09강 데이터베이스를 이용한 회원 관리 시스템 구축하기

데이터 베이스 커넥션 풀

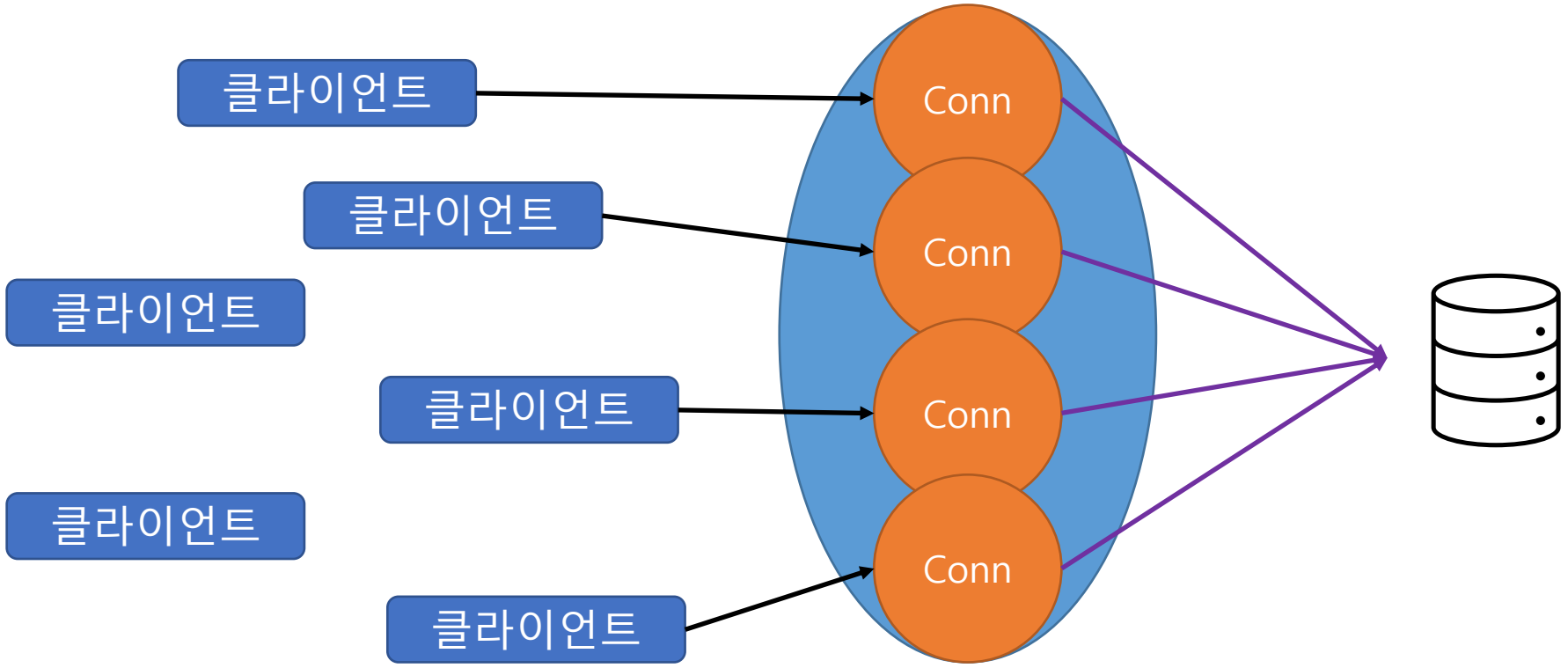
- 데이터 베이스에 어떤 쿼리를 실행하기 위해서 가장 우선해야 할 작업 : 연결된 상태
- 이런 연결된 상태를 '커넥션'이라고 부른다.

- 그러나 웹 페이지에 접속자가 많아지면 커넥션도 동일한 수 만큼 만들어 주어야 하기 때문에 서버에 과부하가 걸리게 되고 심한 경우 서버가 다운되는 현상(서버가 폭파 되었다고도 한다)이 발생한다.

