public static Guest selectOne(int no) {
        Guest guest = null;

        SqlSession session = null;

        try {
            session = SqlSessionUtil.getSession();
            guest = session.selectOne("guestbook.selectOne",no);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            session.close();
        }

        return guest;
    } //selectList() end

    public static List<Guest> selectList() {
        List<Guest> list = null;
        SqlSession session = null;

        try {
            session = SqlSessionUtil.getSession();
            list = session.selectList("guestbook.selectList");
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            session.close();
        }
    }
}
```

```

        return list;
    }//selectList() end

    public static void insert(Guest guest) {

        SqlSession session = null;

        try {
            session = SqlSessionUtil.getSession();
            session.insert("guestbook.insert",guest);

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            session.close();
        }//try~catch~finally end

    }//insert() end

    public static void delete(int no) {

        SqlSession session = null;

        try {
            session = SqlSessionUtil.getSession();

            session.delete("guestbook.delete",no);

        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            session.close();
        }//try~catch~finally end

    }//delete() end

    public static void update(Guest guest) {

        SqlSession session = null;

        try {

```

```
        session = SqlSessionUtil.getSession();

        session.delete("guestbook.update", guest);

    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        session.close();
    } // try~catch~finally end

} // update() end

} // GuestbookDAO class end
```

■ 템플릿의 활용

- <%@ include %> 지시어의 활용

```
<%@ include file="WEB-INF/templates/header.jsp" %>
```

■ 필터의 활용

■ HTML 파일 만들기(UI제작)

- 글이 하나도 없을때



- 글쓰기 페이지

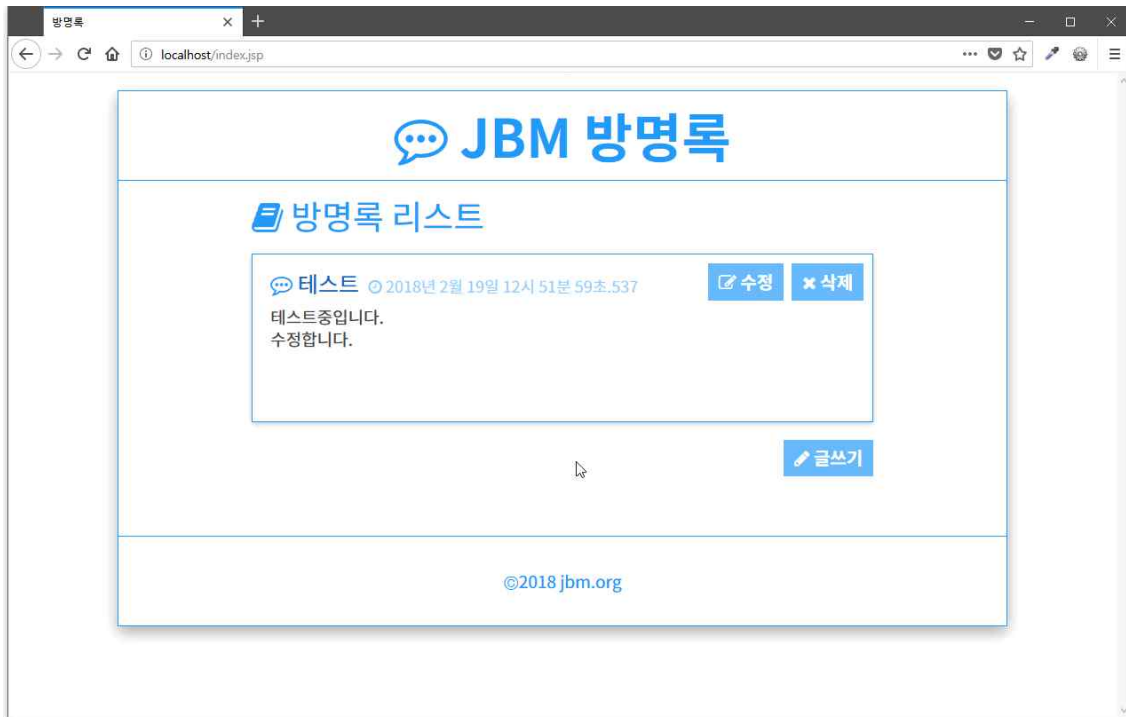


- 글을 쓰고 난 후(자신이 쓴 글의 경우 수정, 삭제 버튼이 보임)



- 글수정 페이지(작성자는 변경불가 / 내용이 출력되어 있음)





- 다른 컴퓨터에서 쓴 글은 수정 / 삭제버튼이 보이지 않음



- 글쓴이를 클릭하면 글쓴이의 방명록으로 링크 이동

■ 페이지 구조도

