웹프로젝트 기말고사 Ver.2 (20241519 조예성)

boardDao의 메소드: getBoardList, getBoard, addBoard updateBoard

- Dao는 (SQL)에 직접 접속한다

Select 문 후 가져오는 메소드: rs.get자료형("컬럼")

jdbc 불러오는 코드: Class.forname("com.mysql.cj.jdbc.Driver")

DB 연결 코드: Connection connection = DriverManager.getConnection(uri, id, pwd)

- 메소드 제작시 리턴 객체: Connection
- Url 간단하게 포메팅: String.format()
- 필요 요서: host, port, dbInstance, id, pw

DB 전체 반환 리턴하는 메소드 자료형: List<Dto>

- List<BoardDto> list = new ArrayList<>();
- sql문 준비 및 저장: PreparedStatement.prepareStatement("SQL문")
- Select 문 실행 및 결과 저장 객체: ResultSet rs = preparedStatement.executeQuery()
- rs에서 반환 값 받기: rs.get자료형("컬럼명")
- 반환은 Dto로 이루어진 List를 반환한다

id참조하여 BoardDto 반환 (boardId 매개변수, getBoardList는 BoardDto의 리스트를 저장하는 객체):

```
BoardDto Board = getBoardList().stream()
       .filter(e -> e.getId() == boardId)
       .findFirst()
       .orElse(null);
```

- ? 형태에 SQL 값 집어넣기: preparedStatement.setString(index, String);
- insert, update 실행: preparedStatement.executeUpdate();
- 위 코드 반환값: int

update 문 형태: update Database set 기존1 = 변경1, 기존2 = 변경2 where 변경할 곳 = 조건;

Dao객체 메소드의 반환값은 대부분 Dto이다

로그인 등 각종 비즈니스 로직 구현: 서비스

URL에서 값 가져오는 jsp 코드: request.getParameter("key")

데이터 한글 깨짐 방지 위하여: request.setCharacterEncoding("utf-8")

서비스는 확장성 위하여 (인터페이스) 객체에 (구현체) 삽입

lombok @들: @Getter @setter @ToString @AllArgsCoustructor @NoArgsCounstrucor @Builder

ip 가져오는 request: getRemoteAddr()

```java
public MemberDto getMember(String userId) {

    MemberDto member = getMemberList().stream()
            .filter(e -> e.getUserId().equals(userId))
            .findFirst()
            .orElse(null);

    return member;
}
```

```java
public boolean addMember(MemberDto member) {

    boolean result = false;

    Connection connection = DbConnector.getConnection();

    //쿼리실행
    String sql = " insert into member (user_id, user_name, password) values (?, ?, ?)
";
    try {
        PreparedStatement preparedStatement = connection.prepareStatement(sql);
        preparedStatement.setString(1, member.getUserId());
        preparedStatement.setString(2, member.getUserName());
        preparedStatement.setString(3, member.getPassword());
        int affectedRow = preparedStatement.executeUpdate();
        if (affectedRow > 0) {
            result = true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return result;
}
```

```java
public List<MemberDto> getMemberList() {

    List<MemberDto> memberList = new ArrayList<>();
    Connection connection = DbConnector.getConnection();

    //쿼리실행
    String sql = " select user_id, user_name, password, create_dt, update_dt from
member order by update_dt desc, create_dt desc ";
    try {
        PreparedStatement preparedStatement = connection.prepareStatement(sql);
        ResultSet rs = preparedStatement.executeQuery();

        while (rs.next()) {
            MemberDto member = MemberDto.builder()
                    .userId(rs.getString("user_id"))
                    .userName(rs.getString("user_name"))
                    .password(rs.getString("password"))
                    .createDt(rs.getDate("create_dt"))
                    .updateDt(rs.getDate("update_dt"))
                    .build();
            memberList.add(member);
        }

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

    return memberList;
}
```

```java
@Override
public List<BoardDto> getBoardList() {
    BoardDao boardDao = new BoardDao();
    List<BoardDto> boardList = boardDao.getBoardList();

    return boardList;
}
```