

# 实训day2


---

功能实现

9:41



## 用户注册


 姓名

请输入你的姓名


 组织

请选择你的公司



 手机号

请输入你的手机号

 短信验证码

请输入验证码

发送验证码

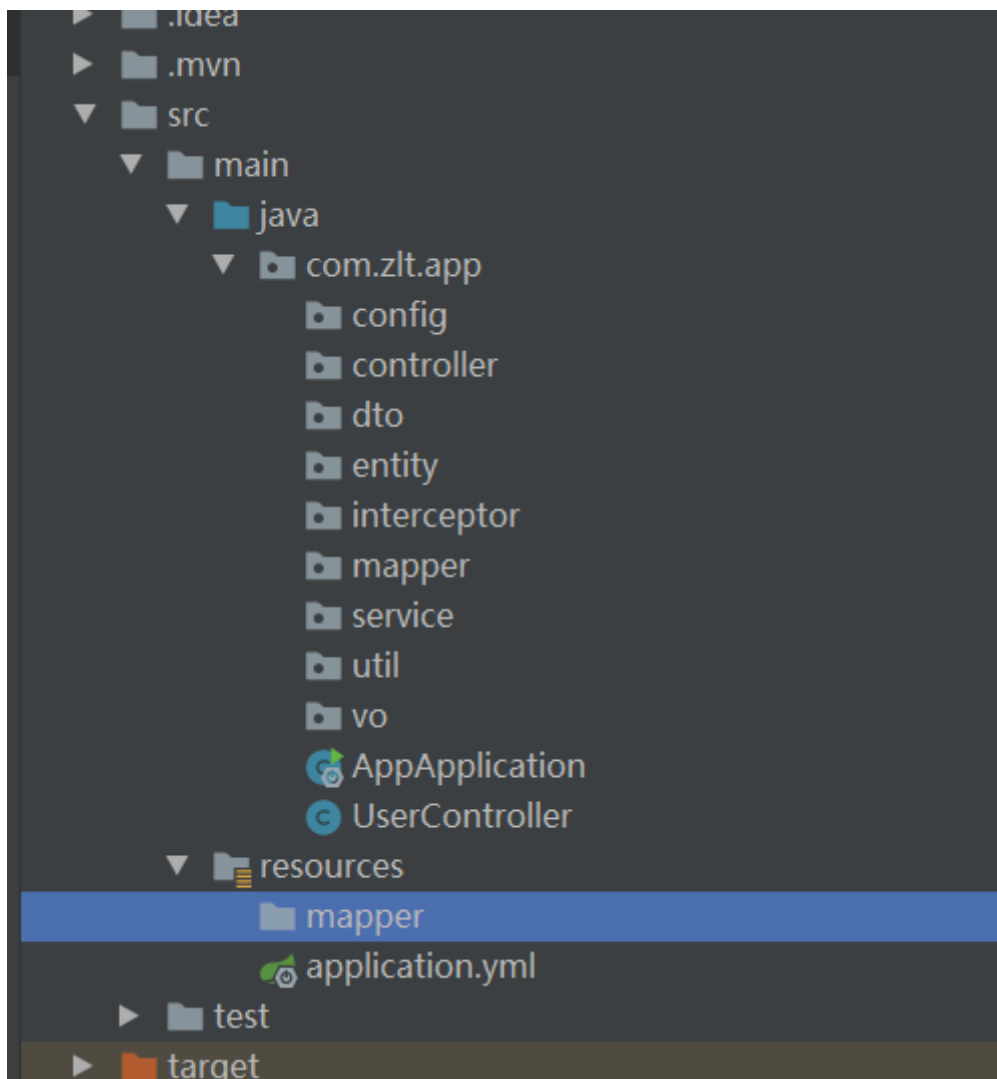
 密码

请输入你的密码

注册

账号登录

新建包如下



## 获取组织列表

### DepartmentController

```
package com.zlt.app.controller;

import com.zlt.app.entity.Department;
import com.zlt.app.service.DepartmentService;
import com.zlt.app.service.impl.DepartmentServiceImpl;
import com.zlt.app.vo.Result;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController//标记当前为控制层
@RequestMapping("department")
public class DepartmentController {
    @Autowired
    private DepartmentService departmentService;
```

```

/**
 * 查询组织列表
 */
@GetMapping("selectDepartment")
public Result<List<Department>>selectDepartment(){
    List<Department> departments = departmentService.selectDepartment();
    return Result.success("查询成功",departments);
}
}

