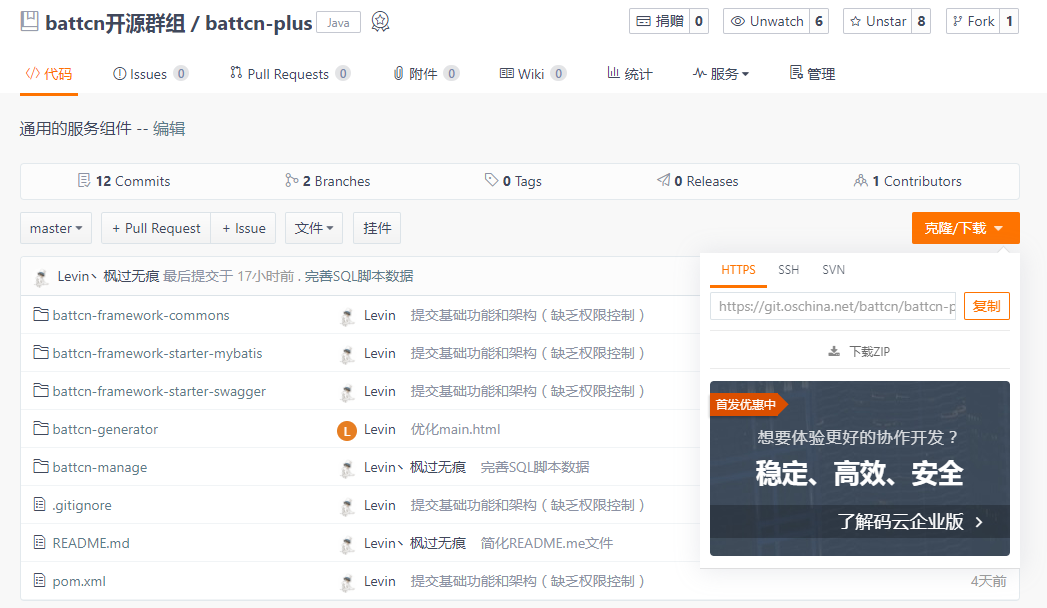
# **Battcn-plus手册**

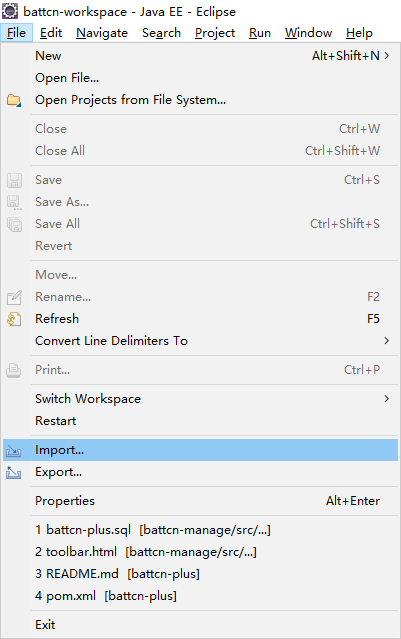
## 一：获取代码到Eclipse



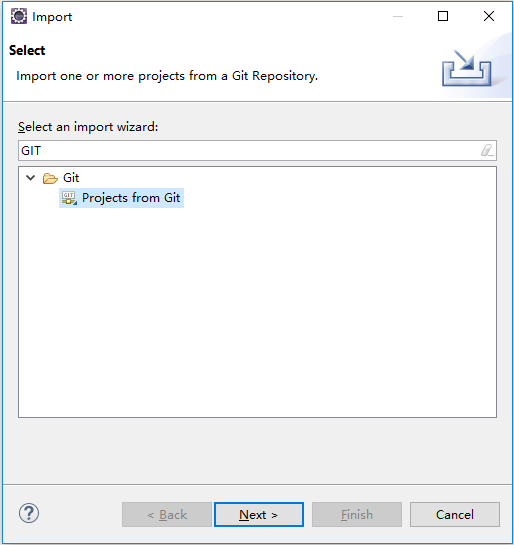
### 1）前往GIT（<https://git.oschina.net/battcn/battcn-plus>）获取代码，可以下载Zip包也可以直接GIT或者SVN剪切代码，此处以GIT的Check方式演示

