학생관리프로그램

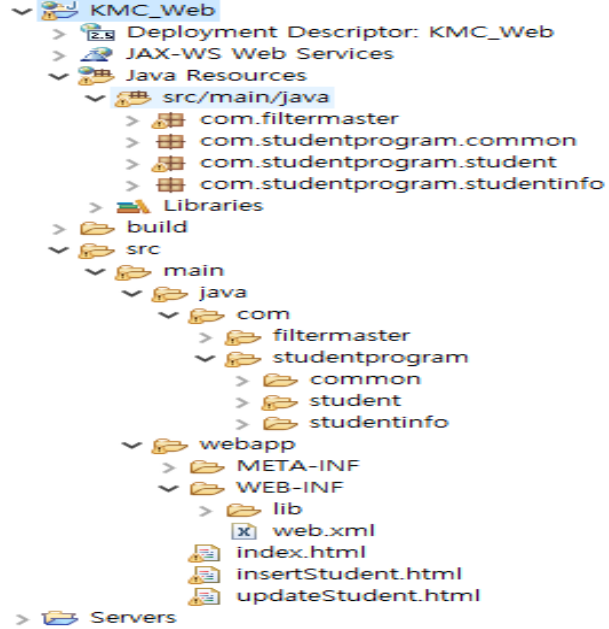
백앤드_27회차

김명찬

2024/09/20

- 목차

- 1. 프로젝트 구조 설명
- 2. 데이터베이스 설명
- 3. 기능설명
- 4.예외처리 설명



- 프로젝트 구조 설명

- com.filtermaster : 필터기능을 사용하여 구현한 클래스들을 담고 있습니다.
- com.studentprogram.common : h2 데이터베이스 객체 로딩 후 연결을 위한 클래스를 담고 있습니다.
- com.studentprogram.student : 동작처리를 위한 서블릿 클래스를 담고 있습니다.
- com.studentprogram.studentinfo : 실직적인 데이터베이스 연동 작업을 담당하는 클래스와 변수를 관리하는 클래스를 담고 있습니다.

show columns from student;

FIELD	TYPE	NULL	KEY	DEFAULT
NAME	CHARACTER VARYING(6)	NO		NULL
STDNUM	CHARACTER VARYING(8)	NO	PRI	NULL
MAJOR	CHARACTER VARYING(12)	NO		NULL
PHONENUM	CHARACTER VARYING(11)	NO		NULL

(4 행, 18 ms)

- 2. 데이터베이스 설명

- name : 학생의 이름을 저장하기 위한 컬럼
- stdnum : 학번을 저장하기 위한 컬럼
- major : 학과를 저장하기 위한 컬럼
- phonenum : 전화번호 저장을 위한 컬럼

```java
public class JDBCUtil {


    public static Connection getConnection() {
        Connection conn = null;

        try {
            //JDBC 1단계 : 드라이버 객체 로딩
            DriverManager.registerDriver(new org.h2.Driver());

            //JDBC 2단계 : 커넥션 연결
            String jdbcUrl = "jdbc:h2:tcp://localhost/~/test";
            conn = DriverManager.getConnection(jdbcUrl,"sa", "");

        }

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

        return conn;

    }
```

```java
public static void close(PreparedStatement stmt , Connection conn ) {

    //JDBC 연결 해제

    try {
        stmt.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        conn.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public static void close(ResultSet rs,PreparedStatement stmt, Connection conn) {

    try {
        stmt.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        conn.close();
    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    try {
        rs.close();
    }
    catch(SQLException e) {
        e.printStackTrace();
    }
}
```

- JDBCUtil 클래스의 getConnection 메소드에서 WEB-INF 파일에 있는 H2데이터베이스 드라이버를 로딩 후
- DB의 주소와 DB의 아이디 비밀번호를 입력 후 연결하는 동작을 수행합니다.

