

IntelliJ Idea 中 Spring JPA 相关插件使用技巧

我们都知道在Java开发界有两大开发者工具Eclipse和IntelliJ Idea，但是真的吐血推荐大家用IntelliJ Idea，如果没有转的抓紧转，它致力于提升开发者的开发效率，让大家开发代码有飞一样的感觉。

IntelliJ Idea的大概介绍

IntelliJ IDEA 主要用于支持 Java、Scala、Groovy 等语言的开发工具，同时具备支持目前主流的技术和框架，擅长于企业应用、移动应用和 Web 应用的开发。它是 JetBrains 公司旗下的，同时 JetBrains 公司还有如下产品，也都吐血推荐。

- PhpStorm 主要用于开发 PHP
- RubyMine 主要用于开发 Ruby
- PyCharm 主要用于开发 Python
- AppCode 主要用于开发 Objective-C / Swift
- CLion 主要用于开发 C / C++
- WebStorm 主要用于开发 JavaScript、HTML5、CSS3 等前端技术
- 0xDBE 主要用于开发 SQL
- Android Studio 主要用于开发 Android (Google 基于 IntelliJ IDEA 社区版进行迭代所以也姑且算上)

IntelliJ Idea 对于Java来说支持

IntelliJ IDEA 主要功能介绍

语言支持上

安装插件后支持	SQL类	基本JVM
PHP	PostgreSQL	Java
Python	MySQL	Groovy
Ruby	Oracle	
Scala	SQL Server	
Kotlin		

Clojure

语言框架上的支持

支持的框架	额外支持的语言代码提示	支持的容器
Spring MVC	HTML5	Tomcat
Spring boot	CSS3	TomEE
Spring Data	SASS	WebLogic
Spring Cloud	LESS	JBoss
Spring全家桶基本上都能找到相应插件	JavaScript	Jetty
Web Services	CoffeeScript	WebSphere
JSF	Node.js	
Struts	ActionScript	
Hibernate		
Flex		

GitChat

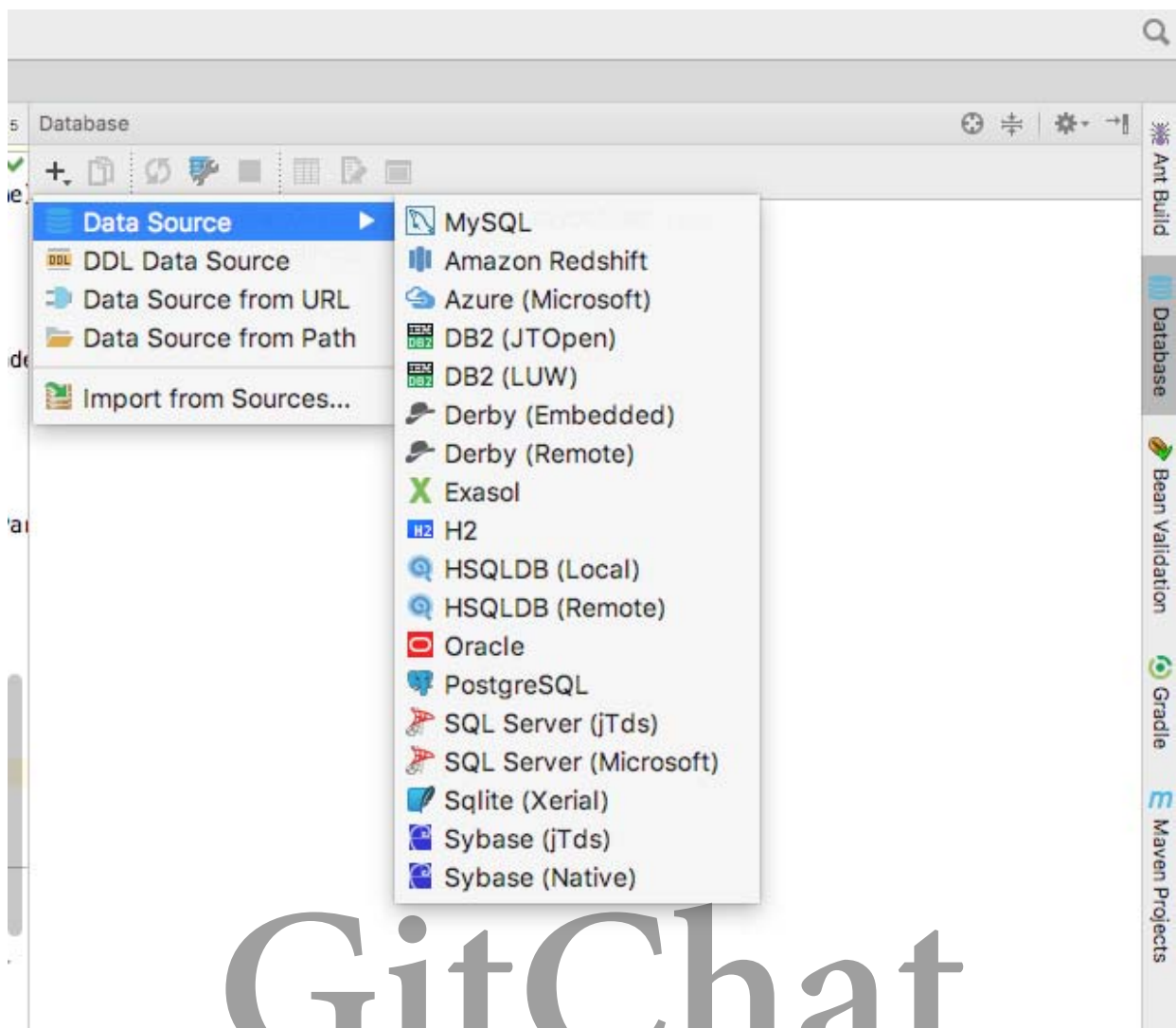
[更多官方介绍](#)

[更多Idea的插件和特性介绍](#)

而我们本章主要介绍一下spring data jpa在Idea中都用到哪些插件。

DataBase (可以代替 N 多数据库客户端)

