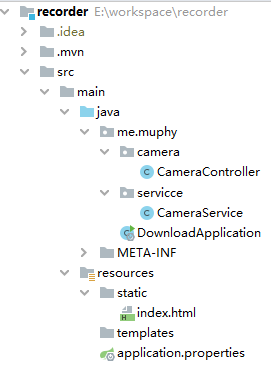
# Java调用计算机摄像头照相（Rest API的页面操作）

使用开源组件webcam-capture：<https://github.com/sarxos/webcam-capture>

项目源码GitHub：<https://github.com/muphy1112/RuphyRecorder>

本例子使用基于Java rest API的页面操作，方便远程拍照

## 新建Spring Boot项目



## pop.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>2.2.5.RELEASE</version>

<relativePath/> <!-- lookup parent from repository -->

</parent>

<groupId>me.muphy</groupId>

<artifactId>recorder</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>recorder</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.github.sarxos</groupId>

<artifactId>webcam-capture</artifactId>

<version>0.3.12</version>

</dependency>

<dependency>

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

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

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

## application.properties

server.port=8080

#保存根路径

record.path=E:/workspace/share/

## index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>莫非照相机</title>

</head>

<body>

<div>

<div><span><a href="/tp">拍照</a></span></div>

</div>

</body>

</html>

## CameraController.java

package me.muphy.camera;

import me.muphy.servicce.CameraService;

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

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class CameraController {

@Autowired

private CameraService cameraService;

@GetMapping("/tp")

public String takePictures(String[] args) {

String msg = cameraService.takePictures();

return "<div><span>" + msg + "</span></div><div>" +

"<span><a href=\"/ll?d=/picture\" >点击查看所有照片</a></span>" +

"<span style=\"margin-left: 20px;\"><a href=\"/\" >返回首页</a></span></div>";

}

}

## CameraService.java

 package me.muphy.servicce;

import com.github.sarxos.webcam.Webcam;

import com.github.sarxos.webcam.WebcamResolution;

import com.github.sarxos.webcam.WebcamUtils;

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

import org.springframework.stereotype.Service;

import java.io.File;

import java.text.SimpleDateFormat;

import java.util.Date;

@Service

public class CameraService {

@Value("${download.path:E:/workspace/download/}")

private String downloadPath;

public String takePictures() {

Webcam webcam = Webcam.getDefault();

if (webcam == null) {

return "没有找到摄像设备！";

}

String filePath = downloadPath + "/picture/" + new SimpleDateFormat("yyyy-MM-dd").format(new Date());

File path = new File(filePath);

if (!path.exists()) {//如果文件不存在，则创建该目录

path.mkdirs();

}

String time = new SimpleDateFormat("yyyMMdd\_HHmmss").format(new Date());

File file = new File(filePath + "/" + time + ".jpg");

webcam.setViewSize(WebcamResolution.VGA.getSize());

WebcamUtils.capture(webcam, file);

return "拍照成功！";

}

}

## CameraApplication.java

package me.muphy;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class CameraApplication {

public static void main(String[] args) {

SpringApplication.run(CameraApplication.class, args);

}

}

## 启动项目

