**iCloud云平台框架方案**

**设计**

|  |  |
| --- | --- |
| Project Code |  |
| Name | iCloud云平台框架 |
| Doc. Number |  |
| Security |  |
| Version | v1.0 |
| Date | 2018.09.07 |

**更改记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Seq.**  **序号** | **Version**  **版本** | **Modification Place & Content**  **更改内容** | **Author/Date**  **更改人/日期** | **Audit/Date**  **审核人/日期** | **Approver/Date**  **批准人/日期** |
| **1** | v1.0 | Creating a draft version | 宋强 /2018.09.07 |  |  |
| **2** |  |  |  |  |  |
| **3** |  |  |  |  |  |
| **4** |  |  |  |  |  |
| **5** |  |  |  |  |  |
| **6** |  |  |  |  |  |
| **7** |  |  |  |  |  |
| 8 |  |  |  |  |  |

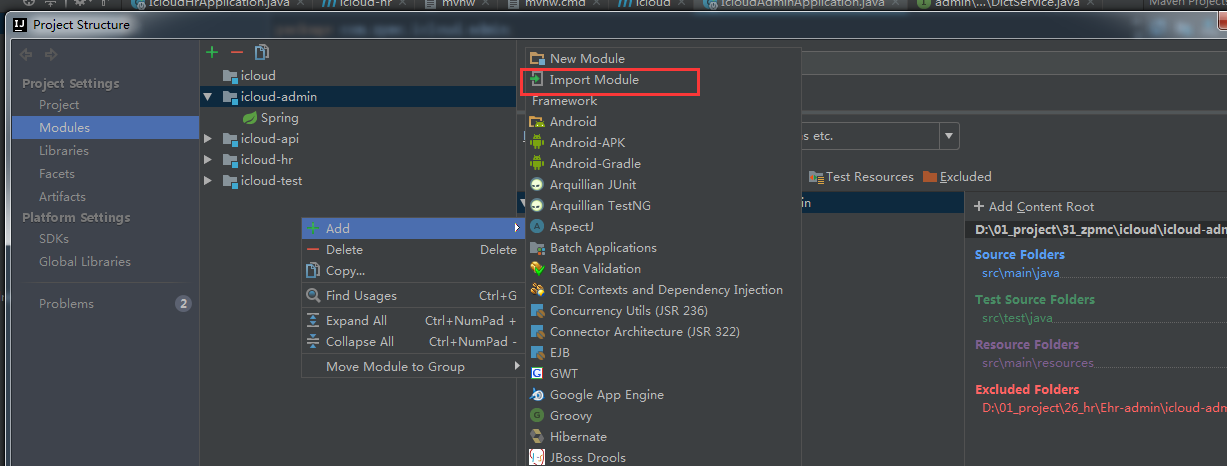
## Intellij新建项目并集成JPA等插件

## 多模块项目搭建

### IDEA 导入多个Module，多个Module在同一个Project 下显示

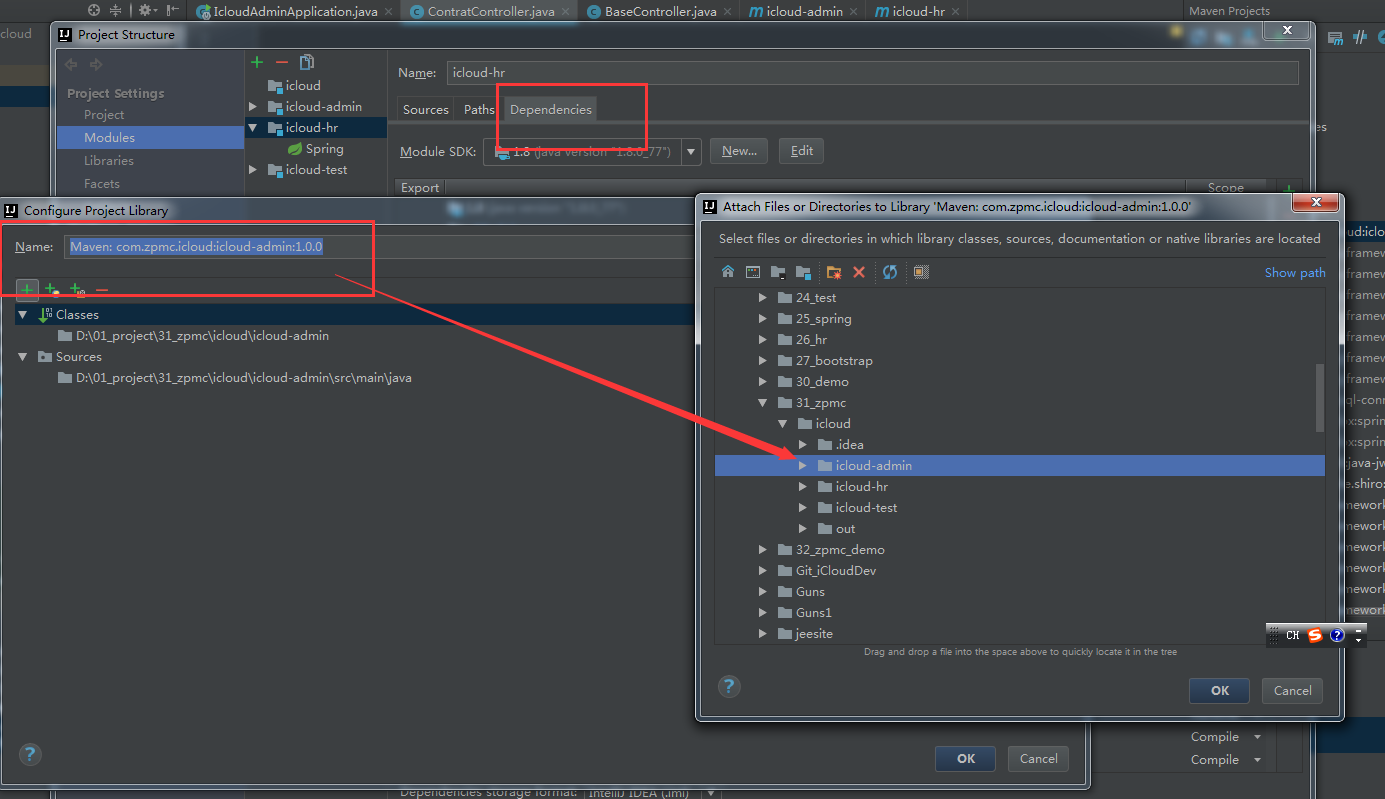
（1）打开IDEA , 点击右上角  File  --->   Project Structure

（2）Project Settings 会默认打开，点击  Modules --> 右侧的空白处（ 绿色 +）  鼠标右键



### 解决不同模块之间不能引用

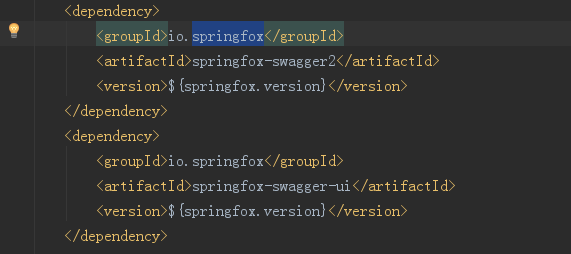
如：icloud-hr要引用icloud-admin模块



## 集成Swagger

参考：<https://www.cnblogs.com/fengli9998/p/7522973.html>