- 연결 후에는 close 메소드를 통해 ResultSet, PreparedStatement, Connection 을 닫아줍니다. (메모리와 시스템 자원 낭비를 방지하기 위해서)

```java
//학생등록
public void insertStudent(StudentVO vo) {
    // TODO Auto-generated method stub


    try {


        conn = JDBCUtil.getConnection();  //DB에 연결
        stmt = conn.prepareStatement(STUDENT_INSERT); //stmt에 쿼리문을 담는다.

        stmt.setString(1, vo.getName());
        stmt.setString(2, vo.getStdnum());
        stmt.setString(3, vo.getMajor());
        stmt.setString(4, vo.getPhonenum());
        // 쿼리문의 ? 에 값을 넣기 위해 ? 의 순서에 대한 인덱스와 담을 값을 지정


        stmt.executeUpdate();
    }
    catch (SQLException e) {
        // TODO: handle exception0
        e.printStackTrace();
    }

    finally {
        JDBCUtil.close(stmt, conn);
    }
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String name = request.getParameter("name");
    String stdnum = request.getParameter("stdnum");
    String major = request.getParameter("major");
    String phonenum = request.getParameter("phonenum");


    StudentVO vo = new StudentVO();
    StudentDAO dao = new StudentDAO();

    vo.setName(name);
    vo.setMajor(major);
    vo.setPhonenum(phonenum);
    vo.setStdnum(stdnum);

    dao.insertStudent(vo);

    response.sendRedirect("/");

}
```

```
@WebServlet("/insert.do")
public class InsertStudentServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub

        String name = request.getParameter("name");
        String stdnum = request.getParameter("stdnum");
        String major = request.getParameter("major");
        String phonenum = request.getParameter("phonenum");


        StudentVO vo = new StudentVO();
        StudentDAO dao = new StudentDAO();

        vo.setName(name);
        vo.setMajor(major);
        vo.setPhonenum(phonenum);
        vo.setStdnum(stdnum);

        dao.insertStudent(vo);

        response.sendRedirect("/");

    }
}
```

- 3. 기능설명

- 3-1(학생등록)

- InsertStudentServlet 에서는 insertStudent.html 에서 post 방식으로 보내온 데이터를 request 를 통해 가져옵니다.
- StudentVO 를 통해 입력받은 데이터를 VO 객체에 저장하고 DAO 클래스의 insertStudent 메소드의 매개변수로 보냅니다.
- StudentDAO 의 insertStudent 메소드를 통해 입력받은 데이터들을 데이터베이스에 넣는 작업을 수행합니다.

```java
public List<StudentVO> getAllStudentList() {
    // TODO Auto-generated method stub
    List<StudentVO> stdList = new ArrayList<StudentVO>();
    // StudentVO 타입을 담을 배열 리스트 선언

    try {
        conn = JDBCUtil.getConnection();
        stmt = conn.prepareStatement(ALLSTUDENT_SELECT);
        rs = stmt.executeQuery();  // SELECT 는 다른 쿼리들과 다르게 ResultSet 를 사용해야 함.

        while(rs.next()) {
            StudentVO vo = new StudentVO();

            vo.setName(rs.getString("NAME"));
            vo.setStdnum(rs.getString("STDNUM"));
            vo.setMajor(rs.getString("MAJOR"));
            vo.setPhonenum(rs.getString("PHONENUM"));
            //rs.next 를 통해서 데이터베이스에 존재하는 학생정보들을 vo 객체에 담은 뒤에

            stdList.add(vo); // 리스트에 추가
        }

    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    finally {
        JDBCUtil.close(rs, stmt, conn);
        //메모리와 시스템의 자원낭비를 방지하기 위해 사용 후에는 닫아줌.
    }

    return stdList;
}
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    StudentVO vo = new StudentVO();
    StudentDAO dao = new StudentDAO();
    PrintWriter out = response.getWriter();

    List<StudentVO> stdList = dao.getAllStudentList();

    out.println("<html>");



    out.println("<head>");
    out.println("<title>studentInfo</title>");
    out.println("</head>");

    out.println("<body>");

    out.println("<center>");

    out.println("<h1>학생정보</h1>");
    out.println(" <form action='getStudent.do' method='post' >  ");
    out.println("학번검색 : <input type='text' name='stdnum'> </input>");
    out.println("<input type='submit' value='전송'> </input>");
    out.println("</form>");


    out.println("<table border='1' cellpadding='0' cellspacing='0' width='700'>");

    out.println("<tr>");
    out.println("<th bgcolor='orange' width='100'>이름</th>");
    out.println("<th bgcolor='orange' width='200'>학번</th>");
    out.println("<th bgcolor='orange' width='150'>학과</th>");
    out.println("<th bgcolor='orange' width='150'>전화번호</th>");
    out.println("</tr>");
```

```
    for (StudentVO stdVO : stdList) {
        out.println("<tr>");
        out.println("<td>" + stdVO.getName()   + "</td>");
        out.println("<td>" + stdVO.getStdnum()   + "</td>");
        out.println("<td>" + stdVO.getMajor()   + "</td>");
        out.println("<td>" + stdVO.getPhonenum()   + "</td>");
        out.println("</tr>");
    }

    out.println("</table>");


    out.println("<br>");
    out.println("<a href='/'>처음으로</a>");



    out.println("</center>");
    out.println("</body>");
    out.println("</html>");

    out.close();




}
}
```

