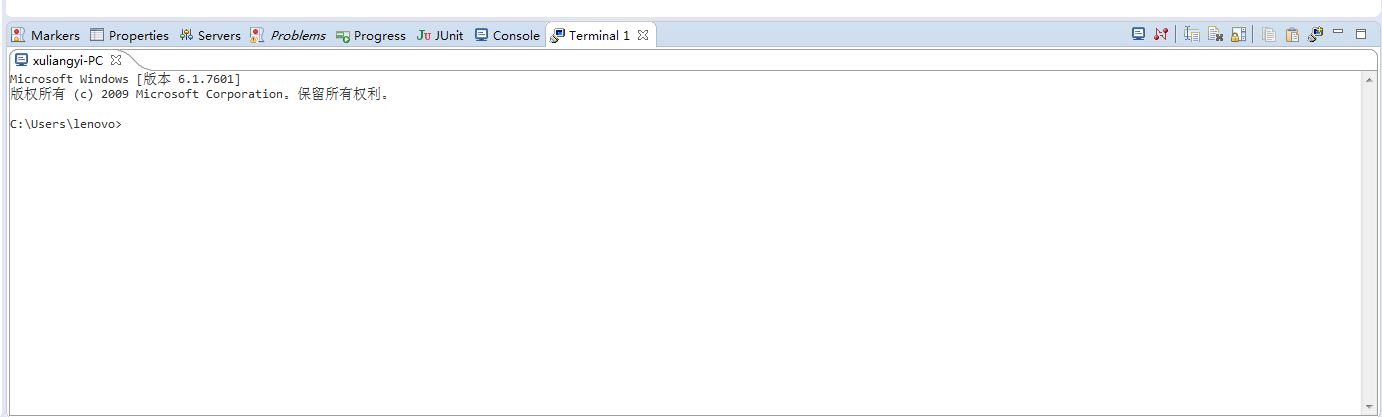
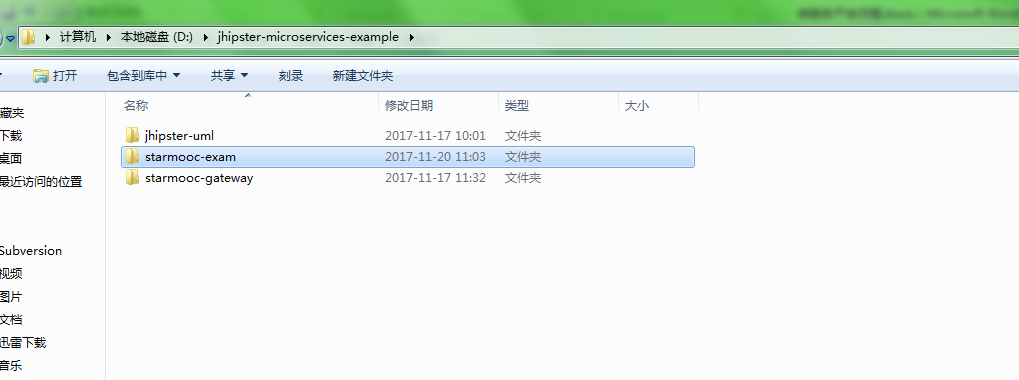
**Jhipster微服务开发流程**

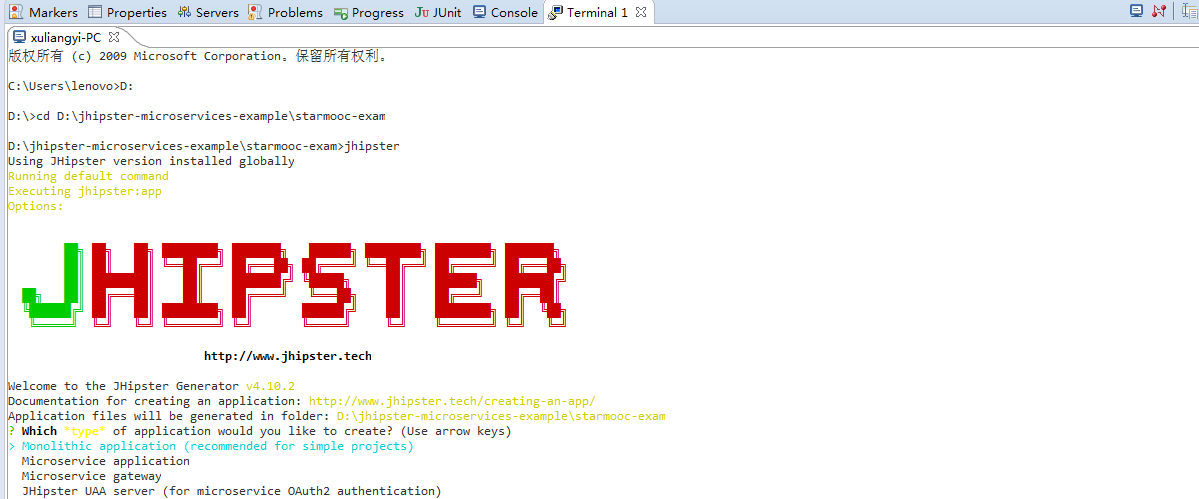
# 创建Application

eclipse使用Terminal开启cmd命令窗口(Spring Tool Suite自带Terminal插件，没有Terminal插件的请到eclipse市场下载插件)；



创建项目文件夹，进入文件夹，使用jhipster命令，开始创建项目；





创建项目过程中，需要回答给出的问题，完成对应用程序的配置。部分问题如下：

Which \*type\* of application would you like to create? (Use arrow keys) 你想创建哪种类型的应用程序？

* Monolithic application (recommended for simple projects) 整体应用程序：这是一个经典的，one-size-fits-all的应用程序。使用和开发更容易，是我们推荐的默认设置。
* Microservice application微服务应用程序：在微服务体系结构中，这是服务之一。
* Microservice gateway微服务网关：在微服务体系结构中，这是一个路由和保护请求的边缘服务器。
* JHipster UAA server (for microservice OAuth2 authentication) JHipster UAA服务器：在微服务架构中，这是一个OAuth2身份验证服务器，用于保护微服务。请参阅[JHipster UAA文档](http://www.jhipster.tech/using-uaa/)以获取更多信息。

我们选择第2个，Microservice application创建一个微服务应用程序。

What is the base name of your application? (jhipster)应用程序基本名称是？(默认为jhipster)

我们使用：starmoocExam

As you are running in a microservice architecture, on which port would like your server to run? It should be unique to avoid port conflicts. (8081)应用程序的端口号为？请使用唯一端口号，避免端口冲突(默认为8081)

我们使用：8081

What is your default Java package name?默认的jar包路径是？

我们使用：com.nercel.microservice.starmooc.exam

Which service discovery server do you want to use? (Use arrow keys) 您要使用哪个服务发现服务器？

这个是与注册管理服务相关的选项，我们选择JHipster Registry (uses Eureka, provides Spring Cloud Config support and monitoring dashboards)，使用JHipster Registry对相关微服务进行管理。

