

개인 DB 상세 정보 가져오기

DB

- 모든 DB 정보에서 PK로 등록된 정보를 클릭하면 상세 페이지로 전환을 목표로 함
- a 태그와 '?'를 이용해 PK 정보를 유지한채 이동

```
<td align="center" width="150">  
<a href="MemberInfo.jsp?id=<%= bean.getId() %>">
```

- 이렇게 넘어온 데이터를 통해 DAO에 접근하여 데이터 커넥션을 진행한다.

```
<%  
    String id = request.getParameter("id"); // memberList id를 받아줌  
    MemberDAO mdao = new MemberDAO();  
    MemberBean mbean = mdao.oneSelectMember(id); // 해당하는 id의 회원 정보를 리턴  
%>  
  
<div align="center">  
  
<h2>회원 정보 상세 </h2>  
  
    <table style="width:400" border="1">  
        <tr height="50">  
            <td align="center" width="150">아이디</td>  
            <td width="250"><%=mbean.getId() %></td>  
        </tr>
```

- 데이터를 처리하는 패턴은 대부분 비슷한 양식을 가짐
- DAO 클래스

```

// 한 사람에 대한 정보를 리턴하는 메소드 작성
public MemberBean oneSelectMember(String id) {
    // 한 사람에 대한 정보만 리턴하기에 bean 클래스 객체 생성
    MemberBean bean = new MemberBean();

    try {
        // 커넥션 연결
        getCon();
        // 쿼리 준비
        String sql = "select * from member where id=?";
        pstmt = con.prepareStatement(sql);
        // ? 의 값을 맵핑
        pstmt.setString(1, id);
        // 쿼리 실행
        rs = pstmt.executeQuery();

        if(rs.next()) { // 레코드가 있다면
            bean.setId(rs.getString(1));
            bean.setPass1(rs.getString(2));
            bean.setEmail(rs.getString(3));
            bean.setTel(rs.getString(4));
            bean.setHobby(rs.getString(5));
            bean.setJob(rs.getString(6));
            bean.setAge(rs.getString(7));
            bean.setInfo(rs.getString(8));
        }
        // 자원 반납
        con.close();

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

    return bean;
}

```

#DataBase