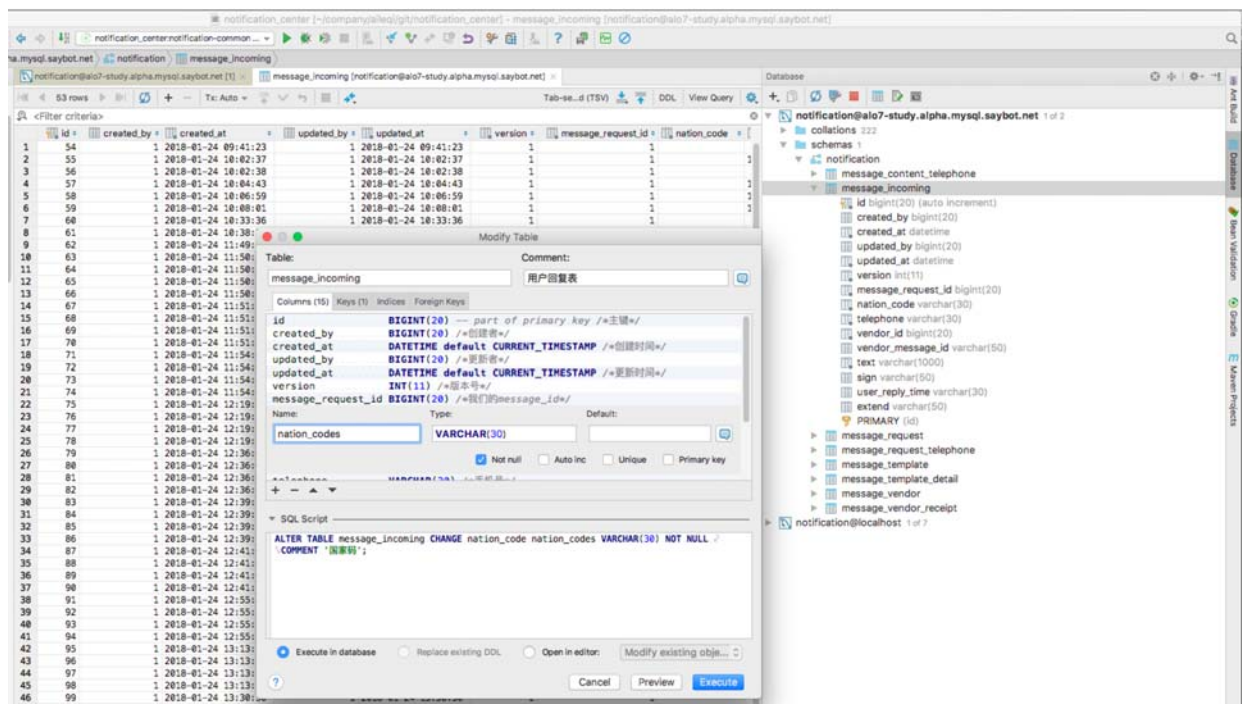
Intellij idea 支持 MySQL、PostgreSQL、Microsoft SQL Server、Microsoft Azure、Oracle、Amazon Redshift、Sybase、DB2、SQLite、HyperSQL、Apache Derby and H2，以及各种内存数据库，只要你有驱动，那就能连接上去。并且跨平台，我们都是用Window、mac、linux得找各种数据库的客户端甚是不方便。