### jar包的引入



### swagger的配置启动类编写

访问地址: <http://localhost:10087/icloud-admin/swagger-ui.html>

## Shiro

## JWT(JSON Web Token)

需要把Web应用做成无状态的，即服务器端无状态，就是说服务器端不会存储像会话这种东西，而是每次请求时access\_token进行资源访问。这里我们将使用 JWT，基于散列的消息认证码，使用一个密钥和一个消息作为输入，生成它们的消息摘要。该密钥只有服务端知道。访问时使用该消息摘要进行传播，服务端然后对该消息摘要进行验

**认证步骤:**

1. 客户端第一次使用用户名密码访问认证服务器，服务器验证用户名和密码，认证成功,使用用户密钥生成JWT并返回
2. 之后每次请求客户端带上JWT
3. 服务器对JWT进行验证

## Redis

Windows下启动：

Cmd下执行：redis-server.exe redis.windows.conf

## Dubbo-admin管理界面安装

（1）下载dubbo-admin-2.8.4.war包，解压到tomcat中

D:\01\_project\32\_zpmc\_demo\apache-tomcat-8.5.8\webapps\dubbo-admin-2.8.4.war

~~（2）修改dubbo.properties文件，里面指向Zookeeper ，使用的是Zookeeper 的注册中心。~~

（3）输入<http://localhost:8080/dubbo-admin-2.8.4>

用户名/密码: root/root

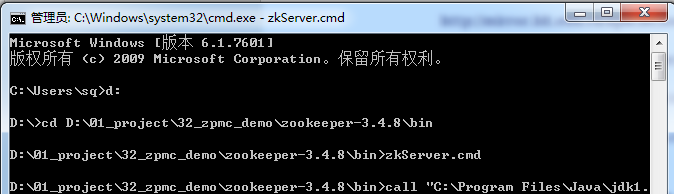
## Zookeper安装

### 启动 Zookeeper

下载:<http://www.apache.org/dyn/closer.cgi/zookeeper/>



Cmd到：D:\01\_project\32\_zpmc\_demo\zookeeper-3.4.8\bin



### 打包制作Zkui

安装zkui，可以参考[zkui的项目地址](https://github.com/DeemOpen/zkui" \t "_blank)来安装，它提供了一个管理界面