```

## DepartmentService

```

package com.zlt.app.service;

import com.zlt.app.entity.Department;

import java.util.List;

public interface DepartmentService {
    List<Department> selectDepartment();
}

```

## DepartmentServiceImpl

```

package com.zlt.app.service.impl;

import com.zlt.app.entity.Department;
import com.zlt.app.mapper.DepartmentMapper;
import com.zlt.app.service.DepartmentService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service //标记当前为业务层
public class DepartmentServiceImpl implements DepartmentService {
    @Autowired
    private DepartmentMapper departmentMapper;
    @Override
    public List<Department> selectDepartment() {
        List<Department> departments = departmentMapper.selectDepart();
        if (departments == null || departments.size() ==0 ){

            throw new RuntimeException("当前没有组织数据");
        }
        return departments;
    }
}

```

## DepartmentMapper

```
package com.zlt.app.mapper;

import com.zlt.app.entity.Department;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Select;

import java.util.List;

@Mapper//标记当前为持久层
public interface DepartmentMapper {

    List<Department> selectDepart();
}
```

## DepartMentMapper.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.zlt.app.mapper.DepartmentMapper" >
<select id="selectDepart" resultType="com.zlt.app.entity.Department">
select did,department_name,department_desc from department_info
</select>

</mapper>
```

## 实现验证码

### UserController

```
package com.zlt.app.controller;

import com.zlt.app.util.StringUtil;
import com.zlt.app.vo.Result;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.redis.core.RedisTemplate;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.concurrent.TimeUnit;

@RestController//标记当前是控制层
@RequestMapping("user")
public class UserController {

    @GetMapping("test") //标记下面的方法是get请求
    public String test(){
        return "hello world";
    }
}
```

```

}
@Autowired
private RedisTemplate redisTemplate;

/**
 * 发送验证码
 * @param phone
 * @return
 */

@GetMapping("send")
public ResponseEntity<Result>send(String phone){
    //生成一个四位数的随机字符串
    String code = StringUtil.getRandomNumber(4);
    System.out.println(code);
    //生成一个redis里面的key
    String key=StringUtil.uuid()+phone;
    System.out.println(key);
    //将生成的验证码放入redis
    redisTemplate.opsForValue().set(key,code,1, TimeUnit.MINUTES);

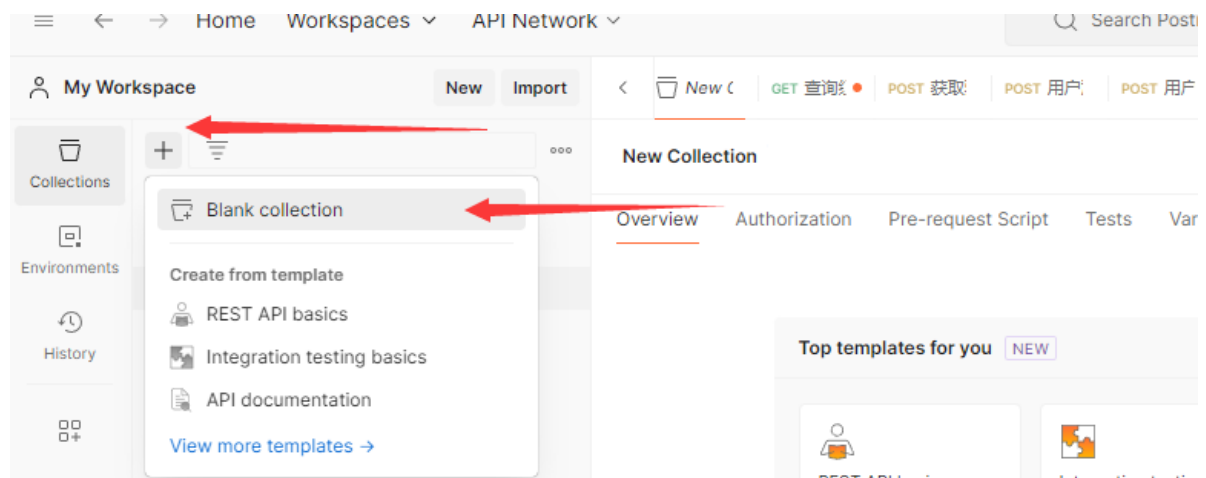
    return
    ResponseEntity.status(200).header("SMS_SEND",key).body(Result.success("发送成功"));
}