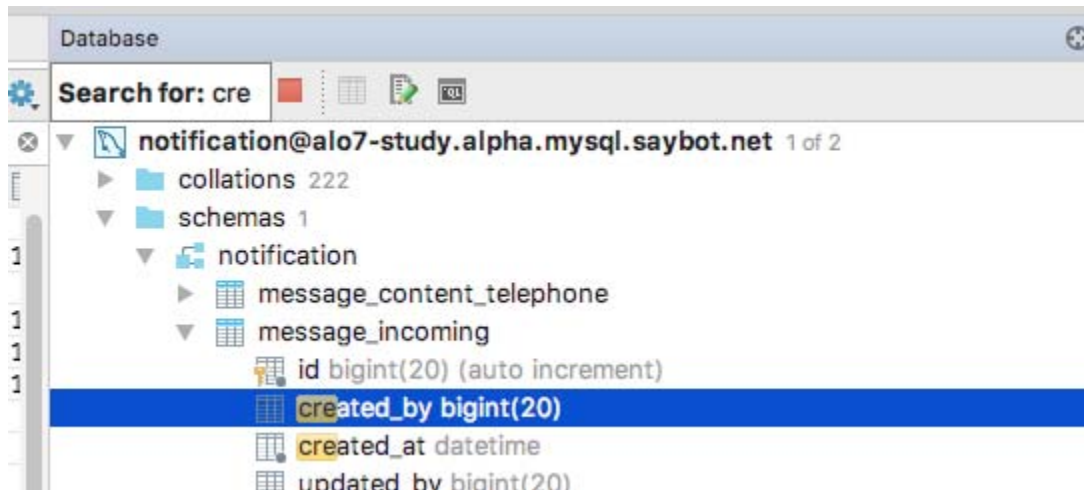
GitChat

在IntelliJ的最右边的Database视图我们可以添加各种DataSource的连接。

方面直观的各种数据库本身的操作

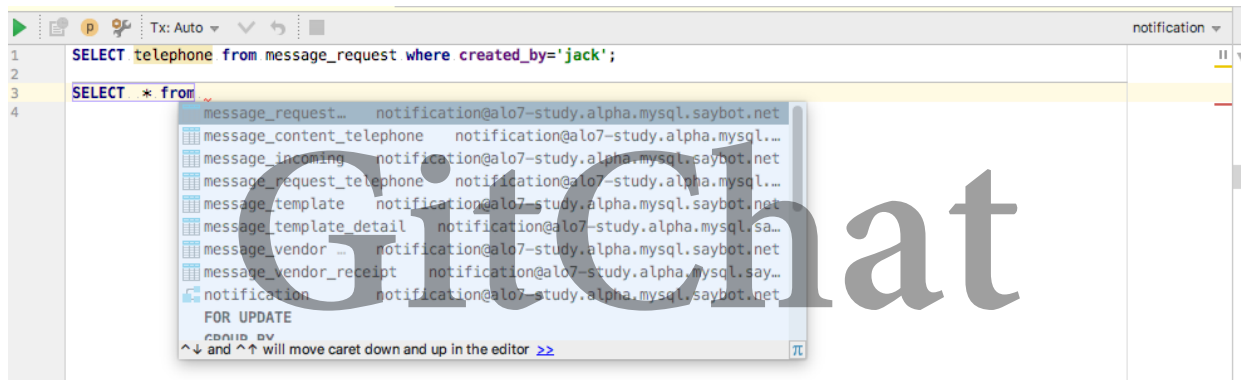


数据库表，字段，数据等直接编辑和修改，并且还会自动帮你生成相应的sql语句。在右边还可以直接进行搜索表，字段等

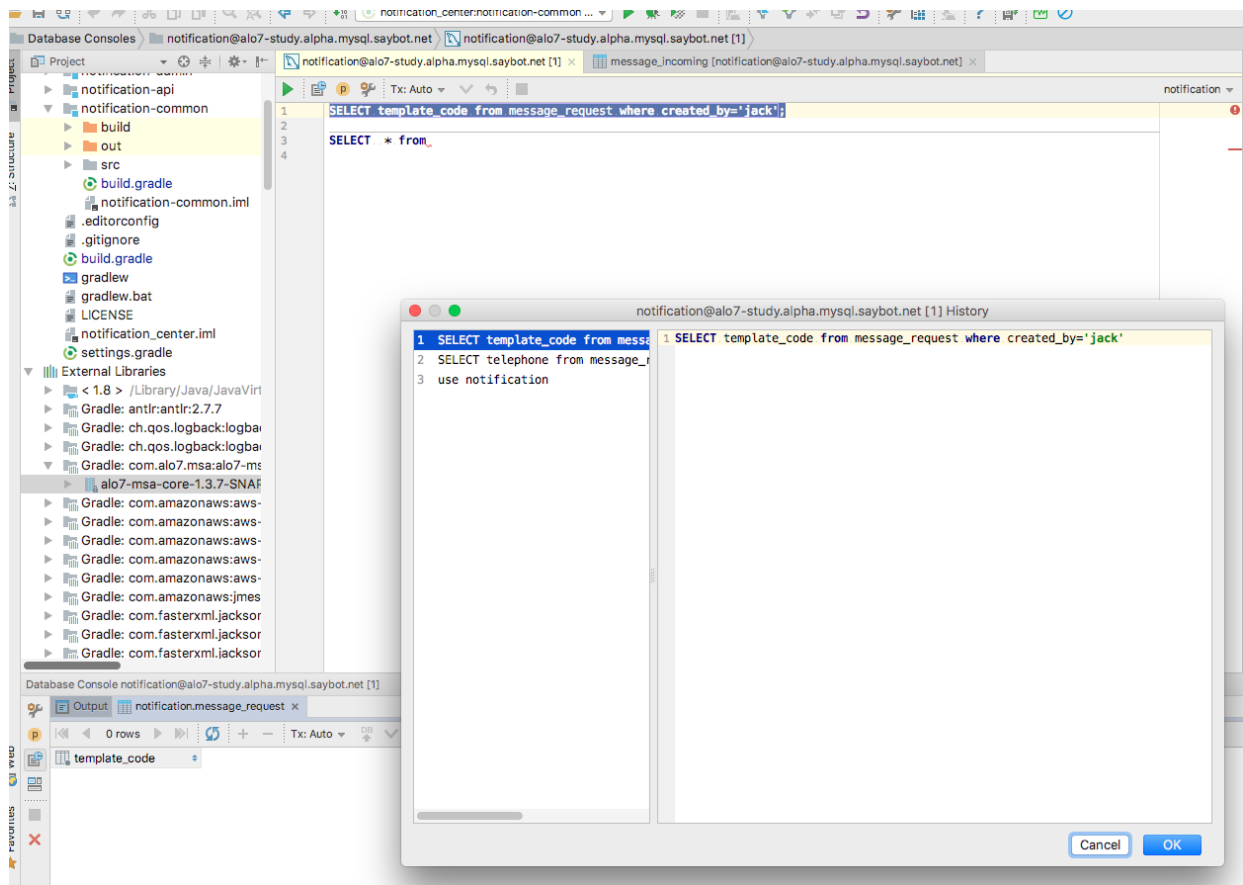


超强和快捷的Query console

sql控制台、完美的提示：

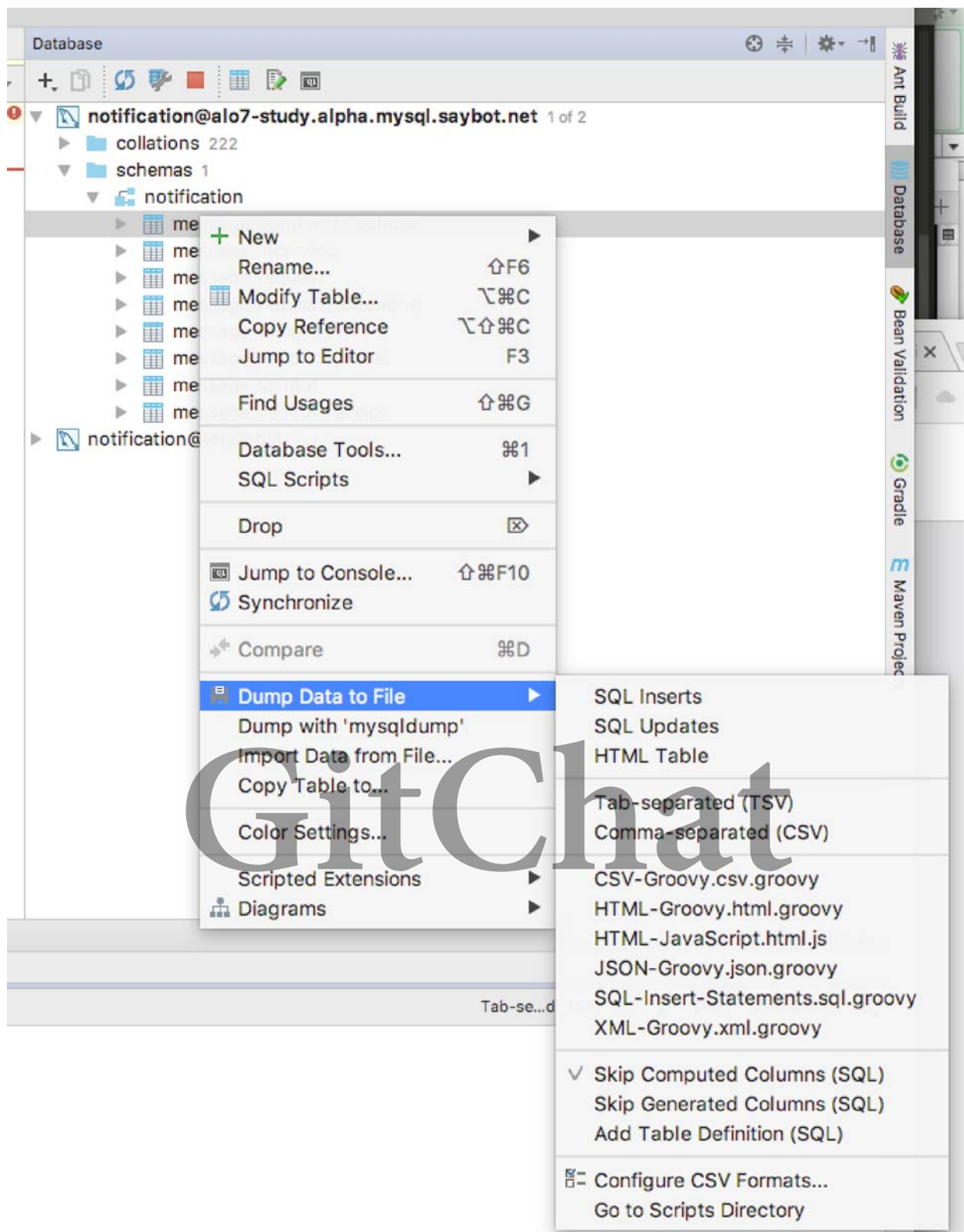


查询历史，sql查询数据结果的呈现



超强的数据导出工具和何种操作

GitChat



- Find Usages可以查出代码里面哪里有用到这张表
- Sql Script里面支持各种ddl的操作
- Dump Data to File：导出功能相当的丰富，各种支持（导出csv、导出text、导出 sql 语句）
- Import Data from File：也是比较喜欢的工具，数据导入