下载地址：<https://github.com/DeemOpen/zkui>

（1）cd D:\01\_project\32\_zpmc\_demo\zkui-master

（2）打包生成jar

D:\01\_project\32\_zpmc\_demo\zkui-master>mvn clean install

（3）拷贝config.cfg 文件到jar包对应目录下，修改zookeeper对应配置

（4）运行jar

D:\01\_project\32\_zpmc\_demo\zkui-master\target>java -jar zkui-2.0-SNAPSHOT-jar-wi

th-dependencies.jar

（5）浏览器运行http://localhost:9090/

## SpringBoot Admin

https://github.com/codecentric/spring-boot-admin

http://codecentric.github.io/spring-boot-admin/2.1.1/

官网：

It provides the following features for registered application.

Show health status

Show details, like

JVM & memory metrics

micrometer.io metrics

Datasource metrics

Cache metrics

Show build-info number

Follow and download logfile

View jvm system- & environment-properties

View Spring Boot Configuration Properties

Support for Spring Cloud's postable /env- &/refresh-endpoint

Easy loglevel management

Interact with JMX-beans

View thread dump

View http-traces

View auditevents

View http-endpoints

View scheduled tasks

View and delete active sessions (using spring-session)

View Flyway / Liquibase database migrations

Download heapdump

Notification on status change (via e-mail, Slack, Hipchat, ...)

Event journal of status changes (non persistent)

功能:

显示健康状况

显示详细信息，例如

JVM和内存指标

micrometer.io指标

数据源指标

缓存指标

显示构建信息编号

关注并下载日志文件

查看jvm系统和环境属性

查看Spring Boot配置属性

支持Spring Cloud的postable / env-和/ refresh-endpoint

轻松的日志级管理

与JMX-beans交互

查看线程转储

查看http跟踪

查看auditevents

查看http-endpoints

查看计划任务

查看和删除活动会话（使用spring-session）

查看Flyway / Liquibase数据库迁移

下载heapdump

状态变更通知（通过电子邮件，Slack，Hipchat，......）

状态更改的事件日志（非持久性）

### Server

### Client

加入步骤：

1. pom加入坐标

<!--springboot admin客户端-->  
<dependency>  
 <groupId>de.codecentric</groupId>  
 <artifactId>spring-boot-admin-starter-client</artifactId>  
 <version>2.1.1</version>  
</dependency>

1. 属性配置

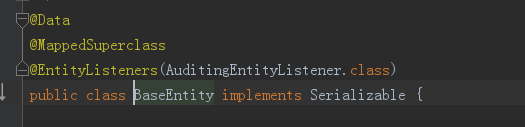
*#springboot admin***management:  
 endpoints:  
 web:  
 exposure:  
 include:** "\*"

## JPA注解

### @MappedSuperclass：

通过这个注解，我们可以将该实体类当成基类实体，它不会隐射到数据库表，但继承它的子类实体在隐射时会自动扫描该基类实体的隐射属性，添加到子类实体的对应数据库表中

标识的不能再有@Entity或@Table注解



### @Column

@javax.persistence.Column(length=15,nullable=false,columnDefinition="",insertable=true,scale=10,table="",updatable=true)

@Column注解指定字段的详细定义

name:字段的名称,默认与属性名称一致

nullable:是否允许为null,默认为true

unique:是否唯一,默认为false

length:字段的长度,仅对String类型的字段有效

columnDefinition:表示该字段在数据库中的实际类型

通常ORM框架可以根据属性类型自动判断数据库中字段的类型,

但是对于Date类型仍无法确定数据库中字段类型究竟是DATE,TIME还是TIMESTAMP,

此外,String的默认映射类型为VARCHAR,如果要将String类型映射到特定数据库的BLOB或TEXT字段类型,该属性非常有用

如: @Column(name="BIRTH",nullable="false",columnDefinition="DATE")