Which \*type\* of authentication would you like to use? (Use arrow keys) 您想使用哪种类型的身份验证？

~~这个问题的答案依赖于以前的答案。因为我们选择了上面的JHipster Registry，则只能使用JWT认证。~~

这里选择OAuth 2.0 / OIDC Authentication (stateful, works with Keycloak and Okta)，使用oauth进行身份验证。

Which \*type\* of database would you like to use? (Use arrow keys) 你想使用哪种类型的数据库？

我们选择：SQL (H2, MySQL, MariaDB, PostgreSQL, Oracle)

Which \*production\* database would you like to use? (Use arrow keys) 选择哪种数据库？

我们选择：MySQL；要配置它，请修改您的src/main/resources/config/application-prod.yml文件。

Which \*development\* database would you like to use? (Use arrow keys) 你想使用哪个开发数据库？

我们选择：MySQL；与生产数据库相同。要配置它，请修改您的src/main/resources/config/application-dev.yml文件。

Do you want to use Hibernate 2nd level cache? (Use arrow keys) 你想使用Hibernate二级缓存吗？

Yes, with ehcache (local cache, for a single node) 是，使用本地缓存

Yes, with HazelCast (distributed cache, for multiple nodes) 是，使用分布式缓存，用于集群

[BETA] Yes, with Infinispan (hybrid cache, for multiple nodes) 是，Infinispan分布式缓存，用于集群，测试中功能。

No 不使用

我们选择：No,不使用。

Would you like to use Maven or Gradle for building the backend? (Use arrow keys) 你想使用Maven还是Gradle来构建程序？

我们选择：maven

Which other technologies would you like to use? (Press <space> to select, <a> to toggle all, <i> to inverse selection) 您想使用哪种其他技术？

( ) Search engine using Elasticsearch 搜索引擎使用ElasticSearch

( ) API first development using swagger-codegen API首先使用swagger-codegen进行开发

( ) Asynchronous messages using Apache Kafka 使用Apache Kafka作为发布/订阅消息代理。

我们选择：都不使用，直接跳过；

Would you like to enable internationalization support? (Y/n) 你想启用国际化支持吗？

我们选择：n， 不使用国际化。

Besides JUnit and Karma, which testing frameworks would you like to use? (Press <space> to select, <a> to toggle all, <i> to inverse selection) 除了JUnit和Karma，还需要使用哪个测试框架？

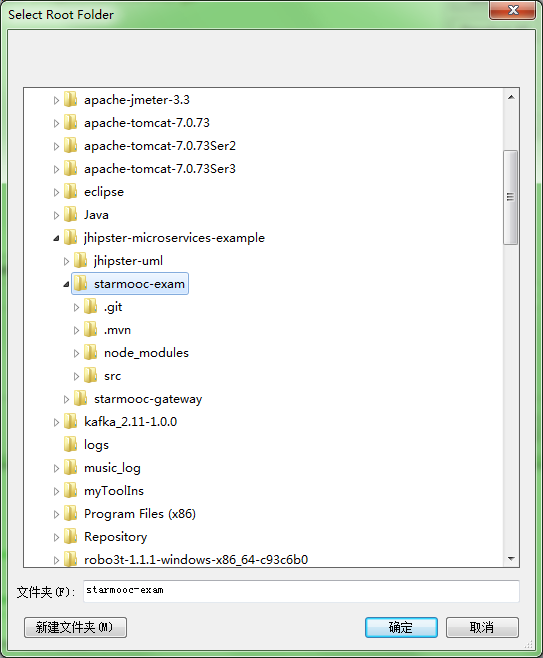
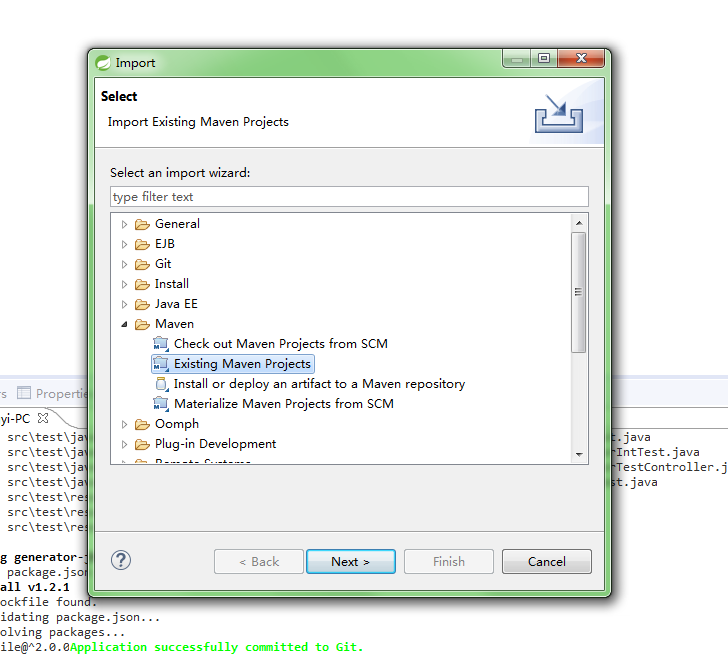
我们选择：都不使用，直接跳过；

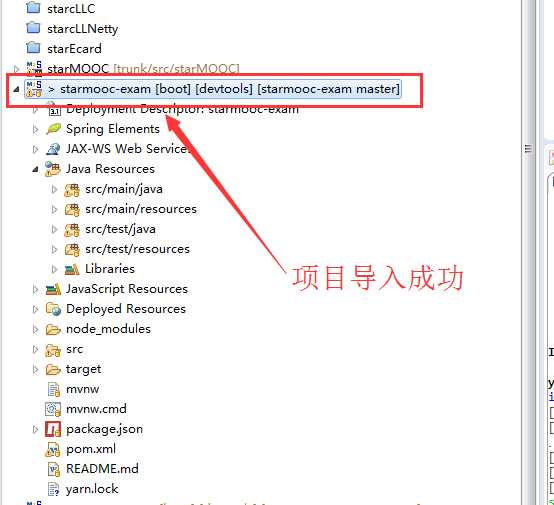
Would you like to install other generators from the JHipster Marketplace? (y/N) 还需要从JHipster应用市场中安装其他工具吗？