Diagrams直接生成数据模型图，老牛叉了，唯一的缺陷是没有注释，要不就完美了。

message_vendor_receipt	
id	bigint(20)
created_by	bigint(20)
created_at	datetime
updated_by	bigint(20)
updated_at	datetime
version	int(11)
message_request_id	bigint(20)
message_request_detail_id	bigint(20)
message_vendor_id	bigint(20)
vendor_message_id	varchar(50)
error_message	varchar(30)
telephone	varchar(30)
nation_code	varchar(30)
status_code	varchar(30)
result_description	varchar(200)
user_received_time	datetime

message_incoming	
id	bigint(20)
created_by	bigint(20)
created_at	datetime
updated_by	bigint(20)
updated_at	datetime
version	int(11)
message_request_id	bigint(20)
nation_code	varchar(30)
telephone	varchar(30)
vendor_id	bigint(20)
vendor_message_id	varchar(50)
text	varchar(1000)
sign	varchar(50)
user_reply_time	varchar(30)
extend	varchar(50)

message_template_detail	
id	bigint(20)
created_by	bigint(20)
created_at	datetime
updated_by	bigint(20)
updated_at	datetime
version	int(11)
language	varchar(30)
message_template_id	bigint(20)
message_vendor_id	bigint(20)
vendor_template_id	varchar(50)
vendor_template_txt	varchar(1000)
vendor_template_status	varchar(30)
vendor_template_type	varchar(30)
vendor_template_sign	varchar(50)

message_request_telephone	
id	bigint(20)
created_by	bigint(20)
created_at	datetime
updated_by	bigint(20)
updated_at	datetime
version	int(11)
message_request_id	bigint(20)
nation_code	varchar(30)
telephone	varchar(30)
telephone_original	varchar(30)
vendor_message_id	varchar(50)
vendor_result_msg	varchar(1000)
vendor_result_fee	int(11)
vendor_result_code	int(11)

message_request	
id	bigint(20)
created_by	bigint(20)
created_at	datetime
updated_by	bigint(20)
updated_at	datetime
version	int(11)
uuid	varchar(32)
content	varchar(1000)
status	varchar(30)
message_vendor_id	bigint(20)
template_code	char(20)
template_detail_id	bigint(20)
template_params	varchar(30)
sign	varchar(30)

message_vendor	
id	bigint(20)
created_by	bigint(20)
created_at	datetime
updated_by	bigint(20)
updated_at	datetime
version	int(11)
name	varchar(30)
sign	varchar(50)
account	varchar(30)
password	varchar(30)
status	varchar(30)
vendor_code	varchar(50)

message_template	
id	bigint(20)
created_by	bigint(20)
created_at	datetime
updated_by	bigint(20)
updated_at	datetime
version	int(11)
title	char(1)
media_type	varchar(30)
template_type	varchar(30)
template_code	int(11)

message_content_telephone	
id	bigint(20)
message_id	bigint(20)
nation_code	varchar(20)
telephone	varchar(30)
telephone_original	varchar(30)

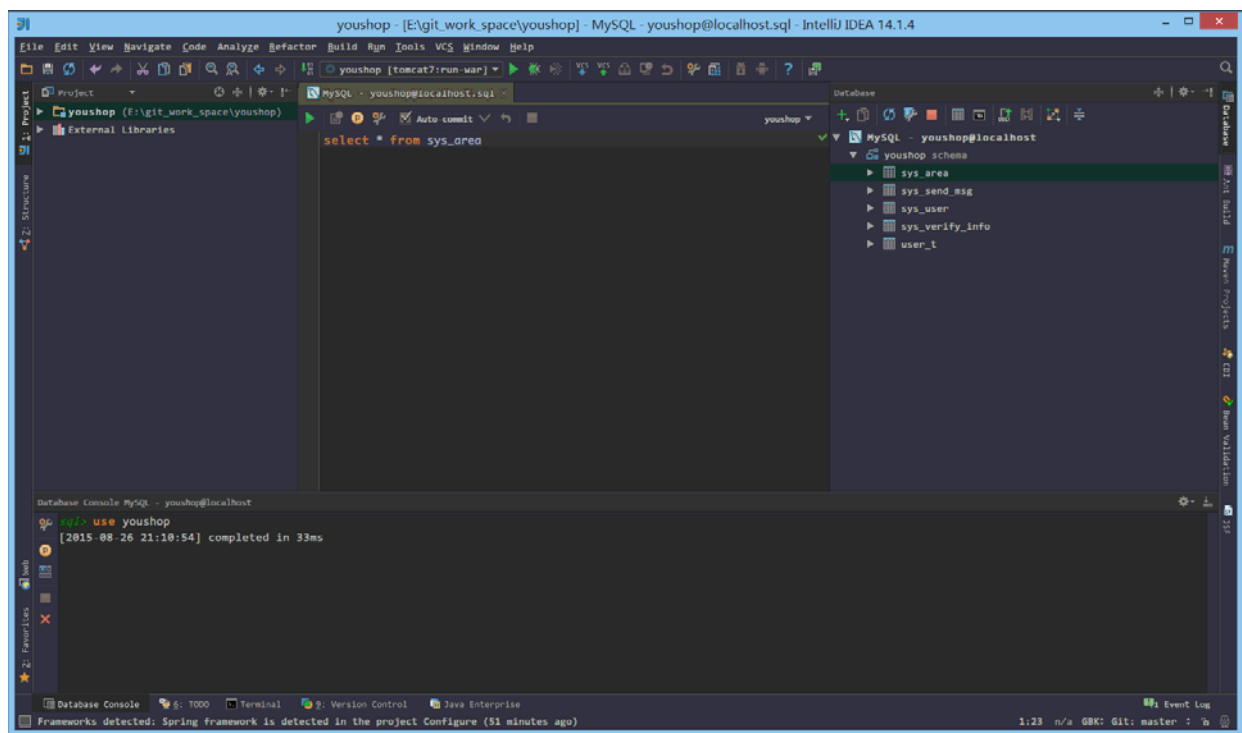
files

GitChat

支持查看执行计划和sql的参数。

The screenshot shows the Database IDE interface. The main window displays a SQL query: `SELECT * from message_request r where r.template_code=code;`. Below the query, the 'Parameters' dialog is open, showing the parameter `code` with the value `r.template_code=`. The 'Database Console' at the bottom shows the execution plan for the query, which is a simple table scan on the `message_request` table.