### 2）打开我们的开发工具（此处以Eclipse演示IDEA自行百度即可）

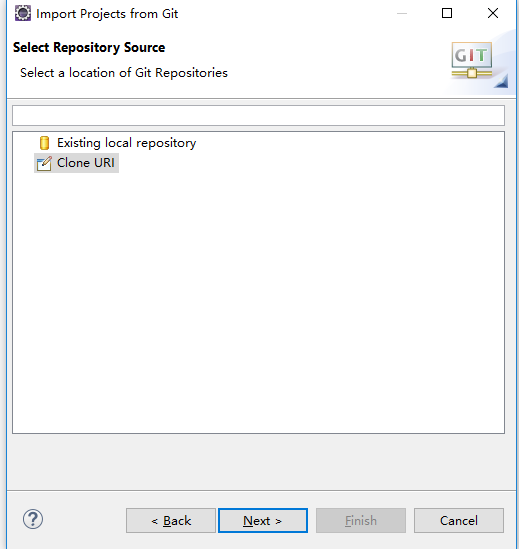
#### 2.1）选择File->Import



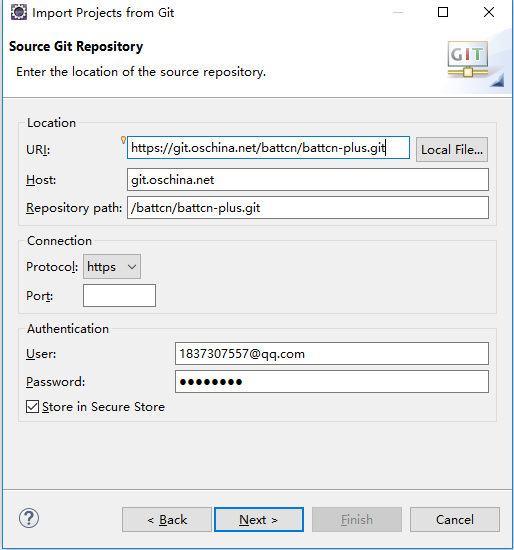
#### 2.2）搜索GIT-Next



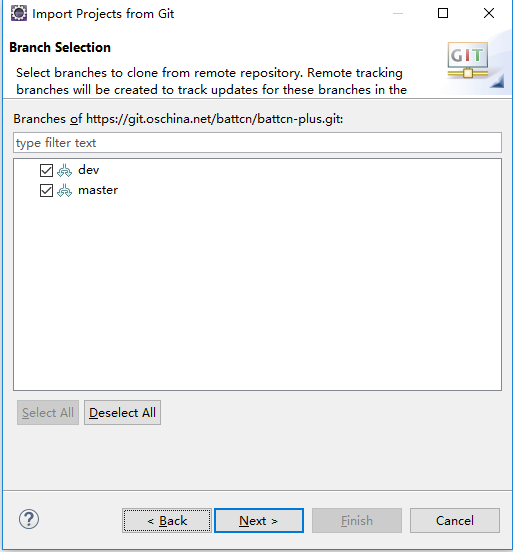
#### 2.3）点击Clone URI->Next



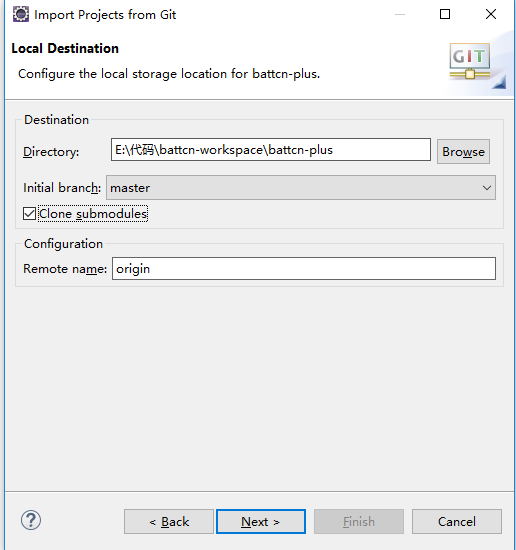
#### 2.4）拷贝<https://git.oschina.net/battcn/battcn-plus.git> 地址，Location会自动匹配上，Authentication填写你们的oschina(码云)账号即可，然后点击Next



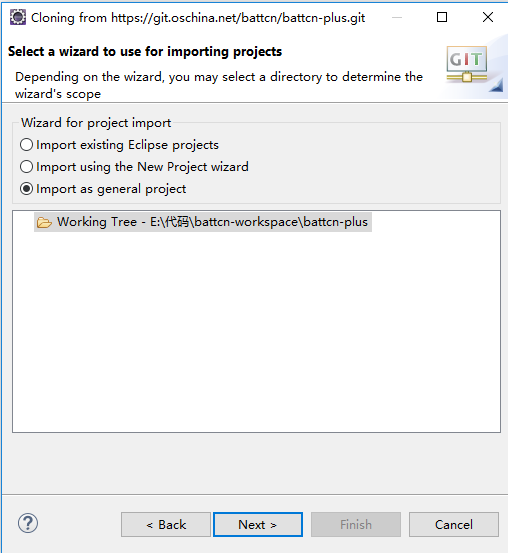
#### 2.5) 分支选择，俩个分支代码同步，所以可以全部剪切，也可以只剪切master的