- 이런 문제를 해결하기 위해서 나온 기법이 '커넥션 풀'이다

데이터 베이스 커넥션 풀

- 커넥션 풀(DBCP : Database Connection Pool)은 dbcp 매니저가 미리 연결 수량을 확보하고서 요청이 들어오면 확보된 연결(커넥션)을 제공하고 처리가 완료되면 다시 연결을 회수해서 재활용하는 방식이다.



커넥션 풀

데이터 베이스 커넥션 풀

- 커넥션 풀을 생성하기 위해서는 우선 프로젝트가 서버에 컨텍스트 패스가 등록되어야 한다.
- ex01_dbcp.jsp파일을 생성한 후 실행을 해서 자동으로 등록시켜준다.

```
<Context docBase="web09" path="/web09" reloadable="true" source="org.eclipse.jst.jee.server:web09"/>
```

- 위 컨텍스트 설정하는 곳에 DBCP 코드를 넣는다.

```
<Context docBase="web09" path="/web09" reloadable="true" source="org.eclipse.jst.jee.server:web09">
  <Resource name="jdbc/myoracle" auth="Container"
        type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"
        url="jdbc:oracle:thin:@127.0.0.1:1521:XE"
        username="HONGTEAM" password="1234" maxTotal="20" maxIdle="10"
        maxWaitMillis="-1"/>
 </Context>
```

데이터 베이스 커넥션 풀

- DBCP를 사용하기 위한 코드는 Tomcat 사이트에 알 수 있다

Documentation

- Tomcat 10.0 (alpha)
- Tomcat 9.0
- Tomcat 8.5
- Tomcat 7.0
- Tomcat Connectors
- Tomcat Native
- Wiki
- Migration Guide
- Presentations

User Guide

1) Introduction
2) Setup
3) First webapp
4) Deployer
5) Manager
6) Host Manager
7) Realms and AAA
8) Security Manager
9) JNDI Resources
10) JDBC DataSources
11) Classloading
12) JSPs
13) SSL/TLS
14) SSI
15) CGI
16) Proxy Support
17) MBeans Descriptors

Table of Contents

데이터 베이스 커넥션 풀

- DBCP를 사용하기 위한 코드는 Tomcat 사이트에 알 수 있다.

1. Context configuration

In a similar manner to the mysql config above, you will need to define your Datasource in your Context. Here we define a Datasource calle
thin driver to connect as user scott, password tiger to the sid called mysid. (Note: with the thin driver this sid is not the same as the tnsnar
will be the default schema for the user scott.

Use of the OCI driver should simply involve a changing thin to oci in the URL string.

```
<Resource name="jdbc/myoracle" auth="Container"
              type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"
              url="jdbc:oracle:thin:@127.0.0.1:1521:mysid"
              username="scott" password="tiger" maxTotal="20" maxIdle="10"
              maxWaitMillis="-1"/>
```

- 추가로 DBCP를 사용하기 위한 코드도 예시로 알려준다.

3. Code example

You can use the same example application as above (assuming you create the required D
like

```
Context initContext = new InitialContext();
Context envContext  = (Context)initContext.lookup("java:/comp/env");
DataSource ds = (DataSource)envContext.lookup("jdbc/myoracle");
Connection conn = ds.getConnection();
//etc.
```