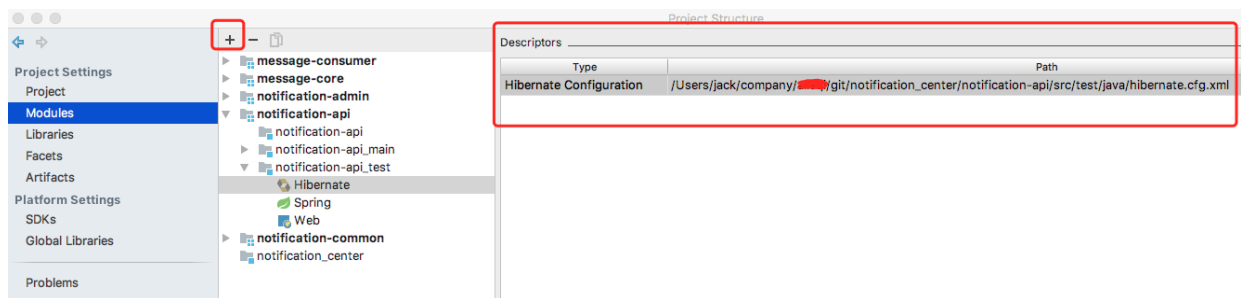
至此Database的基本插件使用介绍完毕，剩下的大家慢慢体会它的便利性吧，个人觉得基本上你的电脑上不需要装任何其它数据库的client了。
一个牛逼的Gif动画如下：大家要等待一下网络速度。



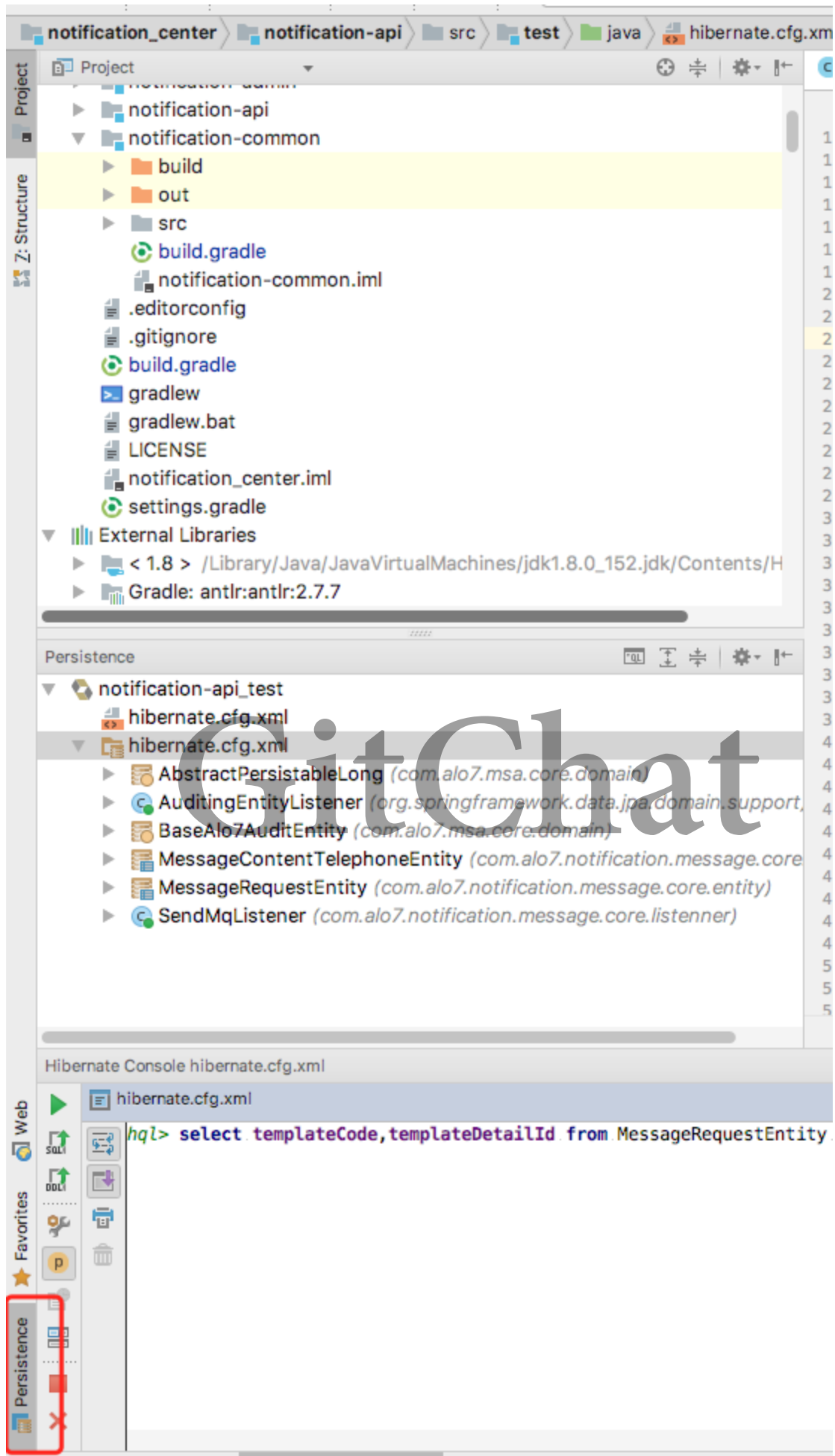
Persistence及其JPA相关的插件介绍

IntelliJ IDEA目前还没有针对Spring Data Jpa的独立的完整插件。但是由于Spring Data Jpa也是满足Java的Persistence API的标准的，所有工具里面的Persistence也做很好的支持。