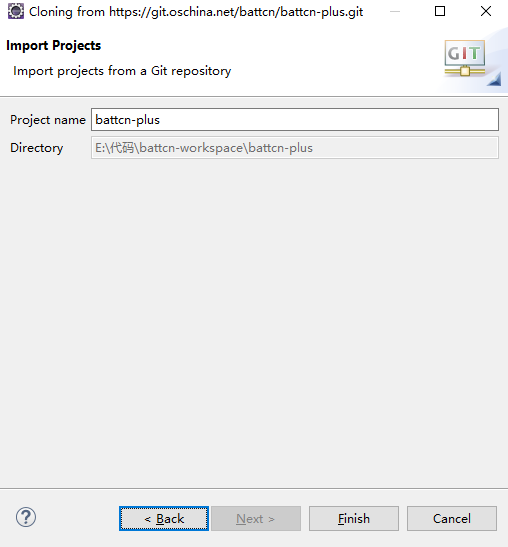
#### 2.6）本地仓库创建，点击Browse选择你所喜欢的路径地址，然后Next

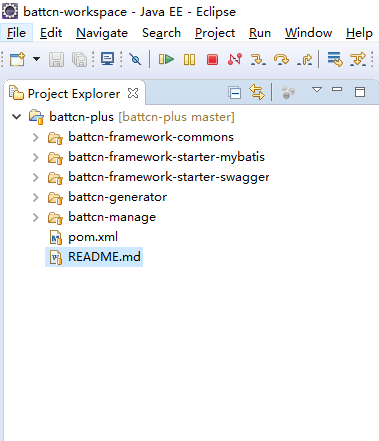


#### 2.7）选择项目并且导入，选择Import as general project，然后Next



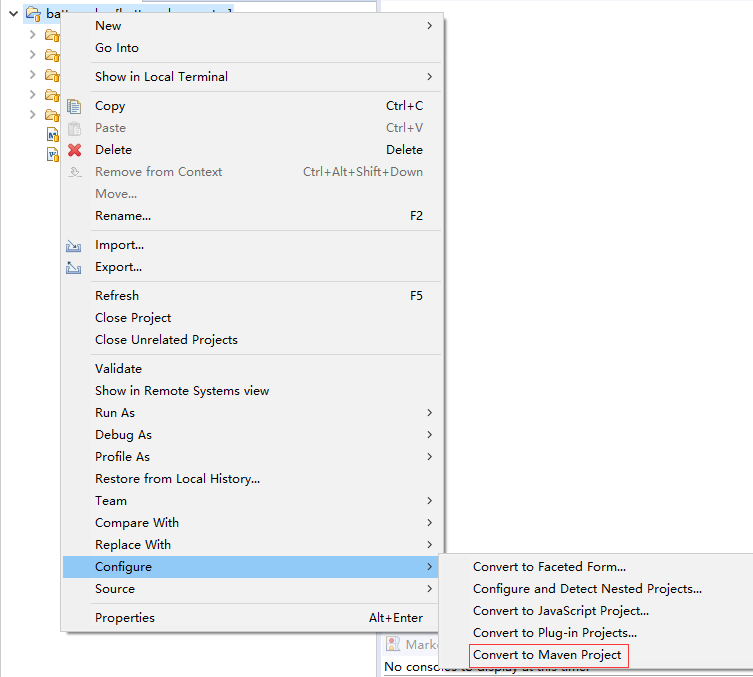
#### 2.8) 最后一步，导入项目，点击Finish



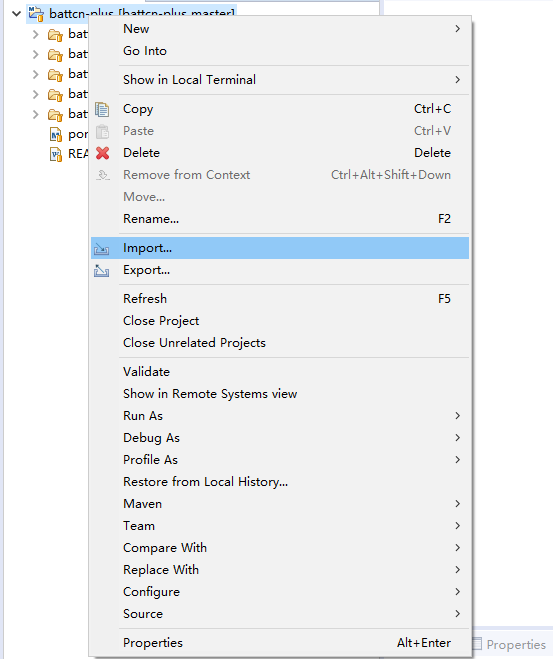


