iTalker Study

**推送平台**：个推 <http://www.getui.com/cn/index.html>

包名：net.zhouxu.italker.push

木聊 ： AppID： CKelVFVSEs6vIsWNz2HMh5

AppSecret： VPU2cWz1oc8swbvMsFuNC5

AppKey： nJz0anafJE9nKD7AGBM6u1

MasterSecret： 0IIt3gJ1138aLXw8Yw2hW2

**阿里云相关账户：**

官网：<https://www.aliyun.com/>

|  |  |
| --- | --- |
| AccessKeyId | AccessKeySecret |
| LTAIV20xNMgFNlXJ | NQWakOanSSpe62GgP17MvW0fdtRJgN |

# 项目配置

Bulid.gradle

group 'iTalker'  
version '1.0-SNAPSHOT'  
  
apply plugin: 'java'  
apply plugin: 'war'  
  
sourceCompatibility = 1.8  
  
repositories {  
 mavenCentral()  
}  
  
dependencies {  
 // Gson json  
 compile 'com.google.code.gson:gson:2.8.0'  
 // Guava java 类封装  
 compile 'com.google.guava:guava:21.0'  
  
 // Jersey 轻量级Restful接口框架  
 compile 'org.glassfish.jersey.core:jersey-client:2.26-b03'  
 compile 'org.glassfish.jersey.core:jersey-server:2.26-b03'  
 compile 'org.glassfish.jersey.containers:jersey-container-servlet:2.26-b03'  
 // 也是一个 Json 解析库  
 compile 'org.glassfish.jersey.media:jersey-media-json-jackson:2.26-b03'  
  
 // 数据库操作框架  
 // https://mvnrepository.com/artifact/org.hibernate/hibernate-core  
 compile 'org.hibernate:hibernate-core:5.2.9.Final'  
 // https://mvnrepository.com/artifact/org.hibernate/hibernate-entitymanager  
 compile 'org.hibernate:hibernate-entitymanager:5.2.9.Final'  
 // https://mvnrepository.com/artifact/org.hibernate/hibernate-c3p0  
 compile 'org.hibernate:hibernate-c3p0:5.2.9.Final'  
  
 // MySQL 驱动库  
 // https://mvnrepository.com/artifact/mysql/mysql-connector-java  
 compile group: 'mysql', name: 'mysql-connector-java', version: '6.0.6'  
  
 testCompile group: 'junit', name: 'junit', version: '4.12'  
}

数据库连接：resources->hibernate.cfg.xm

<?xml version='1.0' encoding='utf-8'?>  
<!DOCTYPE hibernate-configuration PUBLIC  
 "-//Hibernate/Hibernate Configuration DTD 3.0//EN"  
 "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">  
  
<hibernate-configuration>  
 <session-factory>  
 <!-- 数据库链接驱动 -->  
 <property name="connection.driver\_class">com.mysql.cj.jdbc.Driver</property>  
 <!--链接地址用户名密码 -->  
 <property name="connection.url">jdbc:mysql://127.0.0.1:3306/DB\_I\_T\_PUSH?serverTimezone=UTC&amp;useUnicode=true&amp;characterEncoding=utf8&amp;useSSL=false</property>  
 <property name="connection.username">root</property>  
 <property name="connection.password">123456</property>  
  
 <!--<property name="connection.url">jdbc:mysql://qiujuer.net:6968/DB\_I\_T\_PUSH?serverTimezone=UTC&amp;useUnicode=true&amp;characterEncoding=utf8&amp;useSSL=false</property>-->  
  
  
 <!-- JDBC 链接池大小 -->  
 <property name="connection.pool\_size">5</property>  
  
 <!-- SQL 方言 -->  
 <property name="dialect">org.hibernate.dialect.MySQL57Dialect</property>  
  
 <!-- Hibernate session 上下文为线程级别 -->  
 <property name="current\_session\_context\_class">thread</property>  
  