我们选择：N，不需要。

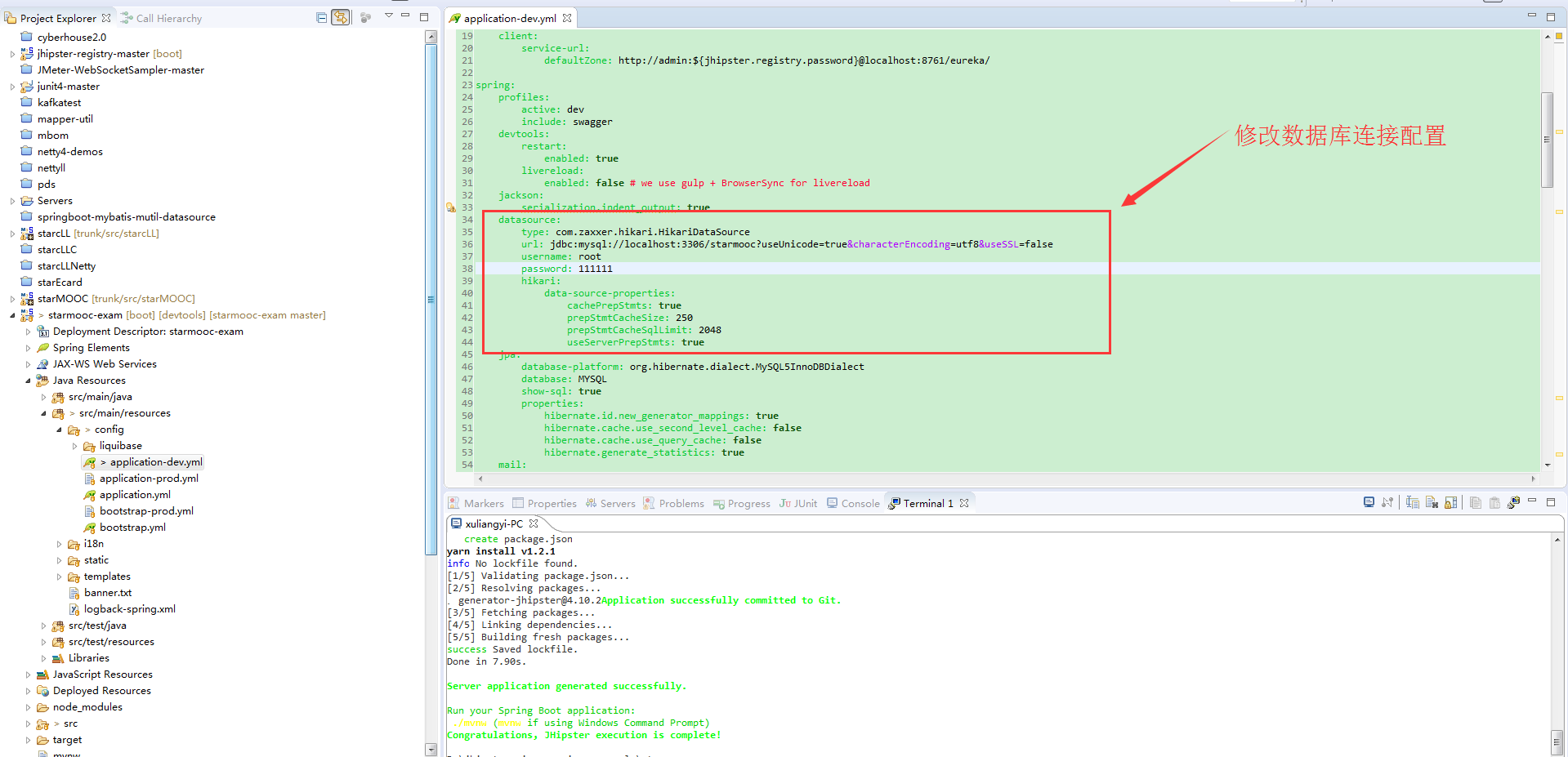
此处仅列出部分问题，如有其他配置问题，请自行研究。选择完上述问题后，JHipster将会为我们自动创建好项目。

使用eclipse，找到刚才创建好的项目，导入到eclipse工作空间。

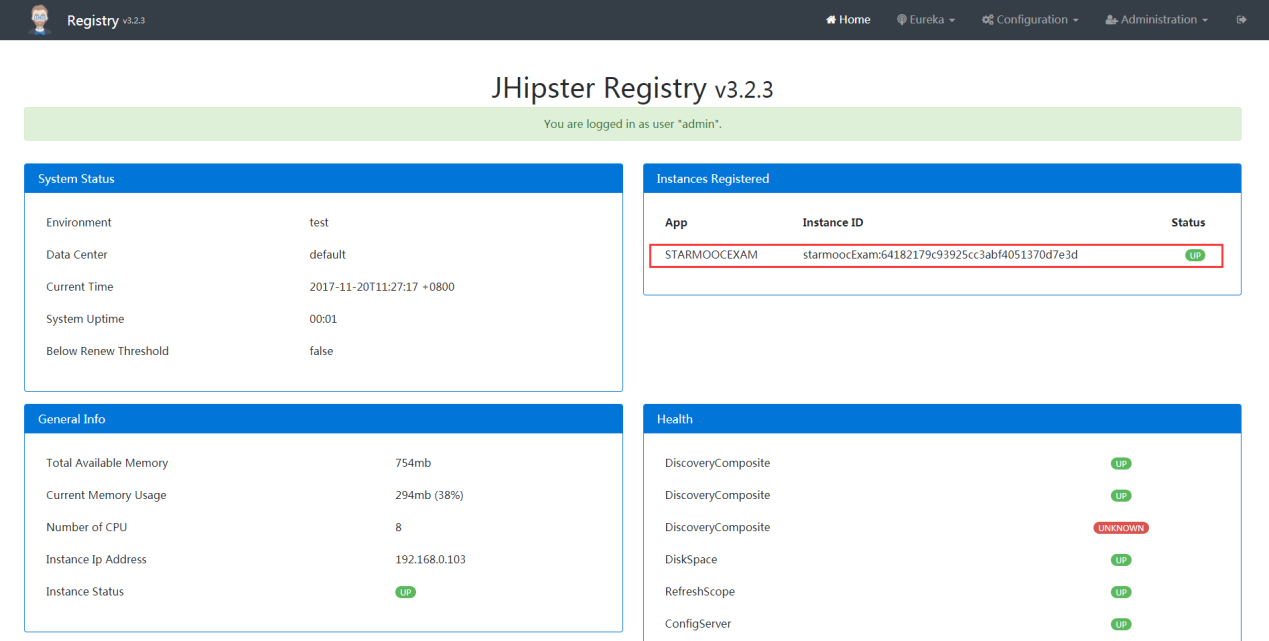




项目导入完毕后，我们修改一下数据库连接配置，打开application-dev.yml文件，修改datasource配置；



然后参考《微服务开发环境搭建.docx》文档中的操作，将本项目注册到JHipster Registry服务中。spring boot app启动JHipster Registry和starmooc-exam项目，可以看到starmooc-exam已成功注册了实例。至此，项目创建成功！以上操作如有任何操作失误，ctrl+c退出，重新创建选择即可；如果已创建完项目，发现有错误，可以直接删除项目文件，重新创建。