데이터 베이스 커넥션 풀

- 예시코드를 통해서 연결을 확인해 보자 – 예제 1

데이터 베이스 커넥션 풀

- 예시코드를 통해서 연결을 확인해 보자 – 예제 1

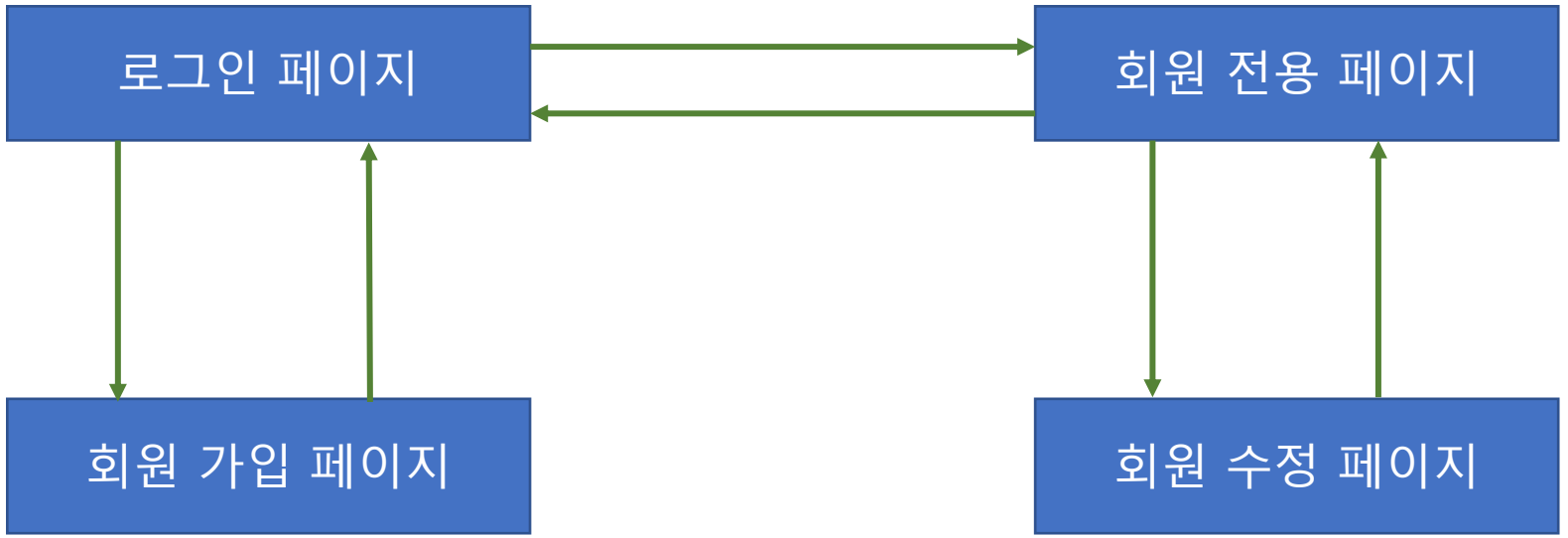
```
Context envContext  = (Context)initContext.lookup("java:/comp/env");
DataSource ds = (DataSource)envContext.lookup("jdbc/myoracle");

<Resource name="jdbc/myoracle" auth="Container"
          type="javax.sql.DataSource" driverClassName="oracle.jdbc.OracleDriver"
          url="jdbc:oracle:thin:@127.0.0.1:1521:XE"
          username="HONGTEAM" password="1234" maxTotal="20" maxIdle="10"
          maxWaitMillis="-1"/>
```

데이터 베이스 커넥션 풀

- DBCP의 요소

속성	설명
initialSize	최초 시점에 커넥션 풀을 채워 넣을 커넥션 개수
maxTotal	동시에 사용가능한 최대 커넥션 개수
maxIdle	Connection Pool에 반납할 때 최대로 유지될 수 있는 커넥션 개수
minIdle	최소한으로 유지할 커넥션 개수
maxWaitMillis	pool이 고갈되었을 경우 최대 대기 시간

데이터 베이스를 연동한 회원 관리 시스템



데이터 베이스를 연동한 회원 관리 시스템

JSP 파일	설명	경로
login.jsp	회원 인증을 위해 아이디와 비밀번호를 입력받는 폼	WebContent/member
join.jsp	회원 가입을 위해 정보를 입력받는 폼	WebContent/member
main.jsp	회원인증후 서비스를 제공하는 폼	WebContent
memberUpdate.jsp	회원 정보를 수정하기 위한 폼	WebContent/member

서블릿 파일	설명	URI 패턴
JoinServlet.java	입력된 회원 정보로 회원 가입 처리	WebContent/member
LoginServlet.java	회원 인증 처리	WebContent/member
MemberUpdateServlet.java	입력된 회원 정보로 회원 정보 수정	WebContent
LogoutServlet.java	로그 아웃 처리	WebContent/member

