[**第三篇：SpringBoot用JdbcTemplates访问Mysql**](http://blog.csdn.net/forezp/article/details/70477821)

参考：<http://blog.csdn.net/forezp/article/details/70477821>

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/70477821>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp)

本文介绍springboot通过jdbc访问关系型mysql,通过spring的JdbcTemplate去访问。

准备工作

* jdk 1.8
* maven 3.0
* idea
* mysql

初始化mysql:

-- create table `account`

DROP TABLE `account` IF EXISTS

CREATE TABLE `account` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(20) NOT NULL,

`money` double DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=utf8;

INSERT INTO `account` VALUES ('1', 'aaa', '1000');

INSERT INTO `account` VALUES ('2', 'bbb', '1000');

INSERT INTO `account` VALUES ('3', 'ccc', '1000');

创建工程

引入依赖：

在pom文件引入spring-boot-starter-jdbc的依赖：

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-jdbc</artifactId>

</dependency>

引入mysql连接类和连接池：

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>com.alibaba</groupId>

<artifactId>druid</artifactId>

<version>1.0.29</version>

</dependency>

开启web:

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

配置相关文件

在application.properties文件配置mysql的驱动类，数据库地址，数据库账号、密码信息。

spring.datasource.driver-class-name=com.mysql.jdbc.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/test

spring.datasource.username=root

spring.datasource.password=123456

通过引入这些依赖和配置一些基本信息，springboot就可以访问数据库类。

具体编码

实体类

public class Account {

private int id ;

private String name ;

private double money;

....省略了getter. setter

}

dao层

public interface IAccountDAO {

int add(Account account);

int update(Account account);

int delete(int id);

Account findAccountById(int id);

List<Account> findAccountList();

}

具体的实现类：

package com.forezp.dao.impl;

import com.forezp.dao.IAccountDAO;

import com.forezp.entity.Account;

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

import org.springframework.jdbc.core.BeanPropertyRowMapper;

import org.springframework.jdbc.core.JdbcTemplate;

import org.springframework.stereotype.Repository;

import java.util.List;

/\*\*

\* Created by fangzhipeng on 2017/4/20.

\*/

@Repository

public class AccountDaoImpl implements IAccountDAO {

@Autowired

private JdbcTemplate jdbcTemplate;

@Override

public int add(Account account) {

return jdbcTemplate.update("insert into account(name, money) values(?, ?)",

account.getName(),account.getMoney());

}

@Override

public int update(Account account) {

return jdbcTemplate.update("UPDATE account SET NAME=? ,money=? WHERE id=?",

account.getName(),account.getMoney(),account.getId());

}

@Override

public int delete(int id) {

return jdbcTemplate.update("DELETE from TABLE account where id=?",id);

}

@Override

public Account findAccountById(int id) {//此处跟.net的dapper差不多

List<Account> list = jdbcTemplate.query("select \* from account where id = ?", new Object[]{id}, new BeanPropertyRowMapper(Account.class));

if(list!=null && list.size()>0){

Account account = list.get(0);

return account;

}else{

return null;

}

}

@Override

public List<Account> findAccountList() {

List<Account> list = jdbcTemplate.query("select \* from account", new Object[]{}, new BeanPropertyRowMapper(Account.class));

if(list!=null && list.size()>0){

return list;

}else{

return null;

}

}

}

service层

public interface IAccountService {

int add(Account account);

int update(Account account);

int delete(int id);

Account findAccountById(int id);

List<Account> findAccountList();

}

具体实现类：

@Service

public class AccountService implements IAccountService {

@Autowired

IAccountDAO accountDAO;

@Override

public int add(Account account) {

return accountDAO.add(account);

}

@Override

public int update(Account account) {

return accountDAO.update(account);

}

@Override

public int delete(int id) {

return accountDAO.delete(id);

}

@Override

public Account findAccountById(int id) {

return accountDAO.findAccountById(id);

}

@Override

public List<Account> findAccountList() {

return accountDAO.findAccountList();

}

}

构建一组restful api来展示

package com.forezp.web;

import com.forezp.entity.Account;

import com.forezp.service.IAccountService;

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

import org.springframework.web.bind.annotation.\*;

import java.util.List;

/\*\*

\* Created by fangzhipeng on 2017/4/20.

\*/

@RestController

@RequestMapping("/account")

public class AccountController {

@Autowired

IAccountService accountService;

@RequestMapping(value = "/list",method = RequestMethod.GET)

public List<Account> getAccounts(){

return accountService.findAccountList();

}

@RequestMapping(value = "/{id}",method = RequestMethod.GET)

public Account getAccountById(@PathVariable("id") int id){

return accountService.findAccountById(id);

}

@RequestMapping(value = "/{id}",method = RequestMethod.PUT)

public String updateAccount(@PathVariable("id")int id , @RequestParam(value = "name",required = true)String name,

@RequestParam(value = "money" ,required = true)double money){

Account account=new Account();

account.setMoney(money);

account.setName(name);

account.setId(id);

int t=accountService.update(account);

if(t==1){

return account.toString();

}else {

return "fail";

}

}

@RequestMapping(value = "",method = RequestMethod.POST)

public String postAccount( @RequestParam(value = "name")String name,

@RequestParam(value = "money" )double money){

Account account=new Account();

account.setMoney(money);

account.setName(name);

int t= accountService.add(account);

if(t==1){

return account.toString();

}else {

return "fail";

}

}

}

可以通过postman来测试，具体的我已经全部测试通过，没有任何问题。注意restful构建api的风格。

源码下载：<https://github.com/forezp/SpringBootLearning>

参考资料

[relational-data-access](https://spring.io/guides/gs/relational-data-access/)

优秀文章推荐：

* 更多springboot 教程：[springBoot非官方教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70341818)
* 更多springcoud 教程：[史上最简单的 SpringCloud 教程 | 文章汇总](http://blog.csdn.net/forezp/article/details/70148833)