}

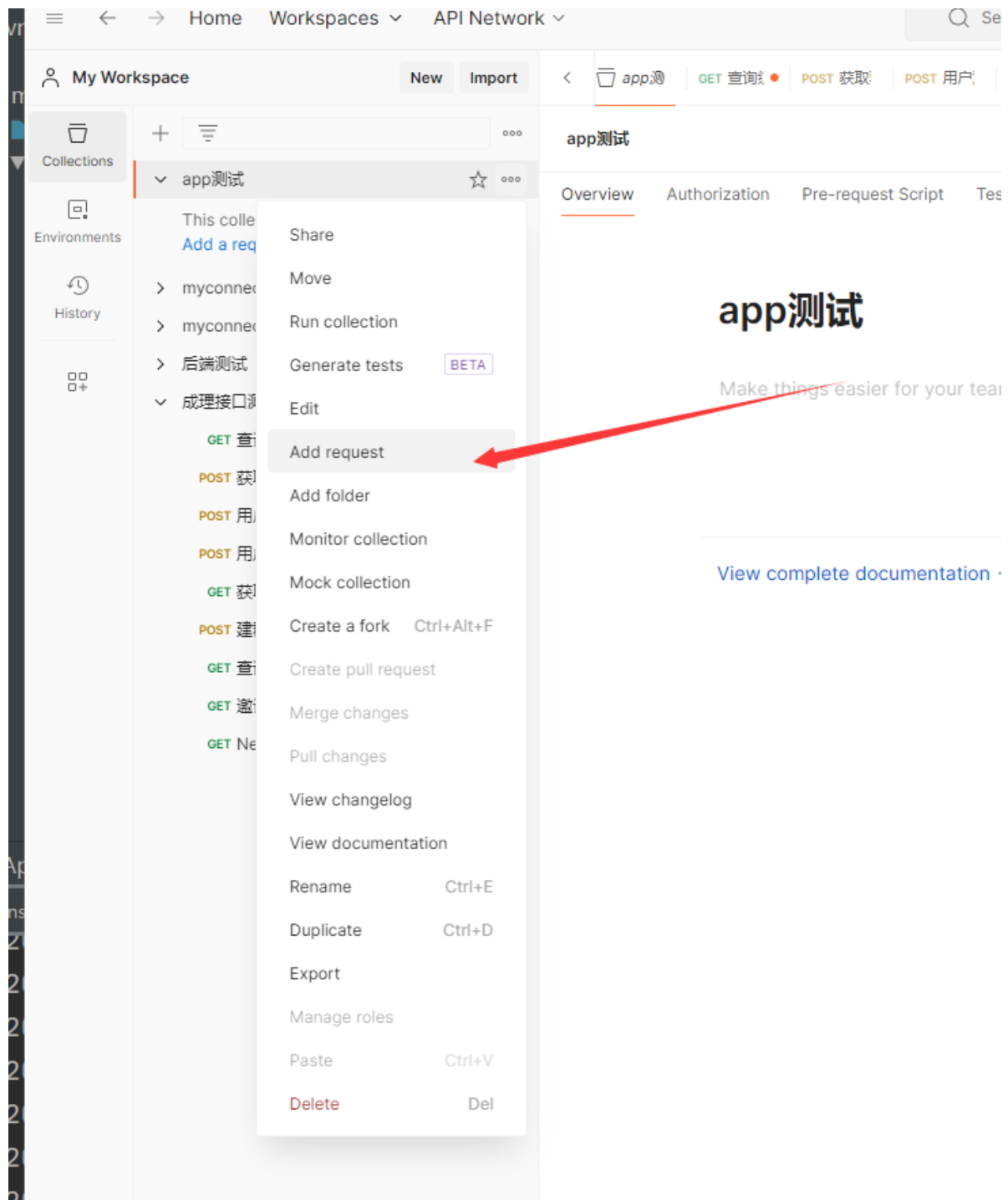
```

## post测试接口

### 1创建测试接口集合



### 2添加测试请求



3发送请求

PI Network

Search Postman

Invite Upgrade

Import < 获取 POST 用户 POST 用户 GET 获取 POST 建群 GET 查询 POST 邀请 POST Nev GET 发送

app测试 / 发送短信验证码

localhost:8080/user/send?phone=17783522966

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

<input checked="" type="checkbox"/> Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> phone	17783522966			
Key	Value	Description		

Response

Click Send to get a response

参数

请求路径

最后点击

结果如图

Body Cookies Headers (6) Test Results

Status: 200 OK Time: 1667 ms Size: 268 B Save as

Pretty Raw Preview Visualize JSON

```
1 {
2   "code": 200,
3   "message": "发送成功",
4   "data": null
5 }
```