데이터 베이스를 연동한 회원 관리 시스템

VO클래스	설명
MemberVo.java	회원 정보를 저장하기 위한 클래스

DAO클래스	설명
MemberDAO.java	데이터베이스와 연동해서 작업하는 데이터베이스 처리 클래스

js 파일	설명
member.js	폼 입력 정보가 정확한지 체크하는 자바스크립트

데이터 베이스를 연동한 회원 관리 시스템

- VO 객체(Value Object)
- VO 객체는 데이터베이스에 정보를 전달하거나 전달 받기 위해서 사용하는 객체이다.
- 데이터베이스에 저장된 하나의 레코드 정보를 통째로 전달 하기 위해서는 레코드가 가지는 컬럼 정보와 동일한 구조로 만들게 된다.
- 클래스로부터 정보를 꺼내거나 담기 위해서 setter, getter 메소드를 사용한다.

COLUMN_NAME	DATA_TYPE
NAME	VARCHAR2(10 BYTE)
USERID	VARCHAR2(10 BYTE)
USERPWD	VARCHAR2(10 BYTE)
EMAIL	VARCHAR2(20 BYTE)
PHONE	VARCHAR2(13 BYTE)
ADMIN	NUMBER(1,0)

```
private String name;
private String userid;
private String userpwd;
private String email;
private String phone;
private int admin;
```

- 회원 정보를 하나로 묶기 위한 매커니즘을 자바 빈이라고 했는데 이를 데이터베이스와의 전달 관계가 형성되면 이런 클래스를 VO클래스라고 한다.

- 데이터 전달을 목적으로 사용하는 클래스이므로 DTO라고도 한다.(VO와 DTO의 차이는 직접 찾아 보길 바람)

데이터 베이스를 연동한 회원 관리 시스템

- DAO 객체(Data Access Object)
- DAO 객체는 데이터베이스에 접근하기 위해 사용되는 객체이다.
- 데이터베이스에 데이터를 조회, 추가, 수정, 삭제등의 역할을 하게 된다.

- 이 객체는 쿼리문을 생성해서 전달하는 기능을 주로 담당하게 되는데 정보를 담는 역할 보다 기능이 주된 내용을 차지하므로 매번 객체를 생성하는 것은 비효율적이다.