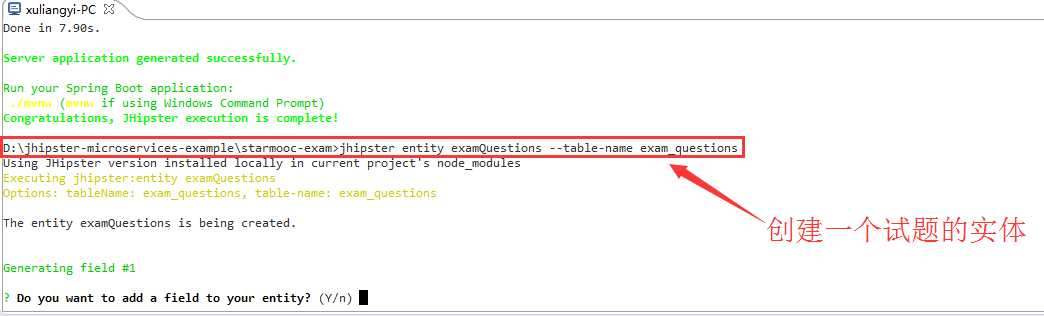


# 创建Entity

## 创建单个实体

cmd命令行中，使用jhipster entity <entityName> --[options]命令创建，以下是支持的选项。

* --table-name <table\_name> - 默认情况下，JHipster会根据你的实体名称生成一个表名，如果你希望有一个不同的表名，你可以通过传递这个选项来实现。
* --angular-suffix <suffix> - 如果你想让所有的Angular路线有一个自定义的后缀，你可以使用这个选项来传递。
* --regenerate - 这将重新生成一个现有的实体，而不会提出任何问题。
* --skip-server - 这将跳过服务器端代码，并将只生成客户端代码。
* --skip-client - 这将跳过客户端代码，并将只生成服务器端代码。
* --db - 跳过服务器端生成时指定数据库，否则不起作用。



命令生效后，会弹出问题：

Do you want to add a field to your entity? (Y/n) 是否想给实体添加一个字段？输入Y，回车确认。

What is the name of your field? 字段的名称是？输入字段名称即可。

What is the type of your field? (Use arrow keys) 字段的类型是？选择字段类型。

Do you want to add validation rules to your field? (y/N) 需要给字段添加限制条件吗？输入y，是。

Which validation rules do you want to add? (Press <space> to select, <a> to toggle all, <i> to inverse selection)需要添加哪些限制条件？(空格选择，回车确认)

( ) Required 必填非空

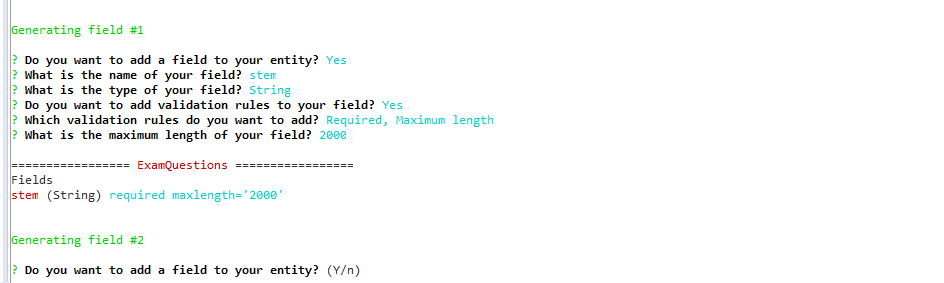
( ) Minimum length 最小长度

( ) Maximum length 最大长度

( ) Regular expression pattern 正则表达式验证

What is the maximum length of your field? 最大长度多少？输入字段的最大长度。其他选项类似。

创建完一个字段后，即可看到字段创建成功的提示，可以接着使用相同操作创建其他字段：



添加完所有字段后，会出现如下问题:

Do you want to add a relationship to another entity? (Y/n) 是否需要和其他实体创建关联关系？我们选择n,不需要。

Do you want to use a Data Transfer Object (DTO)? (Use arrow keys) 是否需要创建一个数据传输用的DTO？按需要选择。

No, use the entity directly 不需要，直接使用实体。此处请选择不需要，如选择是，自动生成的代码会使项目无法启动。

[BETA] Yes, generate a DTO with MapStruct 是，创建一个DTO

Do you want to use separate service class for your business logic? (Use arrow keys) 是否需要为业务逻辑使用独立的服务类吗？

No, the REST controller should use the repository directly 不，REST controller直接使用repository的方法。

Yes, generate a separate service class 是，创建一个独立的service

Yes, generate a separate service interface and implementation 是，创建一个独立的service接口，并创建实现类。我们选择这个。