- 3-2(학생정보 전체 출력)

- GetAllStudentServlet 서블릿에서는 html에서 (<td> 전체출력 </td>) 메소드를 따로 지정해주지 않았기 때문에 get메소드로 설정이 되어 doGet에서 동작을 정의했습니다.

- DAO 클래스의 getAllStudentList 메소드를 통해서 전체 학생정보가 저장된 배열 리스트를 받아 for 문을 통해 저장되어 있는 학생정보들을 출력합니다.

```java
//학생찾기
public List<StudentVO> getStudent(StudentVO vo) {
    // TODO Auto-generated method stub
    List<StudentVO> stdList = new ArrayList<StudentVO>();

    try {
        conn = JDBCUtil.getConnection();
        stmt = conn.prepareStatement(STUDENT_SELECT);

        stmt.setString(1, vo.getStdnum());

        rs = stmt.executeQuery();


        while(rs.next()) {
            StudentVO stdVO = new StudentVO();

            stdVO.setName(rs.getString("NAME"));
            stdVO.setStdnum(rs.getString("STDNUM"));
            stdVO.setMajor(rs.getString("MAJOR"));
            stdVO.setPhonenum(rs.getString("PHONENUM"));

            stdList.add(stdVO);

        }


    } catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    finally {
        JDBCUtil.close(rs, stmt, conn);
    }


    return stdList;
}
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub

    String stdNum = request.getParameter("stdnum");


    StudentVO vo = new StudentVO();
    StudentDAO dao = new StudentDAO();
    PrintWriter out = response.getWriter();

    vo.setStdnum(stdNum);
    List<StudentVO> stdList = dao.getStudent(vo);

    out.println("<html>");



    out.println("<head>");
    out.println("<title>studentInfo</title>");
    out.println("</head>");

    out.println("<body>");

    out.println("<center>");

    out.println("<h1>학생정보</h1>");

    out.println("<table border='1' cellpadding='0' cellspacing='0' width='700'>");

    out.println("<tr>");
    out.println("<th bgcolor='orange' width='100'>이름</th>");
    out.println("<th bgcolor='orange' width='200'>학번</th>");
    out.println("<th bgcolor='orange' width='150'>학과</th>");
    out.println("<th bgcolor='orange' width='150'>전화번호</th>");
    out.println("</tr>");
```

```
    for (StudentVO stdVO : stdList) {
        out.println("<tr>");
        out.println("<td>" + stdVO.getName()   + "</td>");
        out.println("<td>" + stdVO.getStdnum()   + "</td>");
        out.println("<td>" + stdVO.getMajor()   + "</td>");
        out.println("<td>" + stdVO.getPhonenum()   + "</td>");
        out.println("</tr>");
    }


    out.println("</table>");



    out.println("<a href='/'>처음으로</a>");
    out.println("<a href='updateStudent.html'>정보수정</a>");

    out.println("</form>");

    out.println("</center>");



    out.println("</body>");
    out.println("</html>");

    out.close();


}
}
```

```
out.println(" <form action='getStudent.do' method='post' >  ");
out.println("학번검색 : <input type='text' name='stdnum'> </input>");
out.println("<input type='submit' value='전송'> </input>");
out.println("</form>");
```