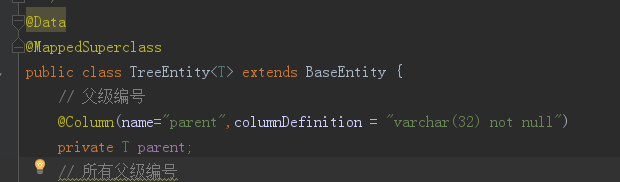
insertable:默认情况下,JPA持续性提供程序假设所有列始终包含在 SQL INSERT 语句中。

如果该列不应包含在这些语句中，请将 insertable 设置为 false

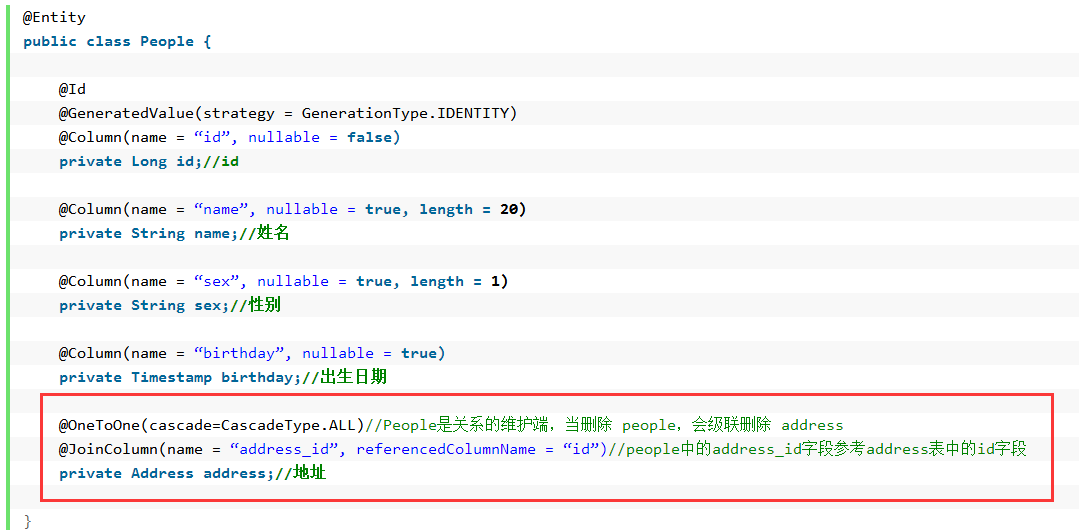
updatable：列始终包含在 SQL UPDATE 语句中。如果该列不应包含在这些语句中，请将 updatable 设置为 false

