Assignment 1 README中文版

目录

[一、运行和测试： 3](#_Toc394864423)

[1.运行应用： 3](#_Toc394864424)

[2.运行测试 4](#_Toc394864425)

[3.运行评分 4](#_Toc394864426)

[二、作业目标 5](#_Toc394864427)

[1. GET /video（获取video列表） 5](#_Toc394864428)

[2.POST /video（上传video原数据） 5](#_Toc394864429)

[3.POST /video/{id}/data （上传video二进制文件） 6](#_Toc394864430)

[4. GET /video/{id}/data（返回video数据） 6](#_Toc394864431)

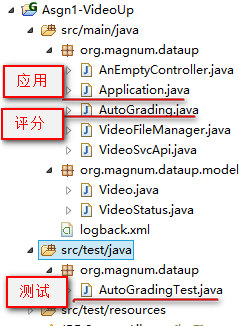
[三、提供的代码 7](#_Toc394864432)

[1. org.magnum.dataup.Video（创建Video类） 7](#_Toc394864433)

[2. org.magnum.dataup.VideoFileManager（存储video二进制文件） 8](#_Toc394864434)

[四、Hints(提示) 9](#_Toc394864435)

## 一、运行和测试：



### 1.运行应用：

运行：包org.magnum.dataup package -> 类Application.java ，右击：As->Java Application

停止：Debug Perspective的Debug View中, 右击运行的Application：点击Terminate

### 2.运行测试

运行：包org.magnum.dataup package -> 类AutoGradingTest.java ，右击：As->JUnit Test

### 3.运行评分

运行：包org.magnum.dataup package -> 类AutoGrading.java ，右击：As->Java Application

## 二、作业目标

### 1. GET /video（获取video列表）

（1）返回video list的JSON。

（2）可以被客户端反序列化。

（3）返回类型是application/json，使用@ResponseBody默认返回此类型

### 2.POST /video（上传video原数据）

(1)video原数据，是application/json request body。

(2）JSON通过jackson被反序列化后，可以生成一个Video类的实例。

（3）服务器将为Video实例生成一个id，通过调用setId(...)方法。

（4）id必须大于0，服务器返回的video JSON中必须包含该id

（5）服务器还要生成一个url, 表示video上传后的地址， 要完全路径，如：http://localhost:8080/video/1/data，如何生成请参见Hint部分。

### 3.POST /video/{id}/data （上传video二进制文件）

(1) 客户端multipart request中包含video二进制文件

(2）客户端先要POST /video，然后得到一个有id的video类实例，再实际POST数据

(3）整个过程结束了，返回一个VideoStatus类，它的state=VideoState.READY，如果一切顺利。

(4) spring的PUT不支持mutlipart data, 所以使用的是POST

### 4. GET /video/{id}/data（返回video数据）

（1）如果该id没有，那么返回 404

## 三、提供的代码

### 1. org.magnum.dataup.Video（创建Video类）

(1)创建video

Video video = Video.create()

.withContentType("video/mpeg")

.withDuration(123)

.withSubject("Mobile Cloud")

.withTitle("Programming Cloud Services for ...")

.build();

（2）获取application/jason中的video

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

public @ResponseBody Video addVideo(@RequestBody Video v)

{

// Do something with the Video

// ...

return v;

}

### 2. org.magnum.dataup.VideoFileManager（存储video二进制文件）

(1)初始化

// Initialize this member variable somewhere with

private VideoFileManager videoDataMgr;

videoDataMgr = VideoFileManager.get();

（2）使用

// You would need some Controller method to call this...

public void saveSomeVideo(Video v, MultipartFile videoData) throws IOException {

videoDataMgr.saveVideoData(video, videoData.getInputStream());

}

public void serveSomeVideo(Video v, HttpServletResponse response) throws IOException {

// Of course, you would need to send some headers, etc. to the

// client too!

// ...

videoDataMgr.copyVideoData(v, response.getOutputStream());

}

## 四、Hints(提示)

1.example还是有用的，可以看看

2.至少有一个类使用了@Controller

3.至少要使用4中不同的RequestMapping

4.任何一个Controller方法都可以，通过一个HttpServletRequest或HttpServletReponse参数，获取底层的HTTP信息。Spring会自动帮你完成参数类型的转换。

@RequestMapping("/some/path/{id}")

public MyObject doSomething( @PathVariable("id") String id, @RequestParam("something") String data, HttpServletResponse response) {

// Maybe you want to set the status code with the response

// or write some binary data to an OutputStream obtained from

// the HttpServletResponse object

....

}

5.id必须使用long类型，int会返回400。其他出现400的情况也可能是参数导致的问题

6.有一个@RequestMapping会使用HttpServletReposnse类作为参数，而且会写入二进制数据返回给客户端。

7.生成video的url地址的方法如下

private String getDataUrl(long videoId){

String url = getUrlBaseForLocalServer() + "/video/" + videoId + "/data";

return url;

}

private String getUrlBaseForLocalServer() {

HttpServletRequest request =

((ServletRequestAttributes) RequestContextHolder.getRequestAttributes()).getRequest();

String base =

"http://"+request.getServerName()

+ ((request.getServerPort() != 80) ? ":"+request.getServerPort() : "");

return base;

}

8.生成id的方法如下

private static final AtomicLong currentId = new AtomicLong(0L);

private Map<Long,Video> videos = new HashMap<Long, Video>();

public Video save(Video entity) {

checkAndSetId(entity);

videos.put(entity.getId(), entity);

return entity;

}

private void checkAndSetId(Video entity) {

if(entity.getId() == 0){

entity.setId(currentId.incrementAndGet());

}

}