- 3-3(학생정보 출력)

- GetAllStudentServlet에서 form 태그안에 검색할 학생의 학번을 입력하면 데이터가 getStudent.do로 post 방식으로 보내집니다.

- GetAllStudentServlet에서 입력된 html 태그와 동일하지만 학번과 일치하는 학생정보만 출력이 되고 학번과 일치하는 학생정보가 없다면 아무것도 출력되지 않습니다.

```java
@WebServlet("/updateStudent.do")
public class updateStudentServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;


    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        StudentVO vo = new StudentVO();
        StudentDAO dao = new StudentDAO();
        PrintWriter out = response.getWriter();

        String stdNum = request.getParameter("stdnum");
        String name = request.getParameter("name");
        String major = request.getParameter("major");
        String phoneNum = request.getParameter("phonenum");

        vo.setName(name);
        vo.setStdnum(stdNum);
        vo.setMajor(major);
        vo.setPhonenum(phoneNum);

        dao.updateStudent(vo);

        response.sendRedirect("/");

    }
```

```java
public void updateStudent(StudentVO vo) {
    // TODO Auto-generated method stub

    try {
        conn = JDBCUtil.getConnection();
        stmt = conn.prepareStatement(STUDENT_UPDATE);

        stmt.setString(1, vo.getName());
        stmt.setString(2, vo.getPhonenum());
        stmt.setString(3, vo.getStdnum());

        stmt.executeUpdate();

    }

    catch (SQLException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }

    finally {
        JDBCUtil.close(stmt, conn);
    }


}
```

- **3-4(정보수정)**

- Post방식으로 데이터를 받아 다른 서블릿과 동일하게 받은 변수들을 vo 객체에 저장하고 dao 클래스의 메소드를 통해서 실행하려는 동작을 처리합니다.
- 다른 서블릿과 동일하게 DB 연결 후 지정한 쿼리문의 ? 에 값을 넣어서 쿼리문을 실행합니다.

- 동작을 실행한 후
- Response.sendRedirect를 통해 "/" 루트로 지정한 페이지로 이동하게 됩니다. (web.xml 의 welcome 페이지)

```
try {
    conn = JDBCUtil.getConnection();
    stmt = conn.prepareStatement(STUDENT_UPDATE);

    stmt.setString(1, vo.getName());
    stmt.setString(2, vo.getPhonenum());
    stmt.setString(3, vo.getStdnum());

    stmt.executeUpdate();

}

catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

finally {
    JDBCUtil.close(stmt, conn);
}
```

- **4. 예외처리 설명**

- 데이터베이스와의 연결 과정에서 발생할 수 있는 예외를 처리하기 위한 구문입니다. try 블록에서 실행되는 코드에서 문제가 발생하면, 그 예외가 catch 블록으로 넘어가고, 여기서 예외를 처리합니다.

- (SQLException은 JDBC 작업을 수행할 때 발생하는 예외입니다. 데이터베이스 연결 실패, SQL 문법 오류, 제약 조건 위반 등과 같은 이유로 발생할 수 있습니다.)