Assignment 2 READ ME 中文版

[一、总览 3](#_Toc397028365)

[1. 本作业在原程序基础上增加： 3](#_Toc397028366)

[2.用户流程如下： 3](#_Toc397028367)

[3.运行状况： 3](#_Toc397028368)

[二、警告 3](#_Toc397028369)

[三、运行Application 4](#_Toc397028370)

[1.运行 4](#_Toc397028371)

[四、指导 5](#_Toc397028372)

[1. POST /oauth/token 5](#_Toc397028373)

[2. GET /video（同Sign1） 6](#_Toc397028374)

[3. POST /video（大部同sign1） 6](#_Toc397028375)

[4. GET /video/{id} 7](#_Toc397028376)

[5. POST /video/{id}/like 7](#_Toc397028377)

[6. POST /video/{id}/unlike 7](#_Toc397028378)

[7. GET /video/{id}/likedby 8](#_Toc397028379)

[8. GET /video/search/findByName?title={title} 8](#_Toc397028380)

[9. GET /video/search/findByDurationLessThan?duration={duration} 8](#_Toc397028381)

[10. 提示点： 9](#_Toc397028382)

[五、测试作业 9](#_Toc397028383)

[六、提交作业 9](#_Toc397028384)

[七、提供给你的代码 9](#_Toc397028385)

[1. org.magnum.mobilecloud.video.repository.Video 9](#_Toc397028386)

[2. example中的OAuth 2.0 Configuration 10](#_Toc397028387)

[3. SecuredRestBuilder 10](#_Toc397028388)

[八、指点 10](#_Toc397028389)

## 一、总览

### 1. 本作业在原程序基础上增加：

(1）OAuth 2.0验证

（2）点击like

### 2.用户流程如下：

(1）通过OAuth 2.0验证

(2）允许点击like或unlike

(3）允许搜索videos

(4）数据都存储在JPA中

### 3.运行状况：

(1）程序运行在https上，8443端口

（2）其他同作业1

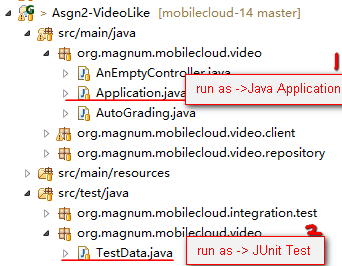
## 二、警告

本例中的KEYSTORE是不安全的

## 三、运行Application

### 1.运行

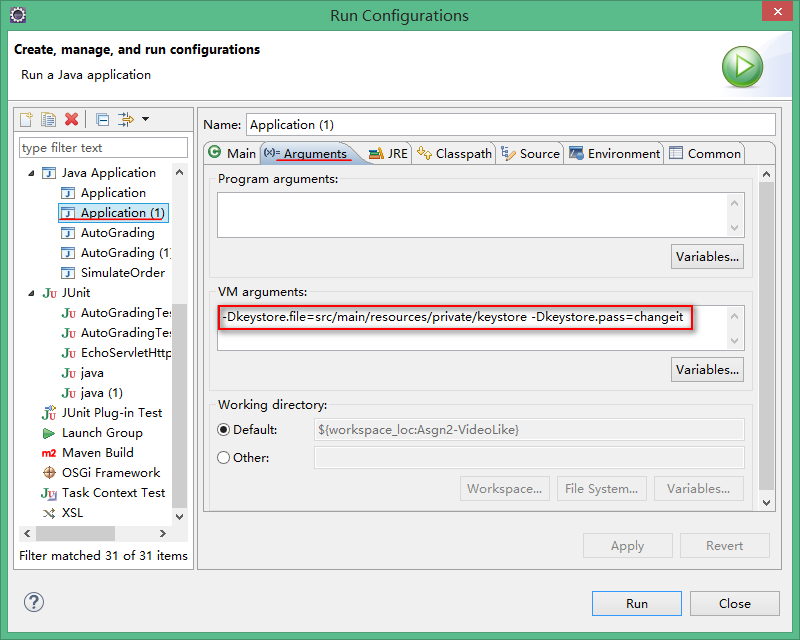
(1）右击项目中的Application类，Run As->Java Application，程序运行。开始下一步骤前，先停止该类运行。



