웹프로젝트 기말고사 (20241519 조예성)

쿠키는 ( )랑 ( )로 구성된다

쿠키의 최대 크기: kb

쿠키 최대 개수: 개

DB 접속을 위하여 필요한 5가지: , , , ,

예외처리 2종류 ,

데이터를 처리하는 객체:

데이터 저장하는 객체:

한번에 처리해야 하는 업무 단위:

트랜잭션의 단위:

```
MemberDao memberDao = new MemberDao();
List<            > memberList =             .getMemberList();

int totalCount = memberList.size();
int i = 0;
for(            member : memberList) {
    StringBuilder sb = new StringBuilder();
```

```
public static Connection getConnection() {
    try {
        //디비저장- 드라이버클래스 로드
        //Class.forName("org.mariadb.jdbc.Driver");
        Class.forName("                              ");

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

    //DB 연결객체생성
    String host = "223.194.169.121";
    int port = 3306;
    String id = "jsp_user";
    String pwd = "1234";
    String dbInstance = "jsp_db";
    //String url = String.format("jdbc:mariadb://%s:%d/%s",  host, port, dbInstance);
    String url = String.format("              s:%d/%s",       ,         ,     );
                    connection = null;

    try {
        connection =                .getConnection(url, id, pwd);
    } catch (SQLException e) {
        e.printStackTrace();
    }

    return connection;
}
```

```
public MemberDto getMember(String userId) {

            member = getMemberList().stream()
          .filter( e-> e.getUserId().equals(userId))
          .                ()//첫 값 찾기
          .orElse(null);

    return member;
}
```

```
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 = connection.                    (    );
        preparedStatement.          (1, member.getUserId());
        preparedStatement.setString(2, member.getUserName());
        preparedStatement.setString(3, member.getPassword());
        int affectedRow = preparedStatement.                  ();
        if(affectedRow > 0) {
            result = true;
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return result;
}
```

SELECT 문 실행:

이외 나머지 문 실행:

preparedStatement 중 ? 에 추가하는 메소드: (,);

```
@G
@S
@B
@All
@No
@To
public class MemberDto {
    String userId;
    String userName;
    String password;
    Date createDt;
    Date updateDt;
}
```

```
request.                              ("utf-8");

String writer = request.              ("writer");
String subject = request.               ("subject");
String contents = request.                ("contents");
String ipAddr = request.                  (); //ip 주소 가져오기

BoardDto board = BoardDto.          ()
       .writer(writer)
       .subject(subject)
       .contents(contents)
       .ipAddr(ipAddr)
       .build();
BoardDao boardDao = new BoardDao();
boolean result = boardDao.addBoard(board);
```

```
String userId =           .getParameter("user_id");

MemberDao memberDao = new MemberDao();
MemberDto member = memberDao.               (userId); //member 갖고오기
```

```
//로그인 정보를 세션에 저장!!!
         .                ("userId", userId);
       .setAttribute("userName", member.getUserName());
//로그인한 사람의 이름정보도 세션에 씀.

//하고, 페이지를 메인으로 보냄
response.                  ("/");
```

```
MemberDao memberDao = new MemberDao();
int memberCount = memberDao.getMemberList().size();

BoardDao boardDao = new BoardDao();
int boardCount = boardDao.getBoardList().size();
// memberCount랑 bbsContentsCoutn 빌딩
return DashboardDto.builder()
       .memberCount(memberCount)
       .bbsContentsCount(boardCount)
       .build();
```

```
//LoginService loginService = new FakeLoginServiceImpl();
LoginService loginService = new DbLoginServiceImpl();
// post 방식에서 가져오기
String userId =           .getParameter("user_id");
String password =           .getParameter("password");

MemberDto member = loginService.login(userId, password);
```

```
TABLE Users (
    user_id INT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    email VARCHAR(100),
    age INT
);
```

```
[redacted] Users (user_id, username, email, age)
[redacted] (1, 'yesung', 'yesung@example.com', 25);
```

```
UPDATE Users
[redacted] email = 'newemail@example.com'
[redacted] user_id = 1;
```