table:实体的所有持久字段都存储到一个其名称为实体名称的数据库表中,如果该列与 @SecondaryTable表关联

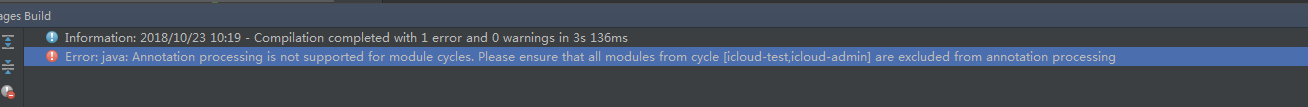
需将 name 设置为相应辅助表名称的String名称



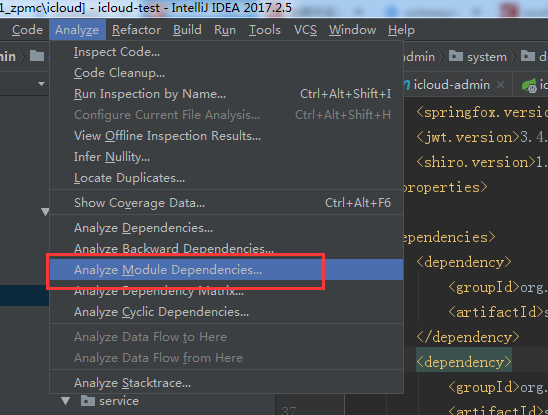
### @OneToOne



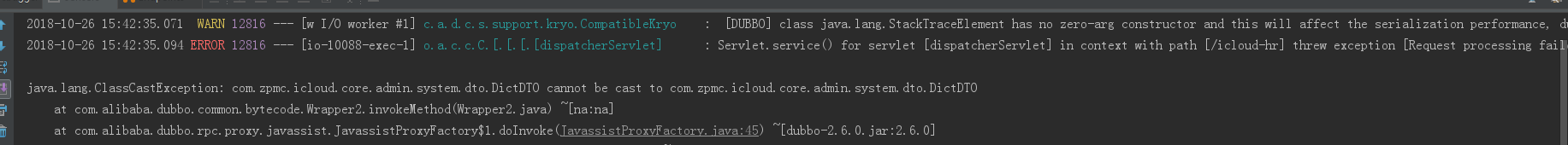
# 搭建框架异常处理



解决方案：分析模块之间的依赖关系，删除掉循环依赖关系

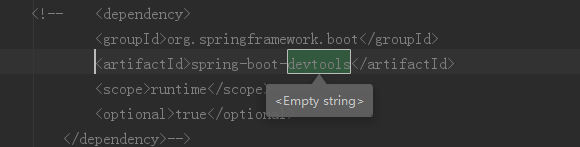


## SpringBoot整合Dubbo报错： java.lang.ClassCastException



原因：用Dubbo做过几个项目了，从来没有碰到过Caused by: java.lang.ClassCastException这种错误，最后经过反复的测试发现原来是因为依赖了SpringBoot的开发者工具才出现的这一个问题，只要把下面的依赖去掉就解决了这一个问题：

解决方案：pom.xml文件中取消spring-boot-devtools的依赖



# Maven常用命令

1、mvn compile 编译,将Java 源程序编译成 class 字节码文件。

2、mvn test 测试，并生成测试报告

3、mvn clean 将以前编译得到的旧的 class 字节码文件删除

4、mvn pakage 打包,动态 web工程打 war包，Java工程打 jar 包。

5、mvn install 将项目生成 jar 包放在仓库中，以便别的模块调用

# 多租户

# 启动顺序

## Icloud-eureka

访问地址：http://localhost:1000/

## Icloud-config

访问地址：<http://localhost:1001/auth-service-dev.yml>

SpringCloud Config为分布式系统外部化配置提供了服务器端和客户端的支持。

访问规则：

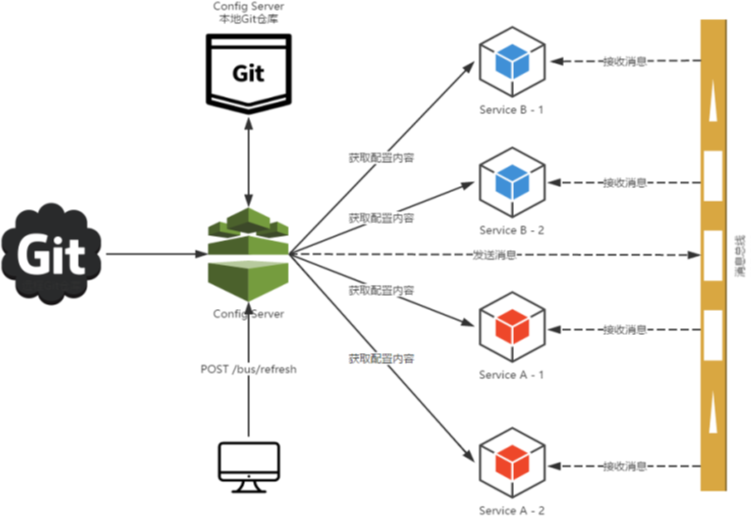
　/{appication}/{profile}/[{label}]