Do you want to add filtering? (Use arrow keys) 是否需要创建过滤器？

Not needed 不需要

Dynamic filtering for the entities with JPA Static metamodel ？？？效果未知

Do you want pagination on your entity? (Use arrow keys) 是否需要分页？

No 不需要

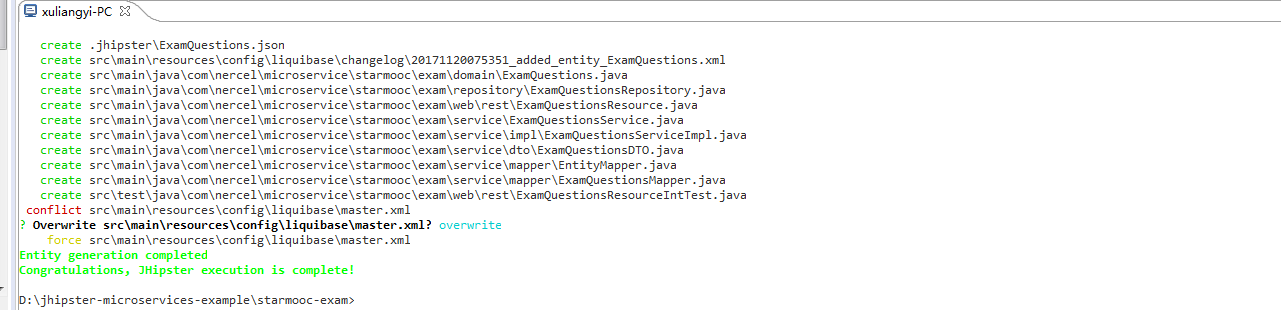
Yes, with a simple pager 一个简单的分页

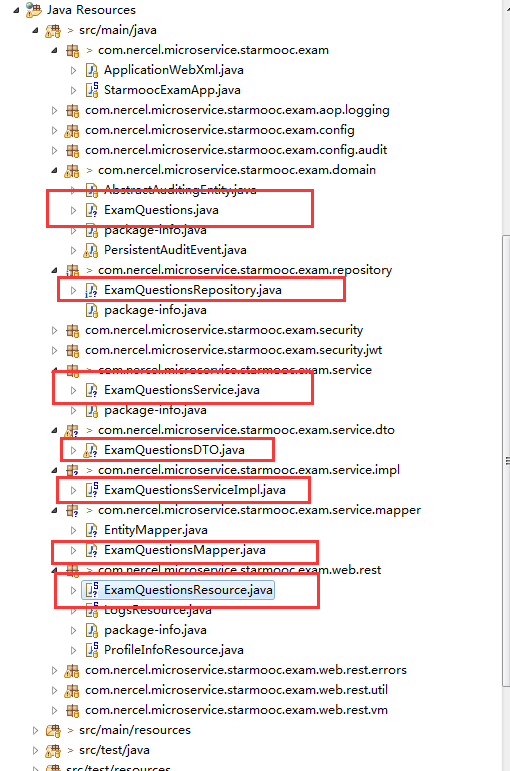
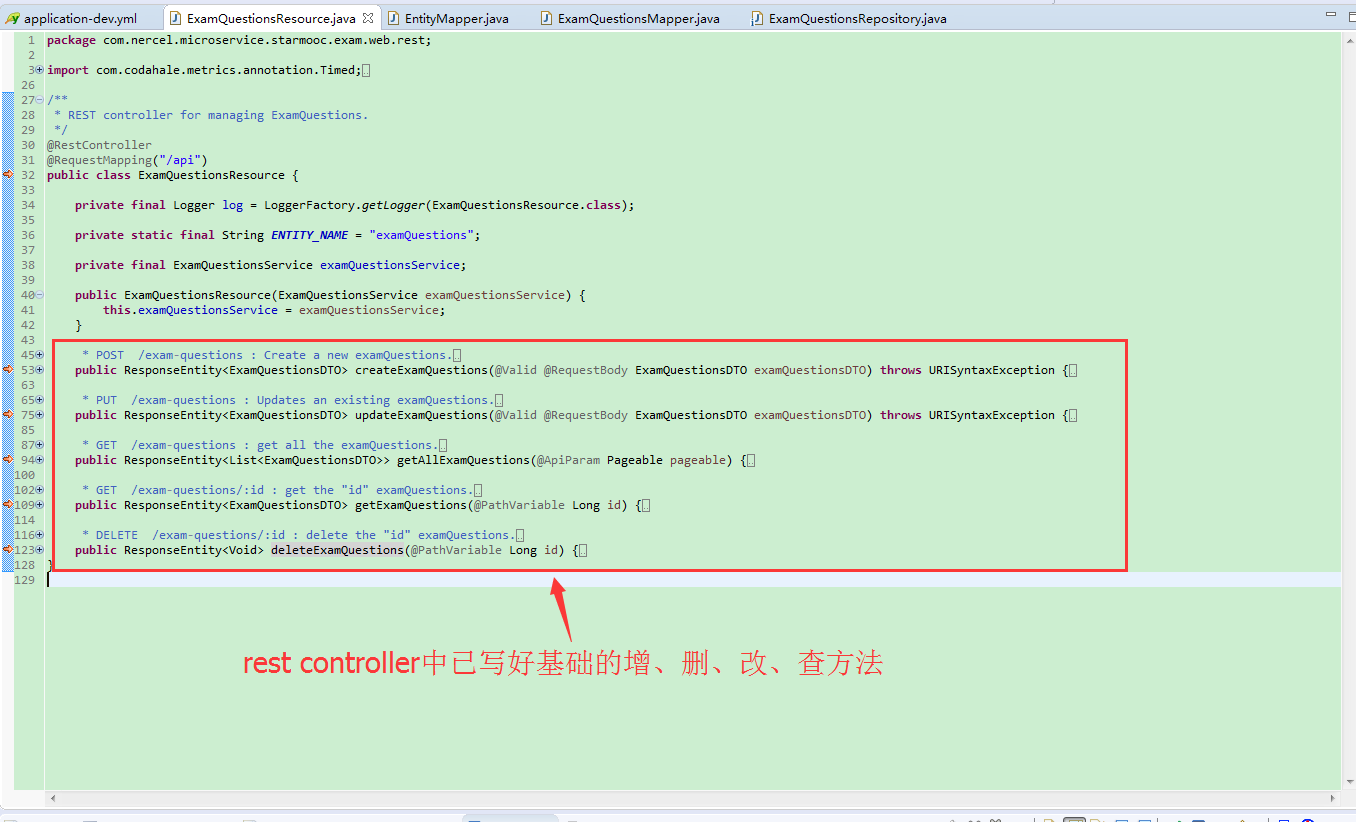
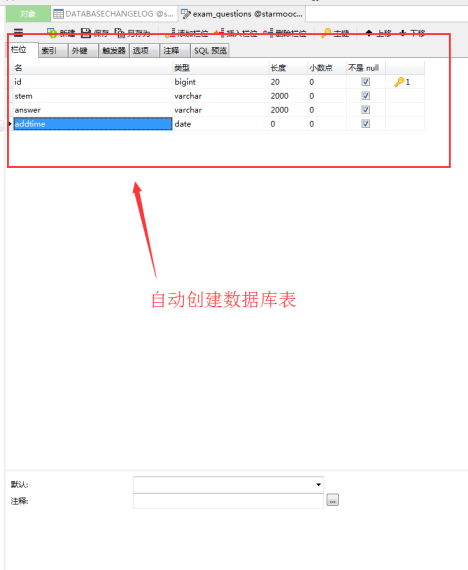
Yes, with pagination links ？？？效果未知

Yes, with infinite scroll ？？？效果未知

Overwrite src\main\resources\config\liquibase\master.xml? 覆盖重写master.xml文件？y,是

至此，一个实体创建成功，jhipster会为我们自动创建好实体相关的controller，service及serviceImpl，Mapper和Repository等类，并自动创建好数据库表(自动创建好的数据库表没有注释信息，请自行添加表、字段的注释)，写好基础的增、删、改、查方法及接口服务。