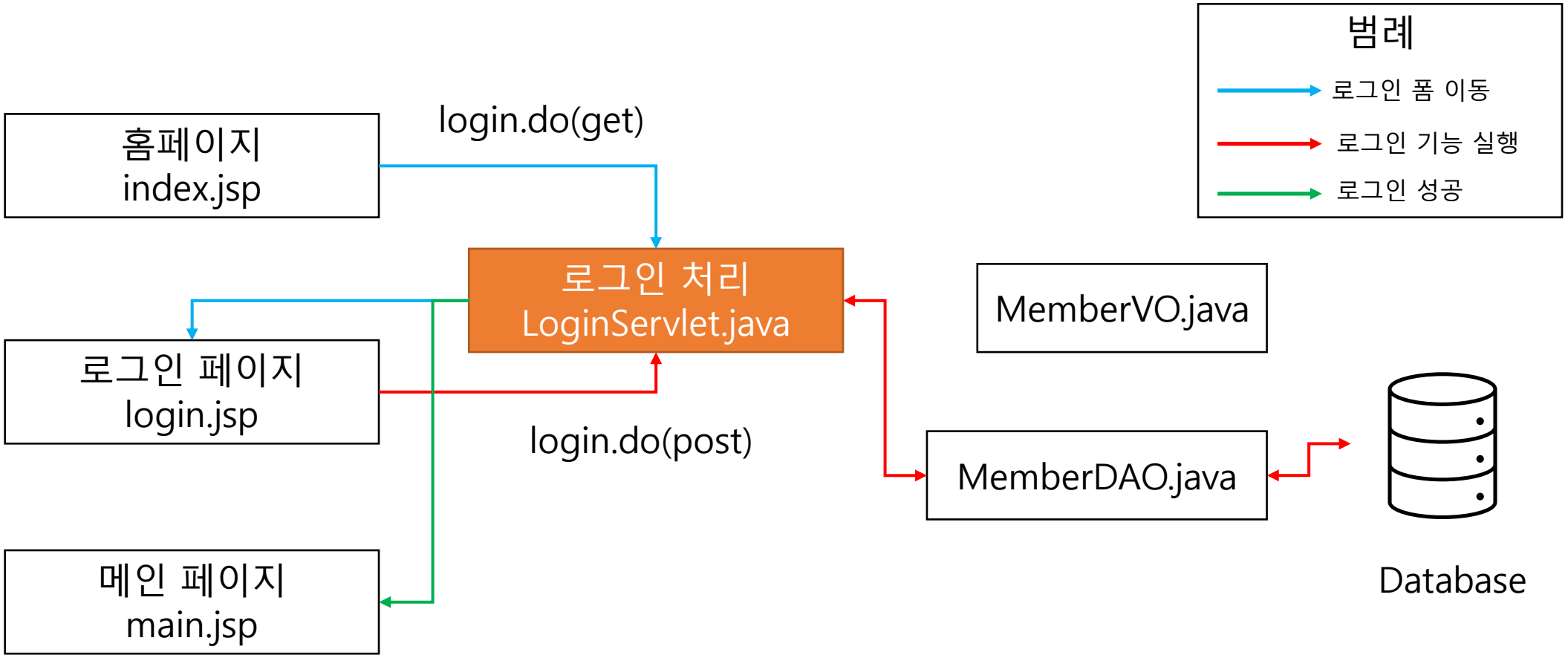
=> 그러므로 싱글톤 패턴으로 클래스를 구성한다.

데이터 베이스를 연동한 회원 관리 시스템

----로그인 기능 구현 ----

- 1단계 DB 테이블을 준비한다. (미리 몇 명의 데이터를 준비해 두자)
- 2단계 Vo 객체 준비한다.
- 3단계 Dao객체를 싱글톤으로 준비한다.
- 4단계 DAO 객체 DBCP 등록 -> 테스트 코드로 데이터베이스 접속을 테스트 한다.
- 5단계 로그인 폼 만들기 -login.jsp , member.js
- 6단계 로그인 폼에 직접 접근하지 못하도록 서블릿을 통한 매핑 만들기
- 7단계 프론트 페이지 만들기
- 8단계 회원 인증 처리 DAO클래스
- 9단계 LoginServlet의 doget메소드와 doPost 메소드 완성
  -get요청시 로그인폼 페이지로 이동시킨다.
  -POST요청시 로그인 인증을 처리한다.
- 10단계 인증된 회원에게만 보여줄 페이지 main.jsp를 만든다.