　/{application}-{profile}.yml

/{application}-{profile}.properties

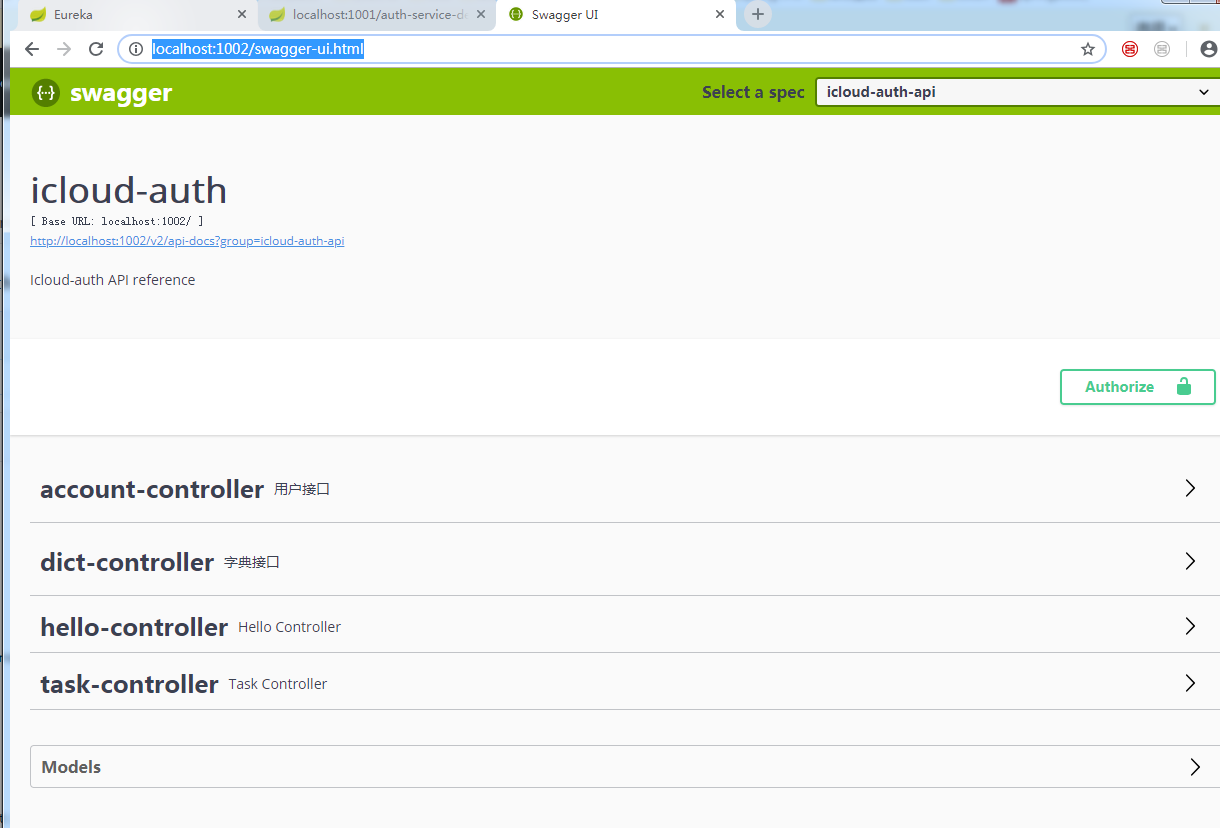
/{label}/{application}-{profile}.properties

/{label}/{application}-{profile}.yml



## Icloud-auth

访问地址：<http://localhost:1002/swagger-ui.html>



## Icloud-gateway

## Icloud-monitor

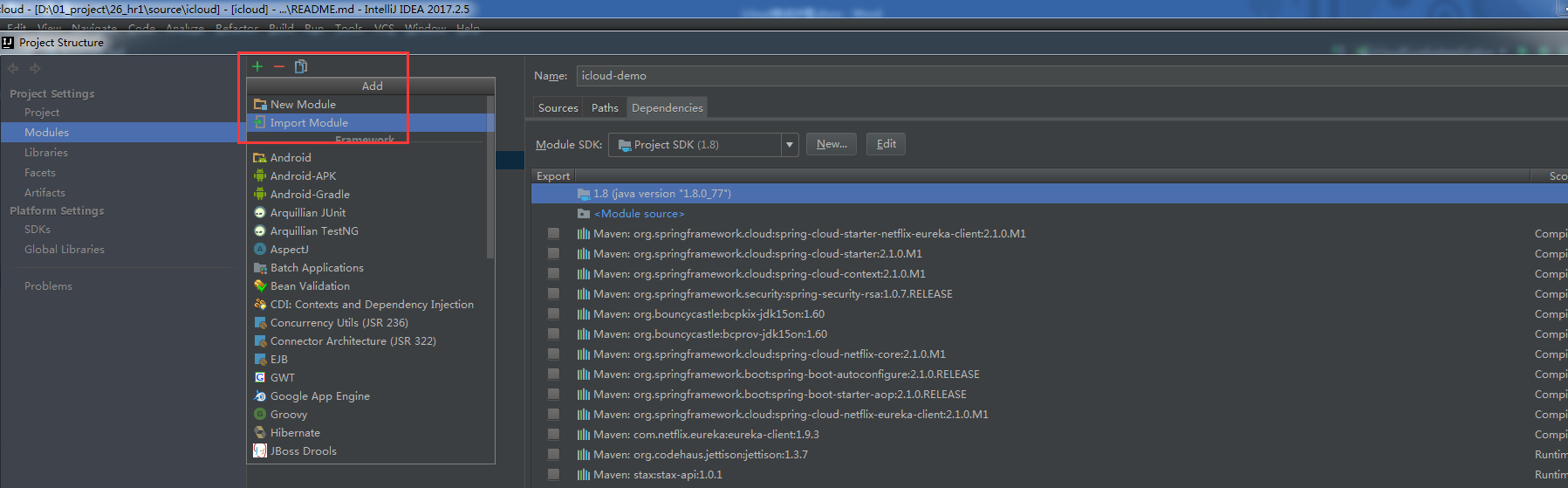
## Icloud-demo

在spring Cloud Netflix栈中，各个微服务都是以HTTP接口的形式暴露自身服务的，因此在调用远程服务时就必须使用HTTP客户端。我们可以使用JDK原生的URLConnection、Apache的Http Client、Netty的异步HTTP Client, Spring的RestTemplate。但是，用起来最方便、最优雅的还是要属Feign了

