[**第六篇：springboot整合mybatis**](http://blog.csdn.net/forezp/article/details/70768477)

参考：<http://blog.csdn.net/forezp/article/details/70768477>

转载请标明出处：   
<http://blog.csdn.net/forezp/article/details/70768477>   
本文出自[方志朋的博客](http://blog.csdn.net/forezp)

本文主要讲解如何在springboot下整合mybatis，并访问数据库。由于mybatis这个框架太过于流行，所以我就不讲解了。

## 引入依赖

在pom文件引入mybatis-spring-boot-starter的依赖：

<dependency>

<groupId>org.mybatis.spring.boot</groupId>

<artifactId>mybatis-spring-boot-starter<artifactId>

<version>1.3.0</version>

</dependency>

引入数据库连接依赖：

<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>

## 引入数据源

application.properties配置文件中引入数据源：

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

spring.datasource.username=root

spring.datasource.password=123456

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

这样，springboot就可以访问数据了。

## 创建数据库表

建表语句：

-- 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');

## 具体实现

这篇通过注解的形式实现。

### 创建实体：

“`   
public class Account {   
private int id ;   
private String name ;   
private double money;

setter…   
getter…   
}

“`

### dao层

@Mapper

public interface AccountMapper {

@Insert("insert into account(name, money) values(#{name}, #{money})")

int add(@Param("name") String name, @Param("money") double money);

@Update("update account set name = #{name}, money = #{money} where id = #{id}")

int update(@Param("name") String name, @Param("money") double money, @Param("id") int id);

@Delete("delete from account where id = #{id}")

int delete(int id);

@Select("select id, name as name, money as money from account where id = #{id}")

Account findAccount(@Param("id") int id);

@Select("select id, name as name, money as money from account")

List<Account> findAccountList();

}

## service层

@Service

public class AccountService {

@Autowired

private AccountMapper accountMapper;

public int add(String name, double money) {

return accountMapper.add(name, money);

}

public int update(String name, double money, int id) {

return accountMapper.update(name, money, id);

}

public int delete(int id) {

return accountMapper.delete(id);

}

public Account findAccount(int id) {

return accountMapper.findAccount(id);

}

public List<Account> findAccountList() {

return accountMapper.findAccountList();

}

}

## controller层，构建restful API

package com.forezp.web;

import com.forezp.entity.Account;

import com.forezp.service.AccountService;

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

AccountService 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.findAccount(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) {

int t= accountService.update(name,money,id);

if(t==1) {

return "success";

}else {

return "fail";

}

}

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

public String delete(@PathVariable(value = "id")int id) {

int t= accountService.delete(id);

if(t==1) {

return "success";

}else {

return "fail";

}

}

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

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

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

int t= accountService.add(name,money);

if(t==1) {

return "success";

}else {

return "fail";

}

}

}

通过postman测试通过。

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

## 参考资料

[mybatis](http://www.mybatis.org/mybatis-3/zh/configuration.html#properties)

[MyBatis整合](http://blog.720ui.com/2016/springboot_02_data_mybatis/)

### 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)