1：开启persistence界面



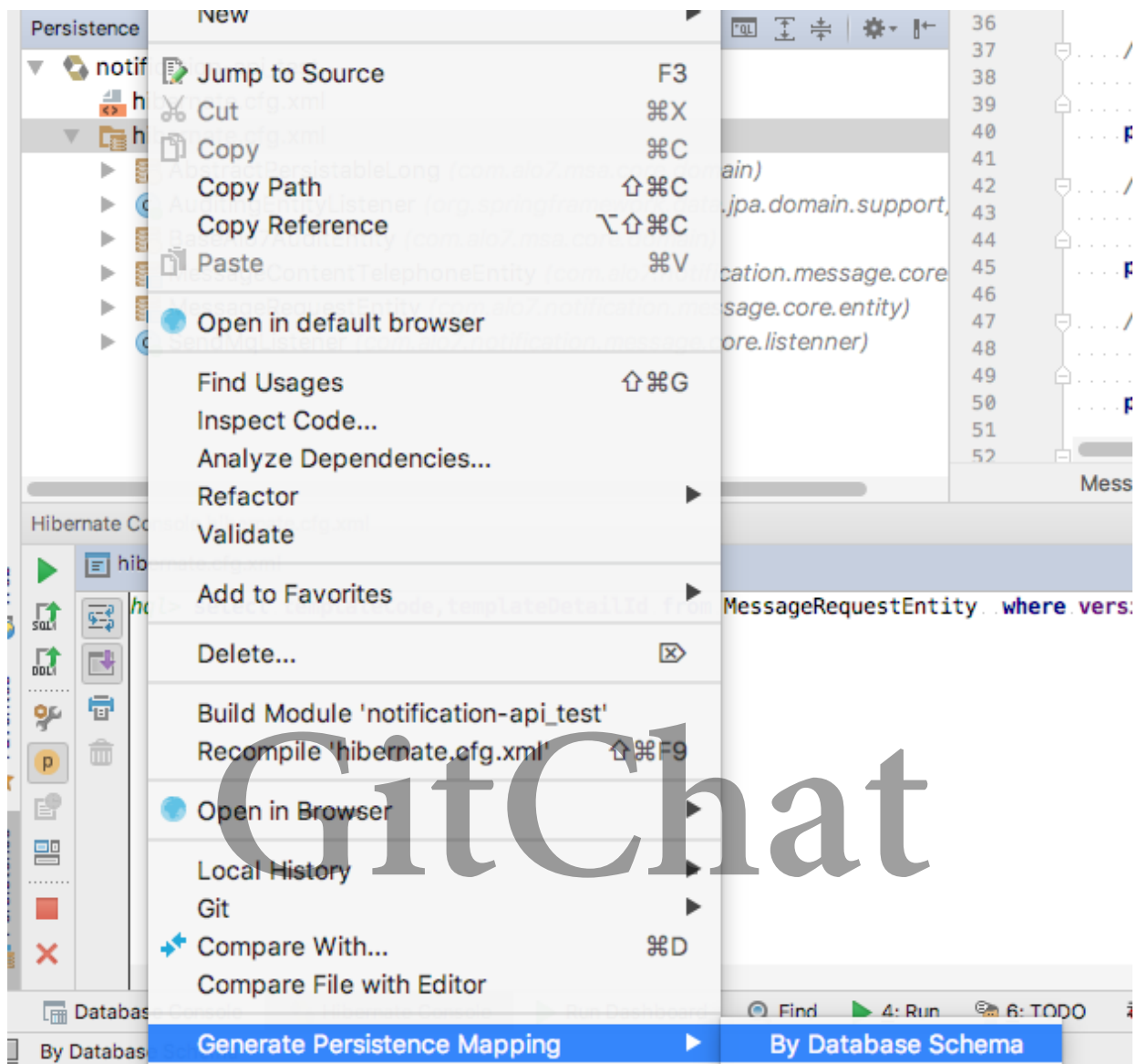
需要打开Project Setting的设置界面Modules里面添加一个Hibernate模块，并且创建一个Hibernate.cfg.xml文件（建议放在test目录，这样子不至于影响你的生产环境）。直接通过图上的+就可以一路默认搞定。



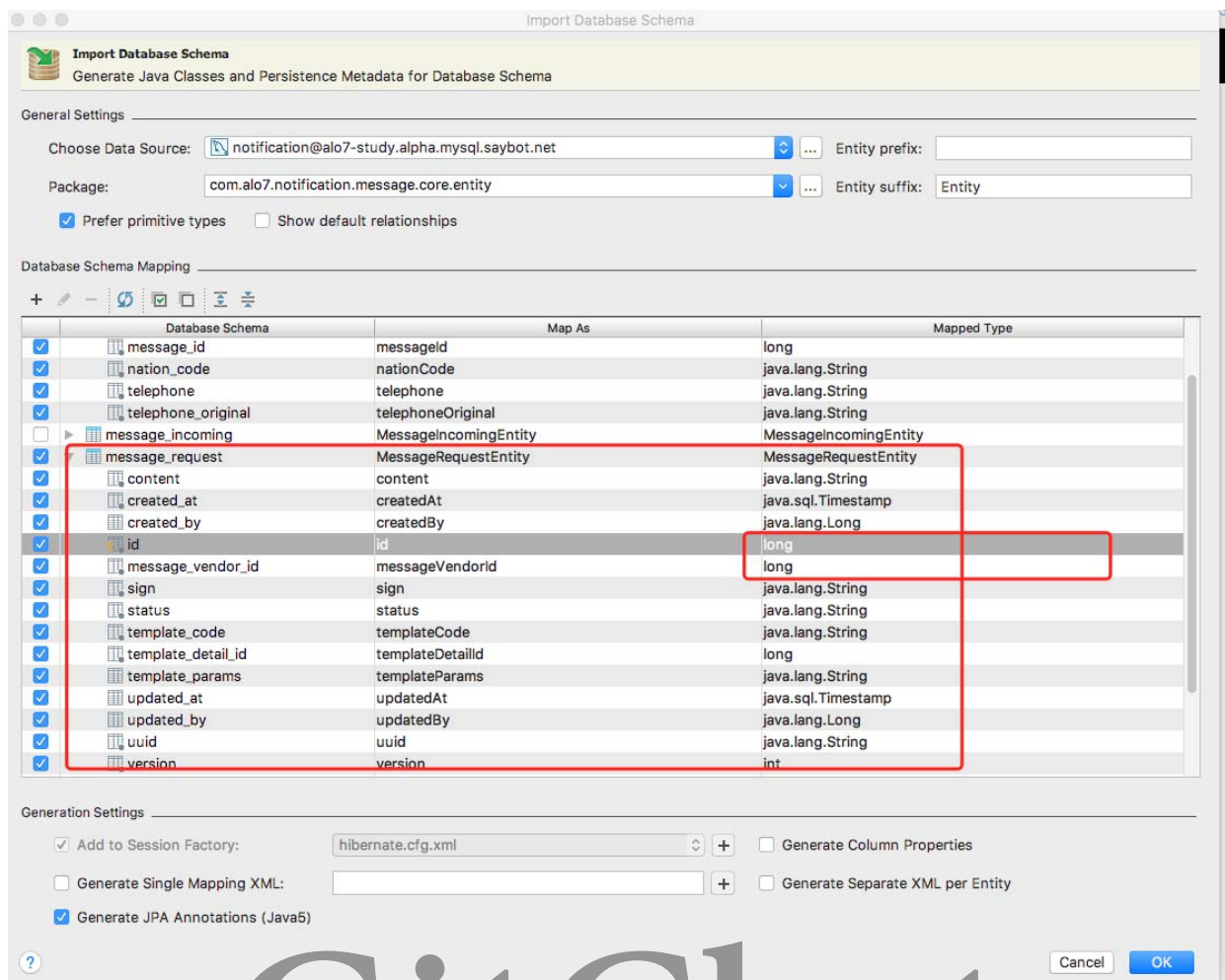
这时候打开IntelliJ Idea的右下角，就会多一个插件视图，打开就会如上图所示。