데이터 베이스를 연동한 회원 관리 시스템

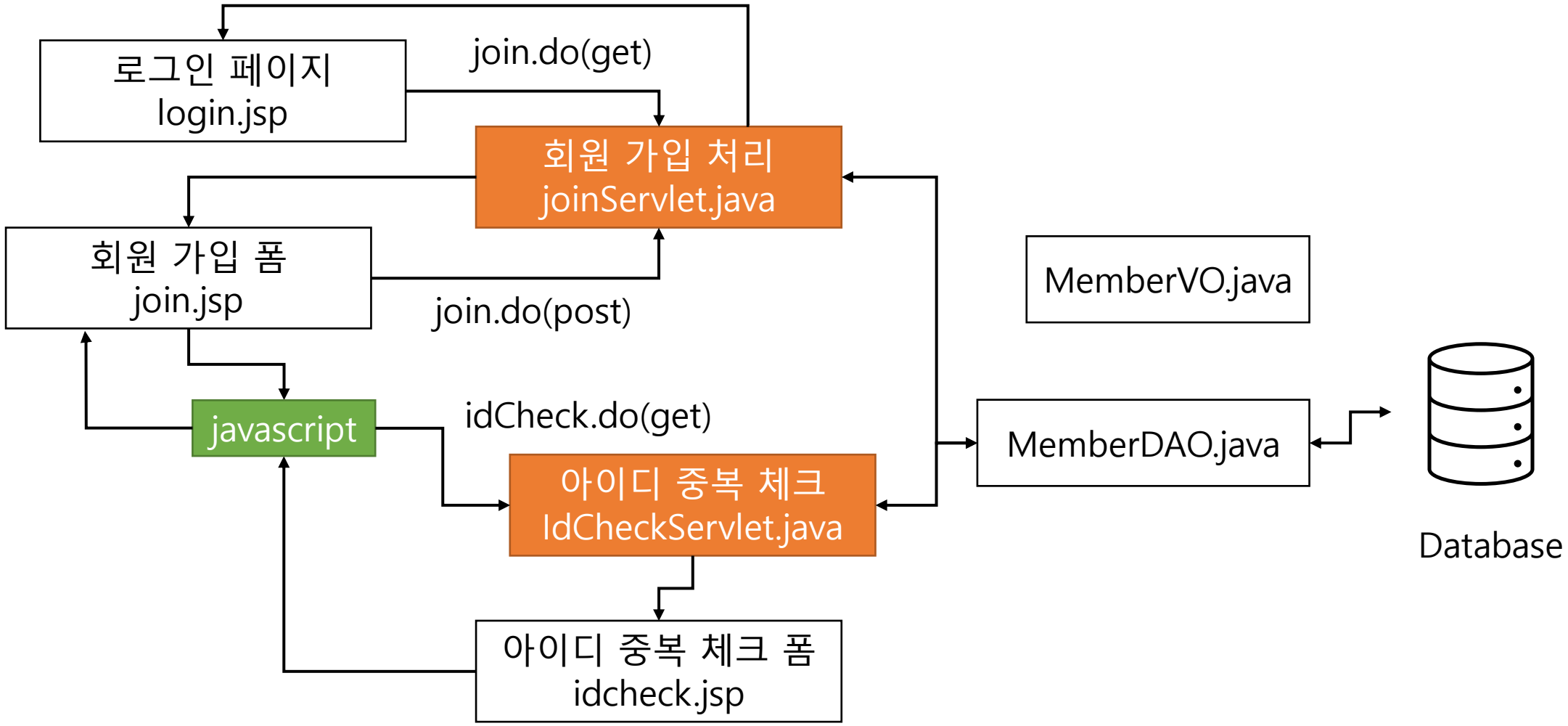


데이터 베이스를 연동한 회원 관리 시스템

--- 회원 가입 기능 구현 ----

- 1단계 로그인 페이지에서 회원 가입 버튼 누르면 회원 가입 폼으로 이동하게 만든다.
  - 서블릿을 통한 매핑 처리를 한다.
- 2단계 아이디 중복 체크를 위한 코드를 작성한다.
  - 만들어진 중복체크 버튼을 자바스크립트 코드로 연결 시킨다.
  - Dao 에 아이디 중복 체크하기 위한 쿼리 메소드를 작성한다.
  - 아이디 중복 체크 페이지로 연결하기 위한 서블릿을 만든다.
  - 아이디 중복 체크 페이지를 만든다.
  - 중복 체크 완료된 후 이전 페이지로 돌아가기 위한 자바스크립트 코드를 추가한다.
- 3단계 회원 가입 폼에서 사용할 유효성 검증 코드를 추가한다.
- 4단계 회원 가입시 사용한 쿼리문을 Dao클래스에 작성한다.
- 5단계 회원 가입폼에서 전달 받은 데이터를 처리하기 위한 코드를 서블릿에 전달하여 처리한다.