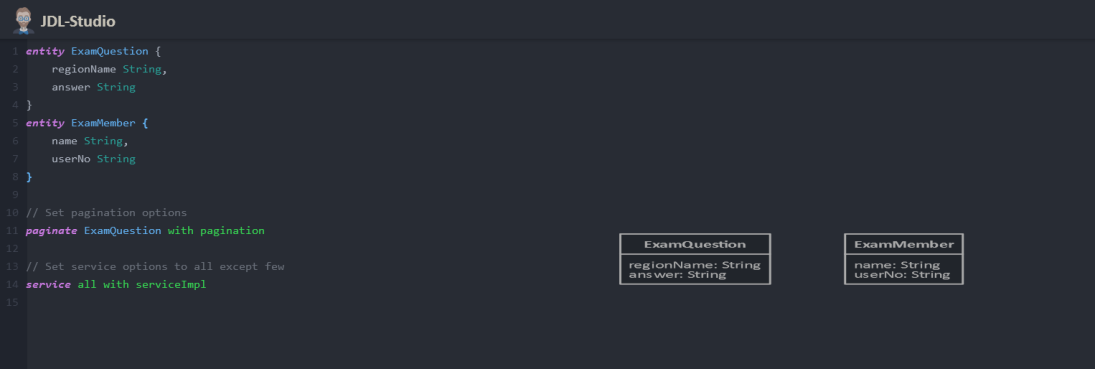


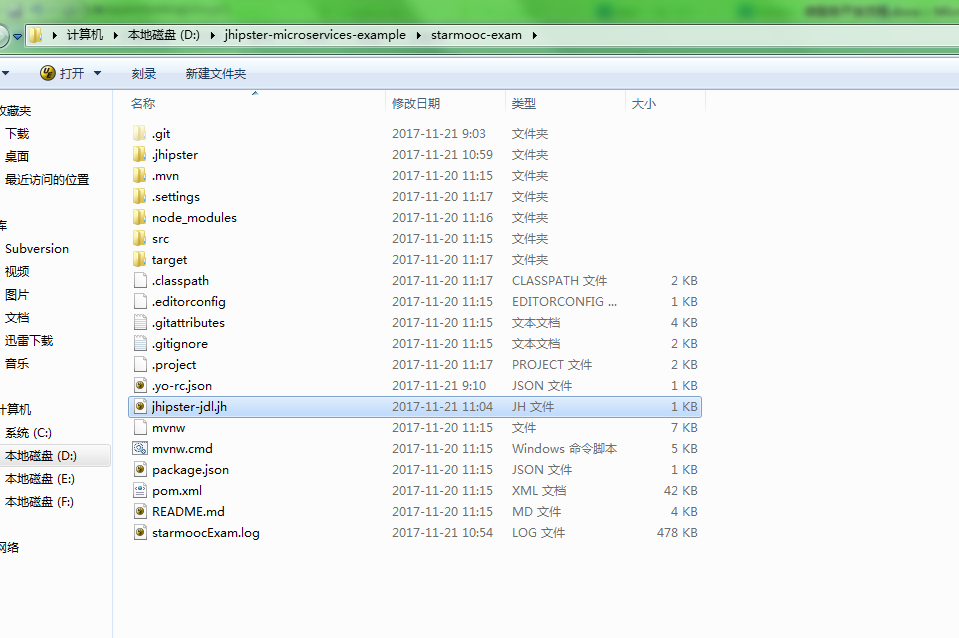
 

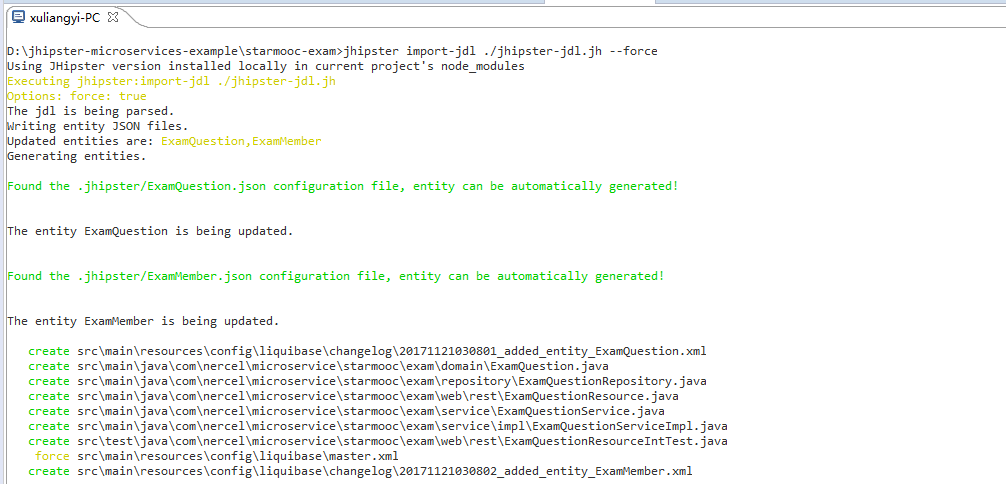
## 导入多个实体

除了单个创建实体，我们还可以使用JHipster UML工具或者JDL Studio (在线网站：http://www.jhipster.tech/jdl-studio/)批量导入多个实体。

使用JDL Studio工具，编辑好实体后，导出下载jdl文件；将jdl文件复制到项目目录，执行jhipster import-jdl ./jhipster-jdl.jh --force 命令，进行实体的批量导入操作。



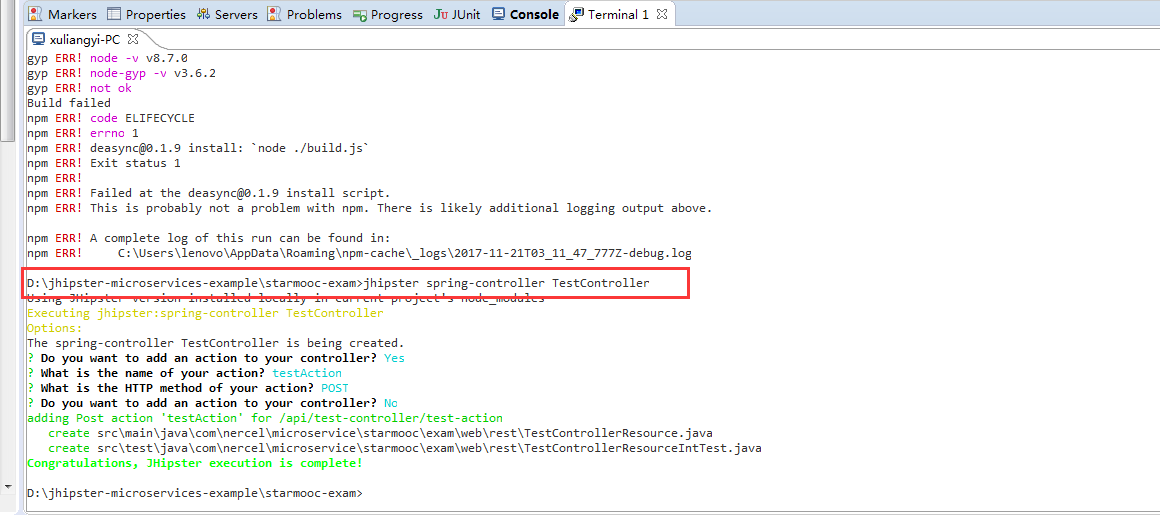




我的JHipster UML无法安装，暂时无法编写JHipster UML导入实体的方式。

# 创建Controller

使用jhipster spring-controller <controllerName>命令来创建Controller，如图：



创建Controller时，会给出几个问题：

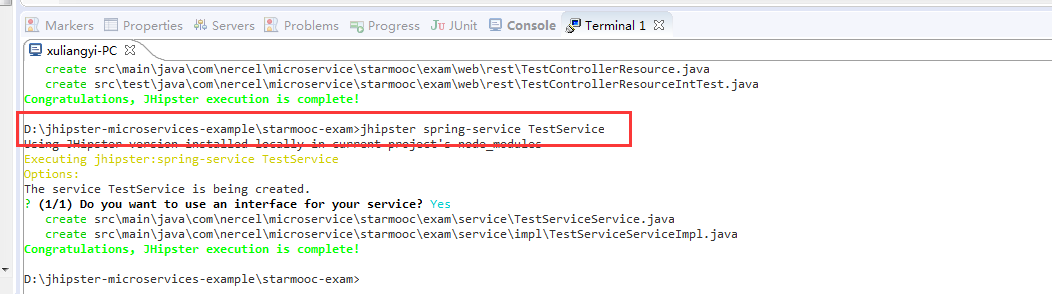
Do you want to add an action to your controller? 是否要给controller创建一个请求服务？输入Y选是，创建一个请求；

