

Spring Database 연동하기

Spring boot 에서 Mybatis를 통해 MariaDB 연동하기!

MyBatis사용하기!

Mybatis는 java코드와 SQL문을 분리하여 작성한다!
이는 코드의 가독성과 간략화에 큰 효율이 있습니다.

(1) 객체!

```
public class Test {  
  
    private String ID;  
  
    private String Password;  
  
    public String getID() {  
        return ID;  
    }  
  
    public void setID(String id) {  
        this.ID = id;  
    }  
  
    public String getPassword() {  
        return Password;  
    }  
  
    public void setPassword(String password) {  
        this.Password = password;  
    }  
  
}
```

Mybatis는 SQL문의 결과를 원하는 객체로 받아올 수 있습니다.
현재 프로젝트에서는 Test라는 객체를 선언하여 Test객체의 상태변수인 ID,password를 받아옵니다.

(2) Mapper

```

package com.example.demo.db.mapper;

import java.util.List;

import com.example.demo.db.dto.Test;
import org.springframework.context.annotation.Bean;

// Mapper이다. 그저 interface로 정의한다.
// interface로 정의된 mapper는 service와의 의존성이 있다.
public interface TestMapper {

    public List<Test> getAll() throws Exception;

}

```

Mapper는 interface로 정의하며 DB의 SQL문을 실행할 껌데기를 지정합니다!
 예시-> public List<Test> getAll() throws Exception;

(3) Service

```

package com.example.demo.db.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.example.demo.db.dto.Test;
import com.example.demo.db.mapper.TestMapper;

// Service로 정의하고 Autowired를 통해 testMapper와의 의존성을 주입해준다.
@Service
public class TestService {

    @Autowired
    TestMapper testMapper;

    // testMapper로 정의된 객체를 주입받고
    // testMapper에 선언되어있는 getAll메소드를 사용한다.

    // getAll 메소드는 resource/mybatis/mapper/TestDB.xml에서 정의한다!!
    public List<Test> getAll() throws Exception{
        return testMapper.getAll();
    }

}

```

interface로 정의된 Mapper는 Service에서 Autowired로 의존성을 주입받아 사용합니다.
 Mapper에 정의된 메소드를 사용하여 Service에서 원하는 대로 정의합니다.

(4) resource/mybatis/mapper/???.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.demo.db.mapper.TestMapper">
  <select id="getAll" resultType="com.example.demo.db.dto.Test">
    SELECT * FROM userinfo
  </select>
</mapper>
```

이상한 점이 있습니다! **Mapper**는 껍데기만 만들고 내용물을 적어주지 않았다는 것!!

이를 이 경로에서 정의해줍니다.

java코드와 SQL문을 분리했다는 것이 이부분입니다.

<예시>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.example.demo.db.mapper.TestMapper">
  <select id="getAll" resultType="com.example.demo.db.dto.Test">
    SELECT * FROM userinfo
  </select>
</mapper>
```

이는 **TestMapper**라는 **interface**에 정의한 메소드를 네임스페이스로 사용하여 정의하는 것입니다.

선언만 해놓은 **getAll** 메소드를 정의해줍니다!

id->메소드명 **resultType**->결과반환 자료형 을 마음대로 정할 수 있습니다.

테스트용이기 때문에 테이블의 내용을 전부 반환해봤습니다.

(5) DatabaseConfig.java

Mybatis를 사용하기위한 부분입니다.

DataSource를 사용하여 설정을 해주는부분인데 자세한 것은 구글링 참고!

추후 공부해서 업데이트하겠습니다.

(6) 실행결과!

```

package com.example.demo.db.Controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;

import com.example.demo.db.dto.Test;
import com.example.demo.db.service.TestService;

// DB와 연결되는 부분은 db파일로 구분했음 db와 연동되는 jsp의 controller
@Controller
public class TestController {


    @Autowired
    TestService testService;

    // query라는 값과 매칭된다.
    // 그저 SQL문 실행결과를 뿌려주지만
    // model객체를 사용하면 원하는 형태로 html문과 섞어 출력할 수 있을 것 같다.
    @RequestMapping("/query")
    public @ResponseBody List<Test> query() throws Exception{
        return testService.getAll();
    }
}

```

테스트를 위하여 DB에 접속하는 부분은 따로 Controller를 정의했습니다.
./query 라는 url에 접속하면 testService.getAll() 결과값을 뿌려줍니다.

ID	Password
jhs	2222
jjs	3333
jks	1111



localhost:8080/query

```
[{"id":"jhs","password":"2222"}, {"id":"jjs","password":"3333"}, {"id":"jks","password":"1111"}]
```

DB의 userinfo 테이블과 spring boot 실행결과창 확인 시 일치하는 것을 확인!!

DB와의 연동 완료!!!!!!!!!!!!!!

Made by 장경석