(2）菜单栏中Run->Run Configurations

(3）设置Application中的Argument如下，添加：

-Dkeystore.file=src/main/resources/private/keystore -Dkeystore.pass=changeit



（4）记住这个kestore是不安全的，要安全的话去这里获取：

http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html

## 四、指导

### 1. POST /oauth/token

（1）通过OAuth2.0验证

（2）客户端可以发送一个请求，包含：username, password, client ID, client secret,如同OAuth课程所叙述。

（3）client ID是"mobile"，密码是空。

（4）要有两个用户，username为"user0"和"admin",密码都是"pass"

（5.0）不要从头做起，我们建议使用example\_9中的类：

https://github.com/juleswhite/mobilecloud-14/tree/master/examples/9-VideoServiceWithOauth2/src/main/java/org/magnum/mobilecloud/video/auth

（5.1）需要@Import the OAuth2SecurityConfiguration

（5.2）要从OAuth2SecurityConfiguration类或Application类中，移除其中一个的containerCustomizer()方法

（5.3）设置users，在OAuth2Config构造函数中、或ResourceServer.configure(...)方法中，需要自己判定哪些要被改变，请对比测试代码来决定

### 2. GET /video（同Sign1）

（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

### 3. POST /video（大部同sign1）

(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。

*(2.1.new）返回的JSON要反应update的内容*

*（3.new）服务器将Video实例存储在Spring Data JPA中。如果设置正确，将会自动生成一个id。*

*（4.new）初始化的Video不该有任何一个like。*

*（5.new)Video类中使用一个或更多的annotations，为了让它能够被JPA存储*

### 4. GET /video/{id}

（1）返回此id的video，或者404

### 5. POST /video/{id}/like

（1.0）允许user点击like时，成功返回200，

(1.1）如果video不存在，返回404

(1.2）如果该user已经like过了，返回400

（2.0）服务要记录哪个user已经like过某个video，且防止其再次like。

（2.1）POJO Video类已经提供给你用了，为了持久化，你需要给他添加annotate

### 6. POST /video/{id}/unlike

（1.0）允许用户对like过的video，进行unlike

（1.1）成功返回200

(1.2）video不存在，返回404

(1.3）之前没有like过，则返回400

### 7. GET /video/{id}/likedby

（1）返回一个string类型的list，包含了like过的usernames

（2）如果没有该id的video返回404

### 8. GET /video/search/findByName?title={title}

（1）返回一个video的list，他们的title都符合参数

(1.1）如果没有符合的，返回空list

### 9. GET /video/search/findByDurationLessThan?duration={duration}

（1）返回一个video的list，他们的duration都比参数小

(1.1）如果没有符合条件，返回空list

### 10. 提示点：

（1）此作业会使用JPA，这样会使findByName和findByDurationLessThan方法，较易实现。

（2）不要修改VideoSvcApi，AutoGrading，AutoGradingTest

## 五、测试作业

同作业1

## 六、提交作业

同作业1

## 七、提供给你的代码

### 1. org.magnum.mobilecloud.video.repository.Video

这个类是video原数据，为了让它能够被JPA存储，要添加以下annotation：

@Entity, @Id, @GeneratedValue, and @ElementCollection.

### 2. example中的OAuth 2.0 Configuration

此代码位于：

9-VideoServiceWithOauth2/src/main/java/org/magnum/mobilecloud/video/auth

你可以直接使用这个代码，也可以自己实现

### 3. SecuredRestBuilder

这个类包裹了Retrofit库，用于构建一个支持OAuth 2.0的client。

## 八、指点

1. 如果你想要测试application不带OAuth，

那么去掉build.gradle中的一下注释：

compile("org.springframework.boot:spring-boot-starter-security:${springBootVersion}") compile("org.springframework.security.oauth:spring-security-oauth2:2.0.0.RC2") compile("org.springframework.security.oauth:spring-security-oauth2-javaconfig:1.0.0.M1")

再右击build.gradle->Refresh All即可。

2. examples很有用

3. 至少有一个类使用了@Controller

4. 有多个方法要使用@RequestMapping

5. 当add新video时，使用Spring Data Rest是不能完成该作业的，因为Spring Data Rest提供的response是不同的。（自己写的注：难道是要用 Retrofit）

6. 任何一个Contoller的方法，都可以接受一个Principal类的参数，判断user是否已经被验证了。

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

public MyObject doSomething( @PathVariable("id") String id, Principal p) {

String username = p.getName();

// Maybe you want to add this users name to

// the list of people who like a video

....

}

7. ID必须是long类型，写成int会返回400

8. 没有类会实现VideoSvcApi