데이터 베이스를 연동한 회원 관리 시스템

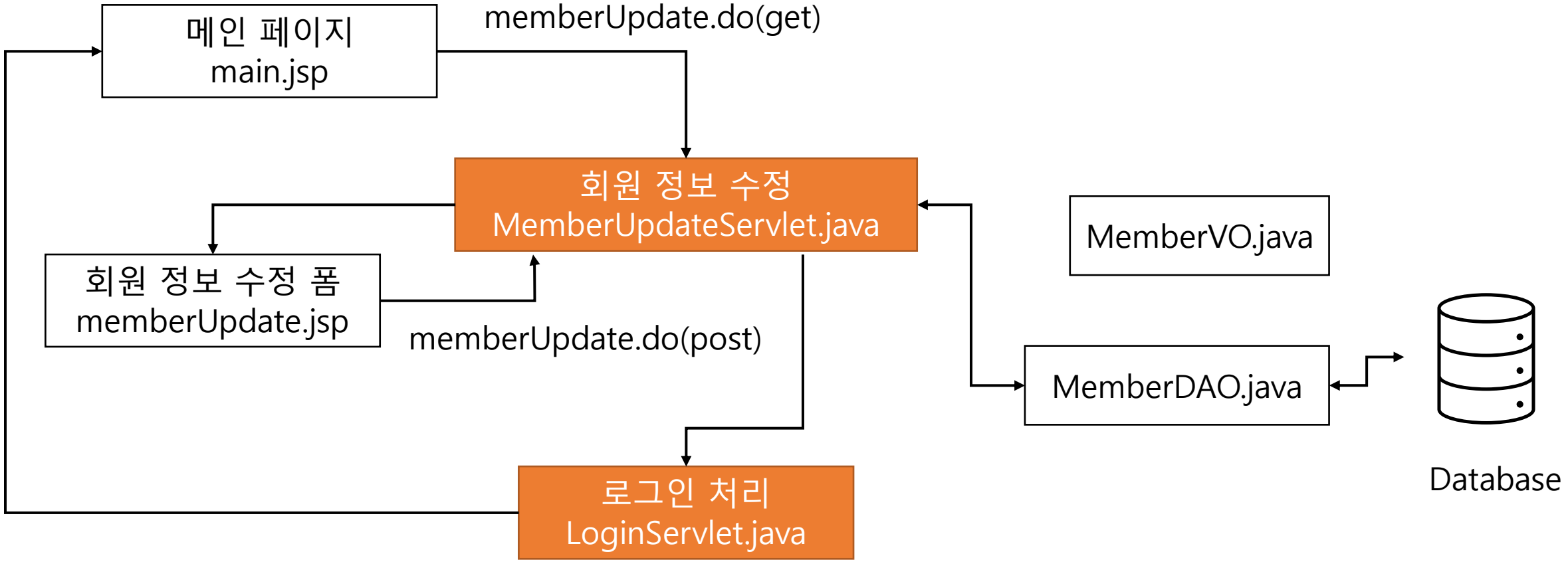


데이터 베이스를 연동한 회원 관리 시스템

--- 회원 정보 수정 기능 구현 ----

- 1단계 메인페이지 회원 정보 수정 버튼을 만든다
- 2단계 회원 수정 폼으로 이동할 서블릿을 만들고 매핑한다.
  - 이미 로그인 된 회원의 정보를 가져오도록 Dao 클래스에서 회원 정보를 호출한다.
- 3단계 회원 수정 폼을 만든다.
  - 알고 있는 회원 정보를 보여준다.
- 4단계 회원 정보를 수정할 쿼리를 Dao 클래스에 작성한다.
- 5단계 회원 수정 폼에서 받은 데이터를 처리하기 위한 코드를 서블릿에 작성한다.
- 6단계 회원 정보를 수정한 회원은 로그인페이지가 아닌 메인 페이지로 이동하도록 수정한다.

데이터 베이스를 연동한 회원 관리 시스템



데이터 베이스를 연동한 회원 관리 시스템

--- 로그 아웃 기능 구현 ----

- 1단계 로그아웃을 위한 서블릿을 만들어서 매핑 처리한다.

데이터 베이스를 연동한 회원 관리 시스템