Feign是一种声明式、模板化的HTTP客户端。在Spring Cloud中使用Feign, 我们可以做到使用HTTP请求远程服务时能与调用本地方法一样的编码体验，开发者完全感知不到这是远程方法，更感知不到这是个HTTP请求。

# 错误总结

## 导入项目之后发现项目目录结构没有Java源码和Source等标志，服务起不来。



## mysq数据库编码utf-8/ utf8mb4

utf-8编码可能2个字节、3个字节、4个字节的字符，但是[MySQL](http://lib.csdn.net/base/14)的utf8编码只支持3字节的数据，而移动端的表情数据是4个字节的字符。如果直接往采用utf-8编码的[数据库](http://lib.csdn.net/base/14)中插入表情数据

修改mysql默认编码

修改mysql配置文件my.cnf（windows为my.ini）



[client]

port=3306

default-character-set=utf8mb4

[mysql]

default-character-set=utf8mb4   
[mysqld]

#add sq start

character-set-server = utf8mb4

collation-server = utf8mb4\_unicode\_ci

init\_connect='SET NAMES utf8mb4'

skip-character-set-client-handshake = true

### 服务启动/关闭

net start mysql/net stop mysql

### cmd下运行mysql

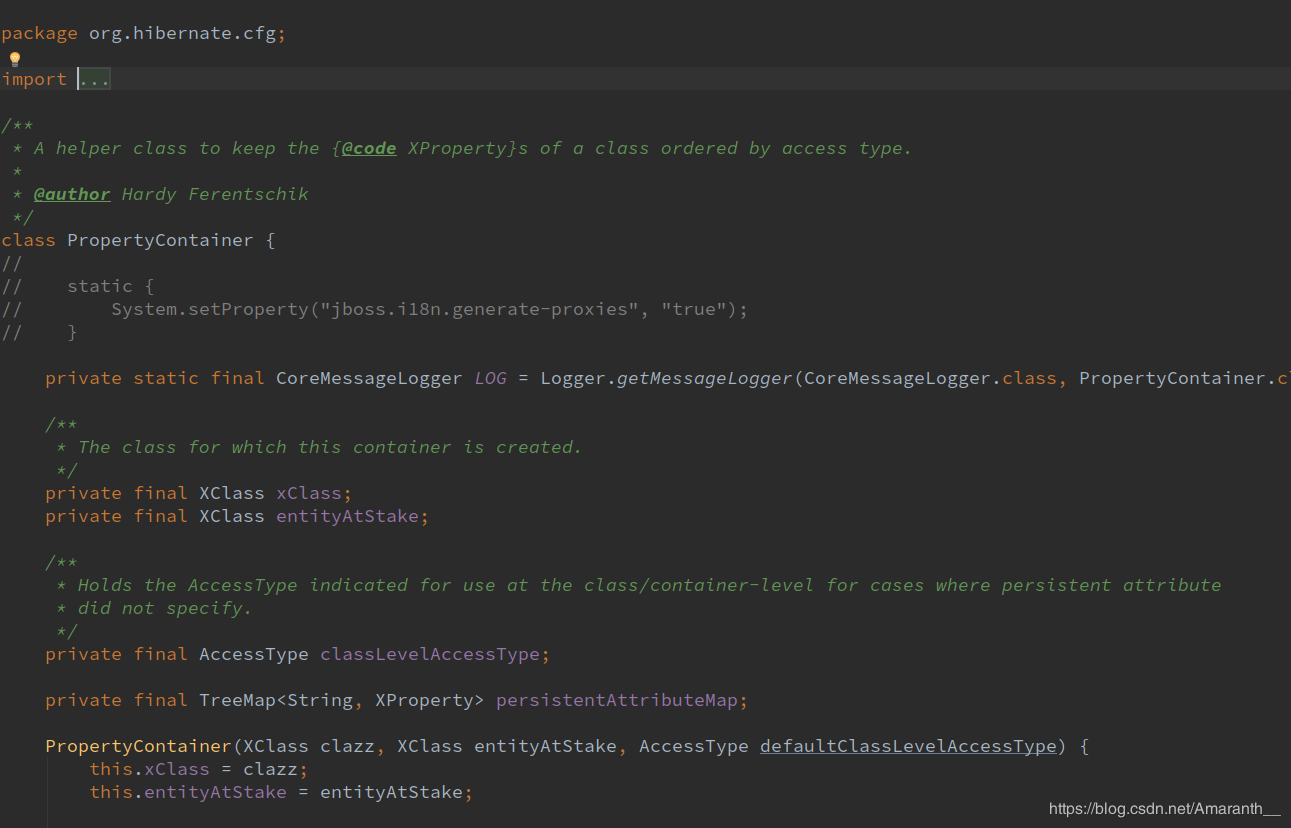
在高级系统变量下的path里添加C:\Program Files (x86)\MySQL\MySQL Server 5.5\bin