 <!-- 配置C3P0缓存链接池 -->  
 <property name="cache.provider\_class">org.hibernate.c3p0.internal.C3P0ConnectionProvider</property>  
 <!--在连接池中可用数据库连接的最小数目-->  
 <property name="c3p0.min\_size">6</property>  
 <!--在连接池中所有数据库连接的最大数目-->  
 <property name="c3p0.max\_size">50</property>  
 <!--设定数据库连接的超时时间-->  
 <!--<property name="c3p0.time\_out">1800</property>-->  
 <property name="c3p0.timeout">100</property>  
 <!--可以被缓存的PreparedStatement的最大数目-->  
 <property name="c3p0.max\_statement">50</property>  
 <!-- 当连接池里面的连接用完的时候，C3P0一下获取的新的连接数 -->  
 <property name="c3p0.acquire\_increment">1</property>  
 <!-- 连接对象因该多长时间被自动校验的时间段，以秒为单位-->  
 <property name="c3p0.idle\_test\_period">100</property> <!-- seconds -->  
 <!--最多可以创建Statements对象的个数. . 就是可以执行SQL语句的对象的个数-->  
 <property name="c3p0.max\_statements">0</property>  
  
 <!-- SQL语句输出 -->  
 <property name="show\_sql">true</property>  
 <property name="format\_sql">true</property>  
  
 <!-- 自动更新数据库的级别 -->  
 <property name="hbm2ddl.auto">update</property>  
 <!--  
 create：表示启动的时候先drop，再create  
 create-drop: 也表示创建，只不过再系统关闭前执行一下drop  
 update: 这个操作启动的时候会去检查schema是否一致，如果不一致会做scheme更新  
 validate: 启动时验证现有schema与你配置的hibernate是否一致，如果不一致就抛出异常，并不做更新  
 -->  
  
  
 <!--<mapping package="net.qiujuer.web.italker.push.bean.db"-->  
 <!--class="net.qiujuer.web.italker.push.bean.db.Group"/>-->  
 <!--<mapping package="net.qiujuer.web.italker.push.bean.db"-->  
 <!--class="net.qiujuer.web.italker.push.bean.db.GroupMember"/>-->  
 <!--<mapping package="net.qiujuer.web.italker.push.bean.db"-->  
 <!--class="net.qiujuer.web.italker.push.bean.db.User"/>-->  
 <!--<mapping package="net.qiujuer.web.italker.push.bean.db"-->  
 <!--class="net.qiujuer.web.italker.push.bean.db.UserFollow"/>-->  
 <!--<mapping package="net.qiujuer.web.italker.push.bean.db"-->  
 <!--class="net.qiujuer.web.italker.push.bean.db.Message"/>-->  
 <!--<mapping package="net.qiujuer.web.italker.push.bean.db"-->  
 <!--class="net.qiujuer.web.italker.push.bean.db.PushHistory"/>-->  
 <!--<mapping package="net.qiujuer.web.italker.push.bean.db"-->  
 <!--class="net.qiujuer.web.italker.push.bean.db.Apply"/>-->  
  
 </session-factory>  
</hibernate-configuration>

服务映射webapp->WEB-INF->web.xml

<?xml version="1.0" encoding="UTF-8"?>  
<web-app>  
  
 <display-name>iTalker</display-name>  
  
 <servlet>  
 <servlet-name>ITalkerApiServlet</servlet-name>  
 <!--容器-->  
 <servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>  
 <init-param>  
 <param-name>jersey.config.server.provider.packages</param-name>  
 <!--映射的包名，用于搜索处理类-->  
 <param-value>net.zhouxu.web.italker.push.service</param-value>  
 </init-param>  
 <init-param>  
 <param-name>javax.ws.rs.Application</param-name>  
 <param-value>net.zhouxu.web.italker.push.Application</param-value>  
 </init-param>  
  
 <load-on-startup>1</load-on-startup>  
 </servlet>  
  
 <!--映射-->  
 <servlet-mapping>  
 <servlet-name>ITalkerApiServlet</servlet-name>  
 <!--访问路径-->  
 <url-pattern>/api/\*</url-pattern>  
 </servlet-mapping>  
</web-app>