What is the name of your action? 你的请求名称？ 输入请求名称。

What is the HTTP method of your action? 选择请求方式？ POST/GET等。

# 创建Service

使用命令：jhipster spring-service <serviceName>来创建一个单独的service服务，如图：



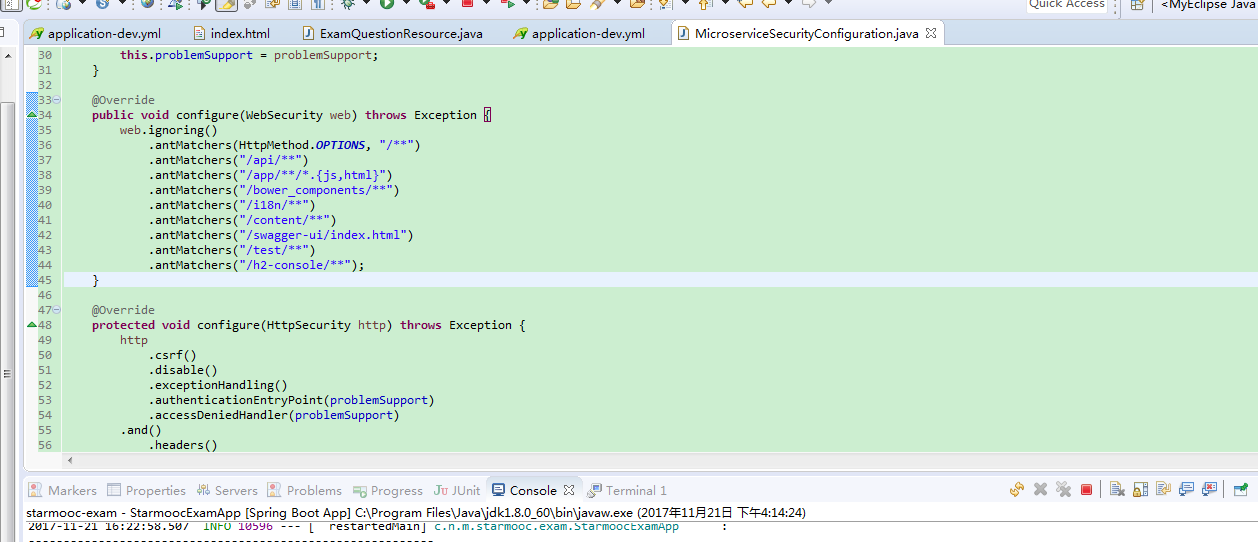
在创建service时，会给出1个配置问题：

(1/1) Do you want to use an interface for your service? 是否需要给你的service创建一个接口？Y，需要。

# 使用gateway调用接口服务

在创建好微服务相关项目后，先启动registry，gateway服务，然后启动应用微服务starmoocexam。此时，registry会为我们管理各个微服务，并提供访问统计等功能；gateway会为我们管理接口服务请求，如想调用接口服务，需要通过gateway调用。例如，如果您的网关正在运行localhost:8082，您可以指向[http：// localhost：8082 / app1 / rest / foos](http://localhost:8080/app1/rest/foos)来获取foos由微服务提供的资源app1。如果您正在尝试使用Web浏览器执行此操作，请不要忘记REST资源在JHipster中是默认保护的，所以您需要发送正确的JWT头（请参阅下面的安全要点），或者删除这些安全性微服务MicroserviceSecurityConfiguration类中的URL 。

如果需要跳过安全验证，请在MicroserviceSecurityConfiguration添加ignoring跳过网关的安全验证。



configure方法中添加忽略安全验证的url

# 微服务配置oauth验证

微服务创建完毕后，如果创建时认证方式选择的是oauth，则需要进行相关配置。本文档基于云课堂已有的oauth认证服务上进行配置，其他系统仅作参考。

* 修改application.yml文件中的security配置(配置中采用的是云课堂测试服务器的oauth认证地址)：

security:

basic:

enabled: **false**

oauth2:

client:

access-token-uri: http://119.97.166.137:1002/starOauth/oauth/token

user-authorization-uri: http://119.97.166.137:1002/starOauth/oauth/authorize

client-id: starMOOC-exam

client-secret: starMOOC-exam-secret

client-authentication-scheme: form

scope: read

resource:

userInfoUri: http://119.97.166.137:1002/starOauth/user\_info

tokenInfoUri: http://119.97.166.137:1002/starOauth/oauth/token

preferTokenInfo: **false**

* com.nercel.microservice.starmooc.security.oauth2路径下，创建自定义的认证服务service CachedUserInfoTokenServices，CachedUserInfoTokenServices代码：



* 修改MicroserviceSecurityConfiguration类中，userInfoTokenServices方法，修改为(其实就是使该方法返回CachedUserInfoTokenServices对象)：

@Bean

@Primary

**public** CachedUserInfoTokenServices userInfoTokenServices(PrincipalExtractor principalExtractor, AuthoritiesExtractor authoritiesExtractor) {

CachedUserInfoTokenServices userInfoTokenServices =

**new** CachedUserInfoTokenServices(resourceServerProperties.getUserInfoUri(), resourceServerProperties.getClientId());

userInfoTokenServices.setPrincipalExtractor(principalExtractor);

userInfoTokenServices.setAuthoritiesExtractor(authoritiesExtractor);

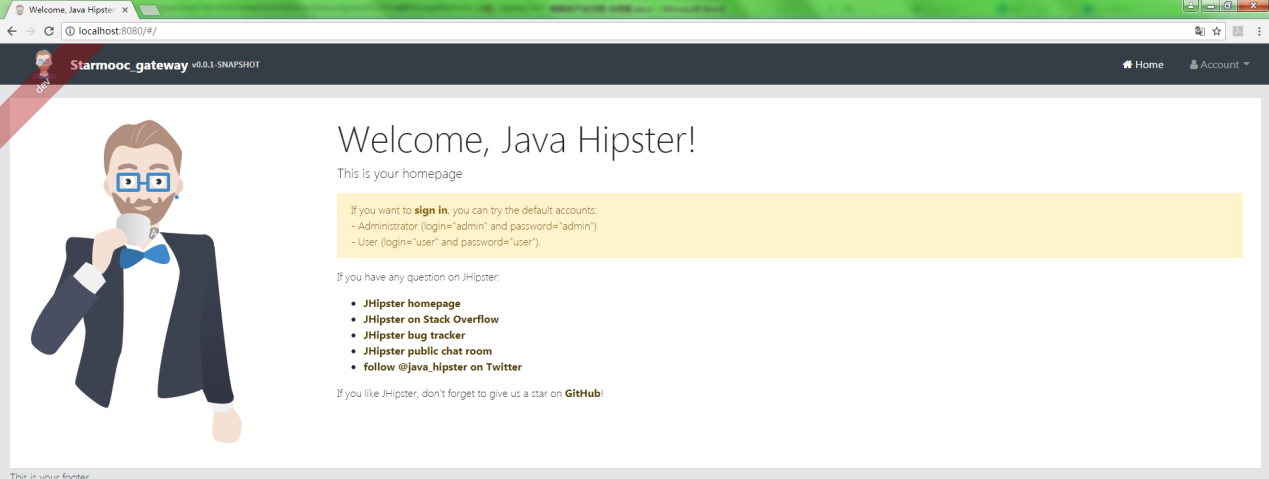
**return** userInfoTokenServices;