### 连接mysql

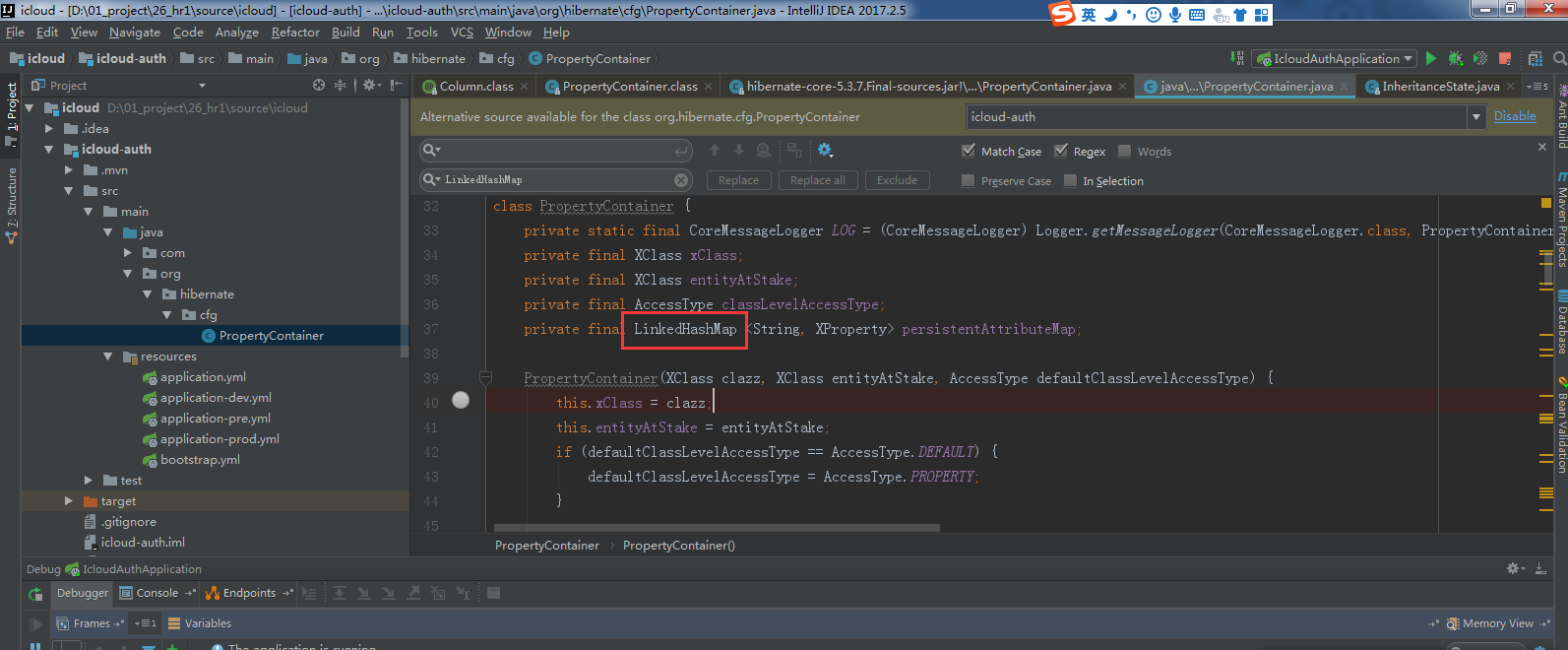
mysql -uroot -p2210

## Hibernate控制生成字段生成顺序

启动项目创建完之后你会发现数据库字段和实体类不一样 是乱序的 是因为hibernate源码中用的是TreeMap存储实体类字段属性是无序的

后来我就把本类中所有的TreeMap替换成LinkedHashMap 就变成有序的了

在本项目中创建一个和源码类一样的包结构



# 参考文档

## 微服务架构下的分布式Session管理

https://blog.csdn.net/t0591/article/details/53636599