商城项目介绍

## 5 idea的快捷键

### 5.1 常用快捷键

智能补全 引包 alt+Enter

由方法自动生成返回值变量 ctrl+alt+v

跳到方法的实现类 ctrl+alt+b

查看方法参数 ctrl+p

查看方法文档 ctrl+q

复制行 ctrl+D

删除行 ctrl+Y

跳转到上一个/下一个位置 ctrl+alt+左右

大小写切换 ctrl+shift+u

### 5.2 Debug:

F8 执行下一行 (相当于eclipse的 F6)

F7 跳入内部 (相当于eclipse的 F5)

F9 继续执行 (相当于eclipse的 F8)

热部署 ctrl+shift+F9 （仅debug模式）

### 5.3 搜索

全文搜索文本 ctrl+shift+f

全文替换文本 ctrl+shift+r

搜索类 ctrl+n

任何地方搜索 双击shift

### 5.4 快速录入

查看快速录入列表 ctrl+j

foreach iter

普通for循环 fori

循环数组 itar

迭代器遍历 itco

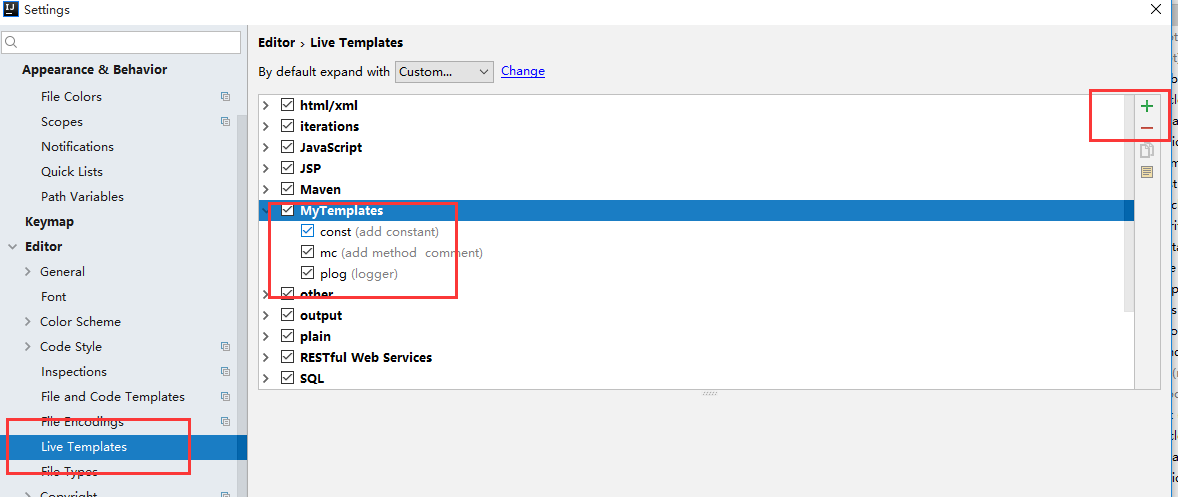
psvm 主函数

pfs 常量

生成代码块： try/ if / for/ while/ synchronized

ctrl+alt+t

## 6 手工加入快捷键模板



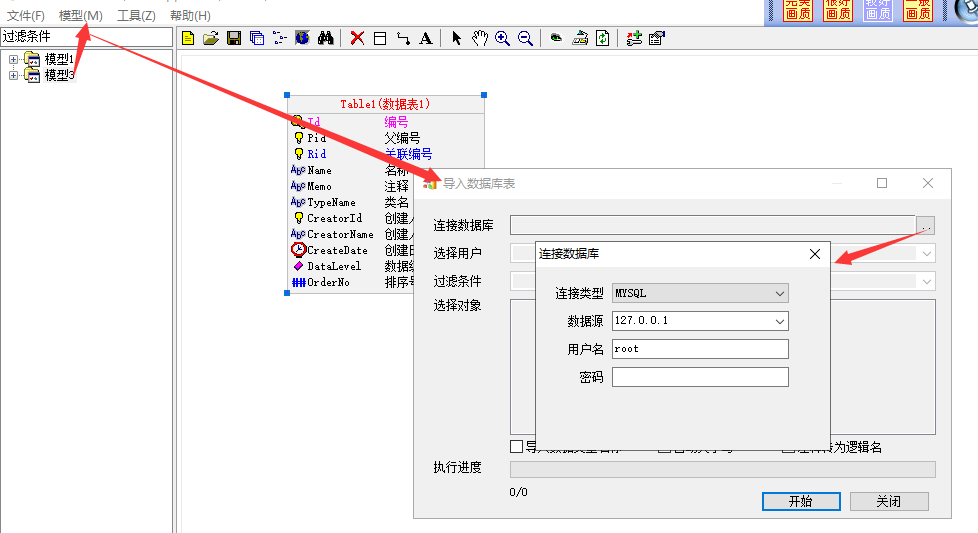
# 1 idea的介绍

|  |
| --- |
| 1 安装和破解idea  进入ide主页面，help-register-license server,然后输入 http://idea.iteblog.com/key.php  2 进入idea(每次新建一个project)  Idea：project->module  Eclipse：workspace->project  Maven仓库      设置jdk环境      3 新建springboot测试    4 刷新maven的idea    5 从git上创建一个工程，下载到本地      6 用git创建一个新的gmall工程    7 将git的gmall工程clone到本地    8 idea的配置文件一定不能提交git    9 工程提交到git      10 提交并且合并仓库 |