2 : 利用Persistence生成Entity

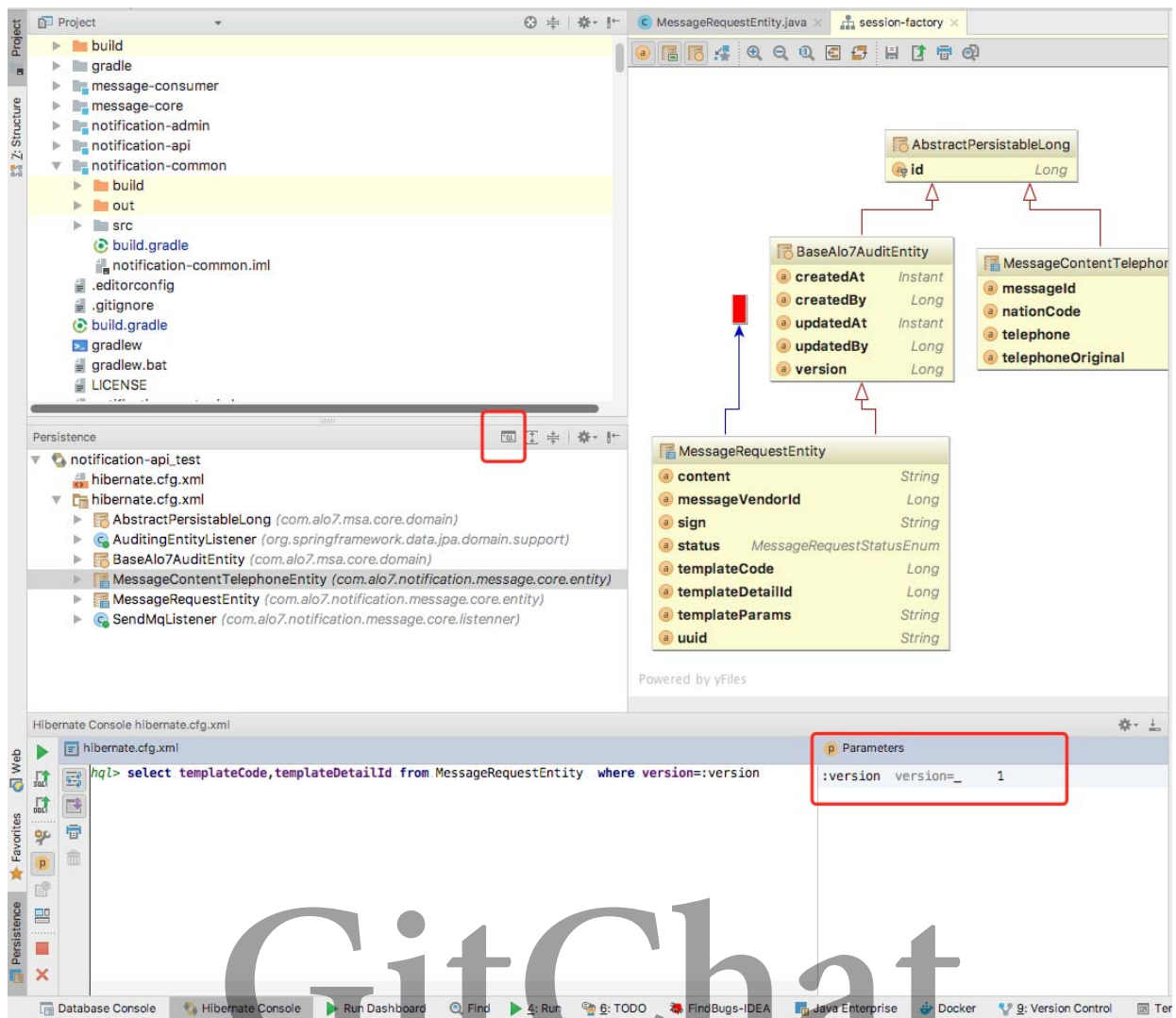
1) 选中上图的Persistence里面的Hibernate.xml，右键打开生成Entity的界面



2) 如下界面选择要生成的实体的表，和字段，非常详细，并且后面的Map as都可以编辑，太人性化了。包路径，实体后缀都可以改。



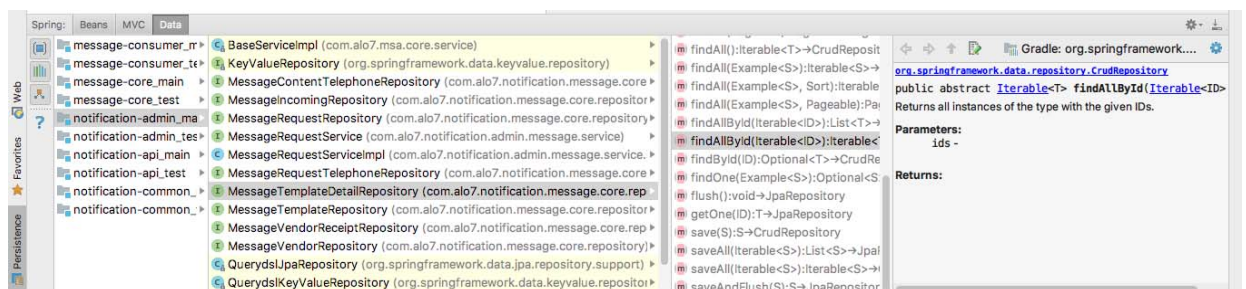
3) Persistence的ERP图和Hql控制台，相当帅气了



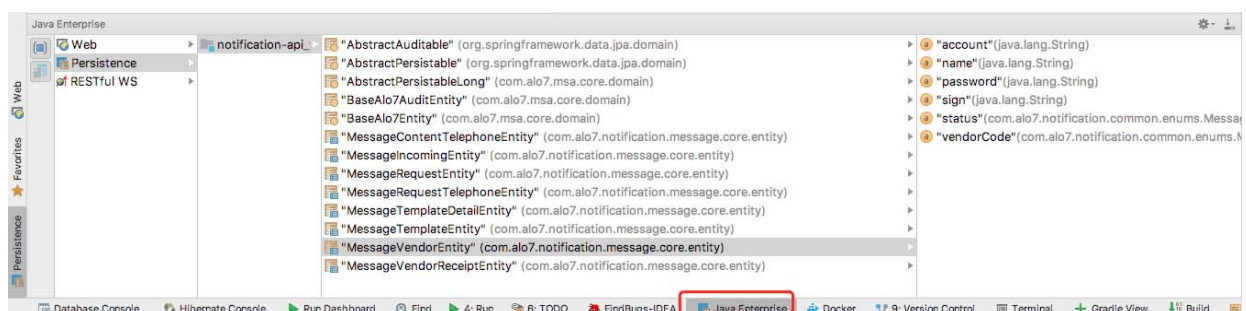
通过这个视图我们可以充分验证我们的JPQL及其里面的参数。

4) 整体项目Spring Data 的预览情况：

查看项目里面的所有的Repository:

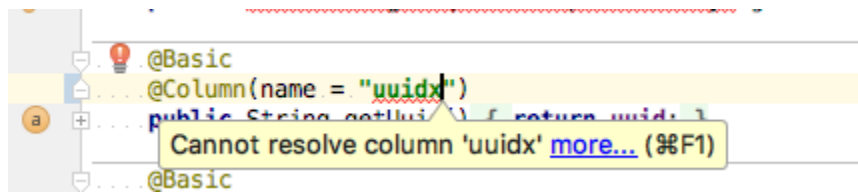


查看项目里面的所有Entity:



5) Spring Data Jpa使用的时候智能提示：

当字段和表里面的字段不一致的时候的提示：

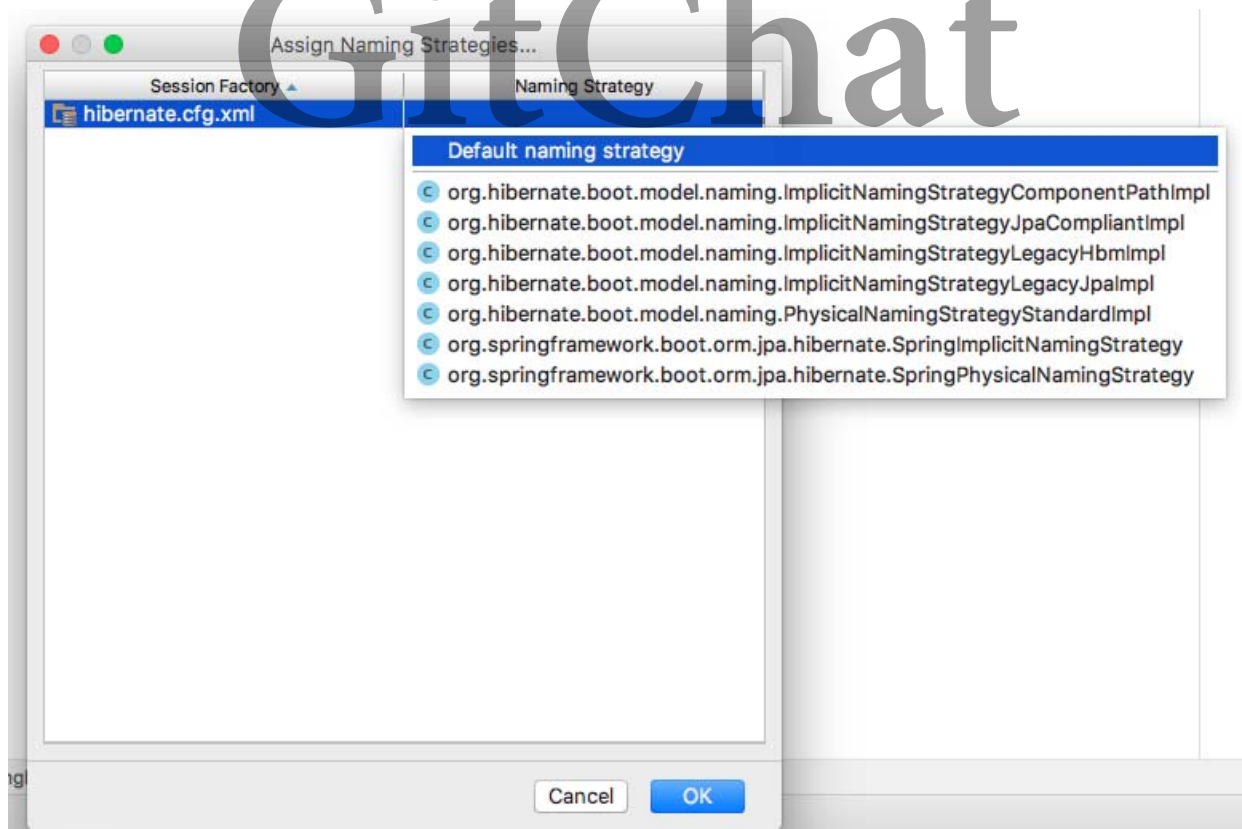


其实，当我们写@ManyToOne、@ManyToMany 字段填写不对的时候提示也挺好的。

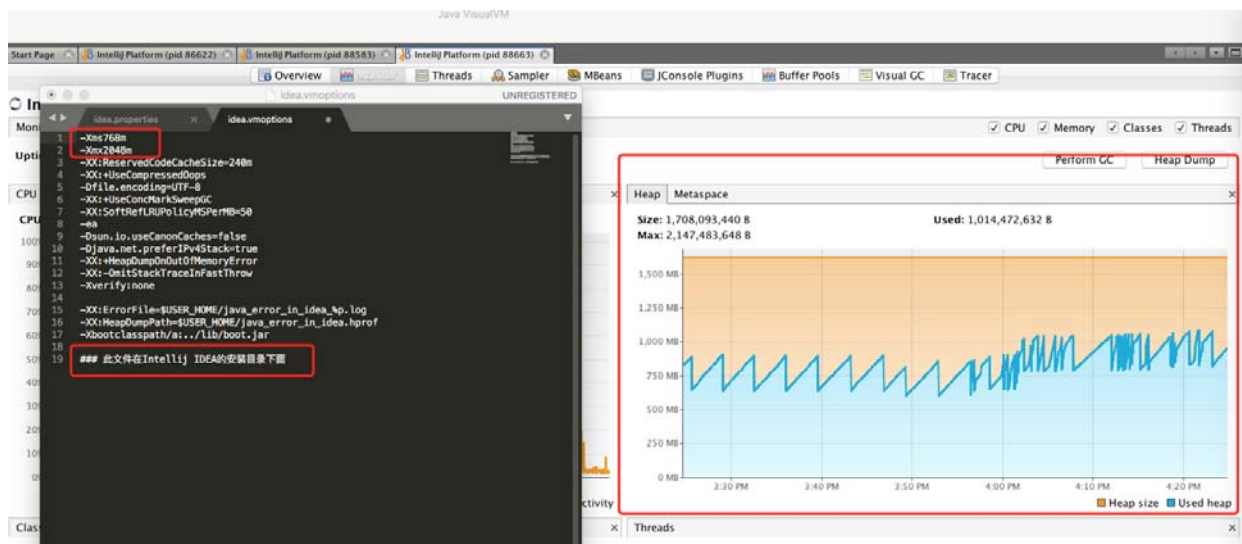
当我们在书写Repository的时候也会根据我们的Entity和database有很好的提示：



Naming Strategy的提示：



Intelli idea 如果使用慢调优方法



我们利用Jvm调优的方法，调一下它的堆的大小，让我们的工具飞起来。有的时候我会同时开7-10个项目。

作者介绍此两个插件的目的：

是希望大家以点带面，全面的熟悉自己的工作工具，因为你熟悉的越多工作效率，越高，越显示出来你的专业能力的。你了解的越多越会发现这个是神工具，你会被的魅力深深吸引。特别是对于Spring家族的东西的插件，大家好好摸索一些，这样有助于大家认识spring和你的项目。

交流：欢迎大家一起留言，把自己碰到的Java数据结构问题一起讨论。

作者：张振华.Jack。

QQ交流群一：240619787

QQ交流群二：559701472

如果大家觉得老师的辛苦值得，欢迎给5星好评。