}

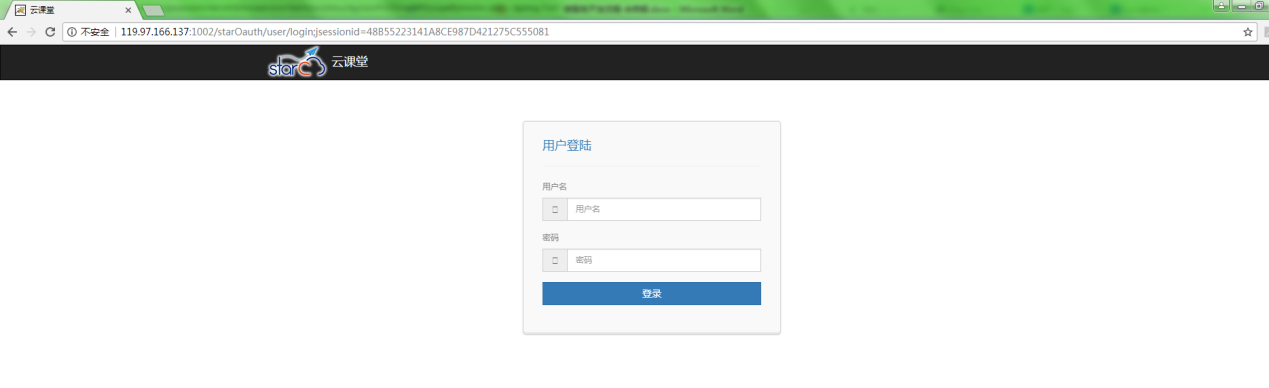
* 修改SimpleAuthoritiesExtractor类和SimplePrincipalExtractor类，类文件如下：



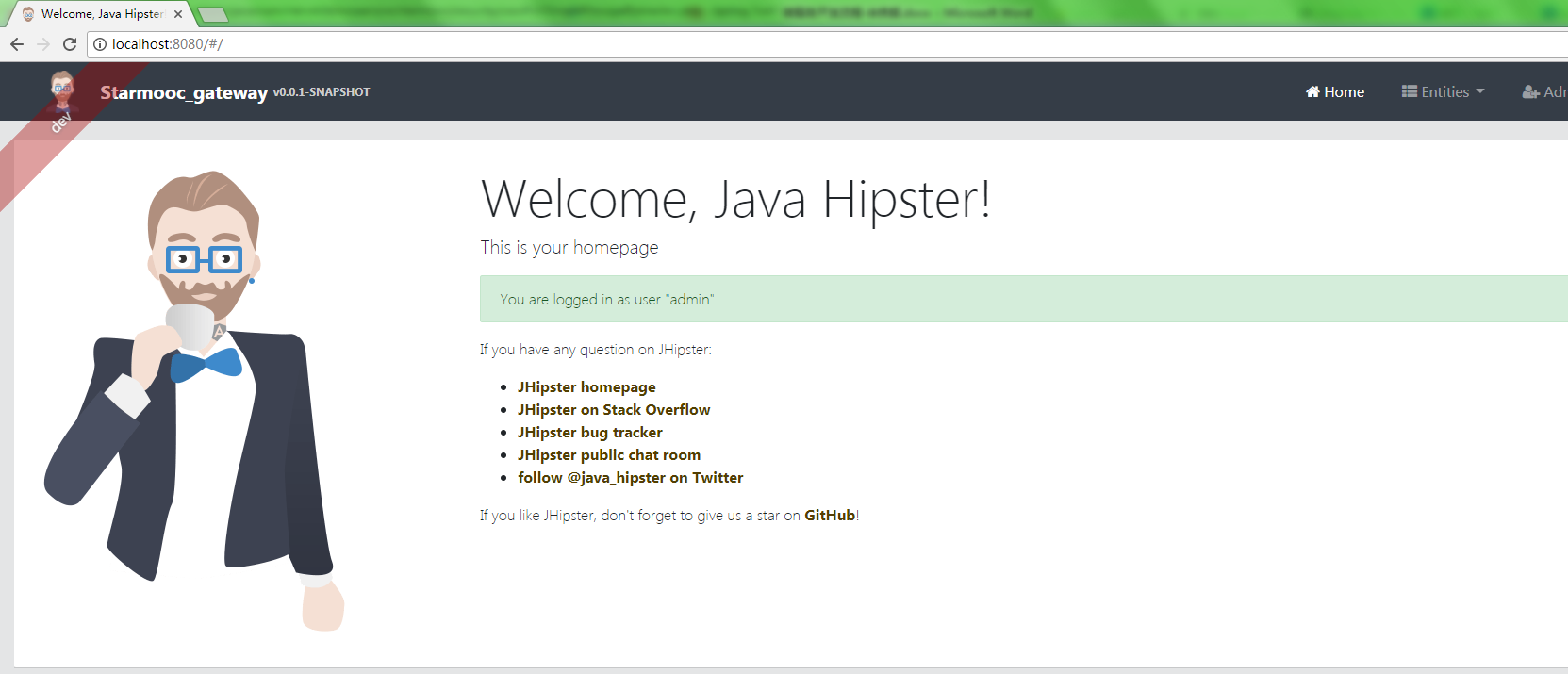
修改以上类的原因，主要是适配云课堂oauth服务返回的user\_info信息，以上信息修改完毕后，进入微服务网关登录，会跳转到云课堂oauth登录页面，输入云课堂账号登陆，返回网关页面提示登录成功，如图：



微服务网关页面



云课堂oauth登录页面



登录成功

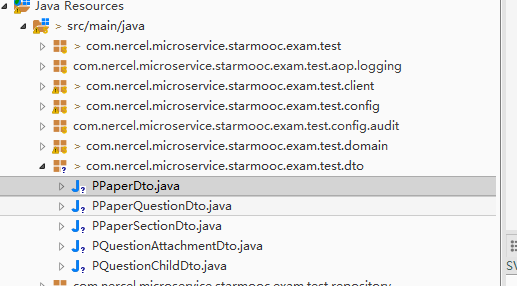
# 微服务之间调用接口服务

微服务之间调用接口服务使用Feign调用；只需创建interface类ExamPaperClient；

并在ExamPaperClient中指定需要调用的其他微服务的方法，如下图：



同时在自己的微服务中创建好相关的接收对象类：



最后直接使用依赖注入调用该方法：



# 可能出现的问题

1.创建实体时，无法自动产生数据库表？

解决方法：删除数据库中DATABASECHANGELOG表中对应实体的创建记录，重新启动spring boot程序，程序将会为我们自动创建数据库表。

2.添加的rest controller请求无法访问，出现“Full authentication is required to access this resource”错误。

解决办法：这是网关请求的安全验证，暂时先在MicroserviceSecurityConfiguration中对加入需要忽略掉的请求url,之后完善此验证。