# 2 数据结构和设计的介绍

## 1 用ezdml设计数据表然后导出到mysql数据库中

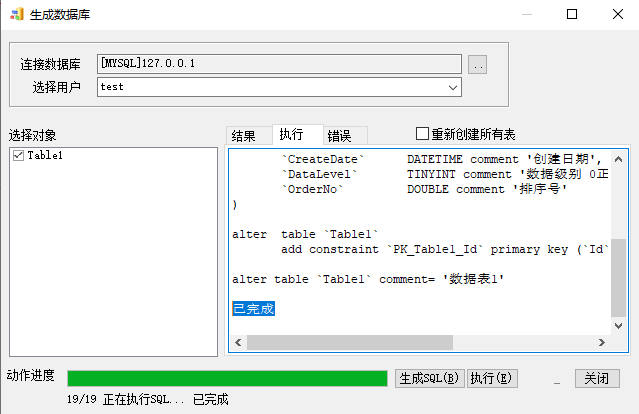
点击生成模型



## 2 选择库



## 3 点击开始生成



## 4 从数据库中导入到ezdml

# 3 新建一个user项目(将来用来提供用户服务)

|  |  |
| --- | --- |
| 1 新建module，选择springweb、mysql驱动、jdbc、mybatis  springboot配置文件   |  | | --- | | *# 服务端口* **server.port**=**8080** *# jdbc* **spring.datasource.password**=**123456 spring.datasource.username**=**root spring.datasource.url**=**jdbc:mysql://localhost:3306** *# mybtais配置* **mybatis.mapper-locations**=**classpath:mapper/\*Mapper.xml mybatis.configuration.map-underscore-to-camel-case**=**true** |   2 写controller、service和mapper  自动导包    新建java类型    Idea @Autowired误报    3 写bean映射类，映射数据库ums\_user\_member表  Alt + inster    4 写具体功能（member表和member\_receive\_address增删改查）  Ums\_member 的主键 对应 ums\_member\_receive\_address 外键member\_id  通用mapper的整合 |

# 4 配置域名

|  |
| --- |
| 1 配置本机的dns  C:\Windows\System32\drivers\etc\hosts  127.0.0.1 localhost user.gmall.com cart.gmall.com manage.gmall.com [www.gmall.com](http://www.gmall.com) …  可以直接访问<http://user.gmall.com:8080/index> 相当于访问了<http://localhost:8080/index>  2 可以通过nginx去代理端口号  upstream user.gmall.com{  server 127.0.0.1:8080;  }  server {  listen 80;  server\_name user.gmall.com;  location / {  proxy\_pass http://user.gmall.com;  proxy\_set\_header X-forwarded-for $proxy\_add\_x\_forwarded\_for;  }  } |

# 5 idea配置mysql数据库插件

|  |
| --- |
|  |

# 6 通用mapper的整合(可以将单表的增删改查操作省去)

|  |
| --- |
| 1 导入pom依赖  *<!-- 通用mapper -->* <**dependency**>  <**groupId**>tk.mybatis</**groupId**>  <**artifactId**>mapper-spring-boot-starter</**artifactId**>  <**version**>1.2.3</**version**>  <**exclusions**>  <**exclusion**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-jdbc</**artifactId**>  </**exclusion**>  </**exclusions**> </**dependency**>  2 配置mapper，继承通用mapper  **public interface** UserMapper **extends** Mapper<UmsMember>  3 配置通用mapper的主键和主键返回策略  @Id @GeneratedValue(strategy = GenerationType.***IDENTITY***)  4 配置启动类扫描器MapperScan，使用通用mapper的tk……..MapperScan  tk.mybatis.spring.annotation.MapperScan  5 f8逐步进入，f9跳过断点 |

# 7 根据member\_id查询用户收获地址集合

|  |
| --- |
| 1 通过用mapper的条件查询，通过**memberId**查询用户的收获地址信息集合  UmsMemberReceiveAddress umsMemberReceiveAddress = **new** UmsMemberReceiveAddress(); umsMemberReceiveAddress.setMemberId(**memberId**);  **umsMemberReceiveAddressMapper**.**select**(umsMemberReceiveAddress);  2 如果想查查看执行的sql，可以将日志级别改为debug，打印sql语句  *# 日志级别* **logging.level.root**=**debug** |

# 8 作业

|  |
| --- |
| 在gmall-user项目中，完成ums\_member和ums\_member\_receive\_address两张表的增删改查 |