### 3) 转换成Maven工程

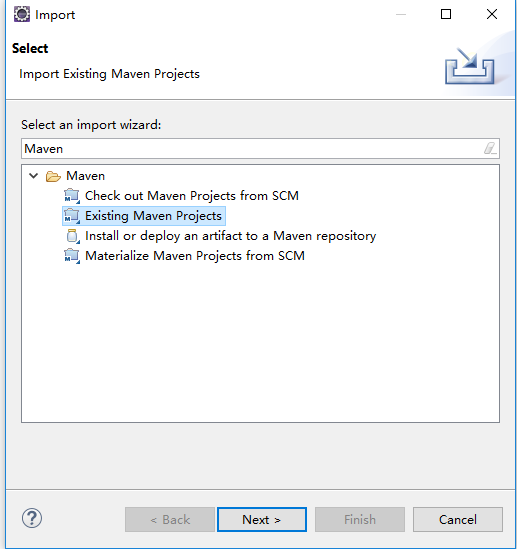
右键项目 点击 Configure->Convert to Maven Project，如图所示



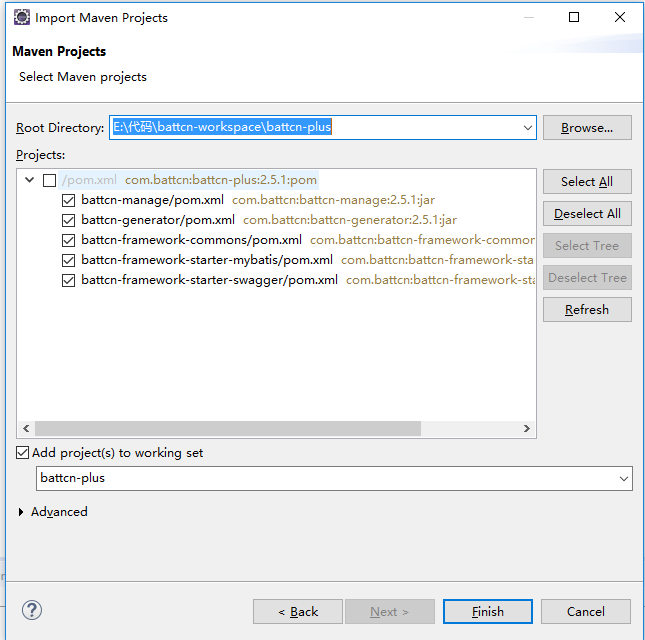
右键battcn-plus 选择Import



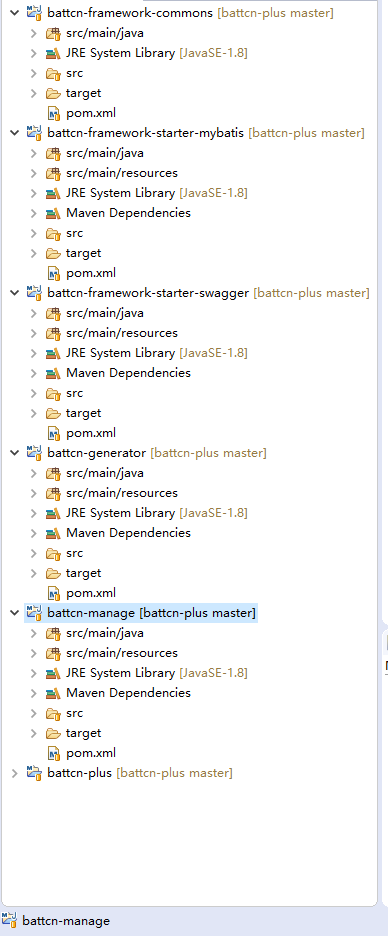
选择Existing Maven Projects 点击Next



点击Finish



到此，就是我们所熟悉的Maven工程了，图文并茂，新手看两遍即可学会如何从GIT上剪切代码，导入Maven工程，项目转换等 是不是很简单



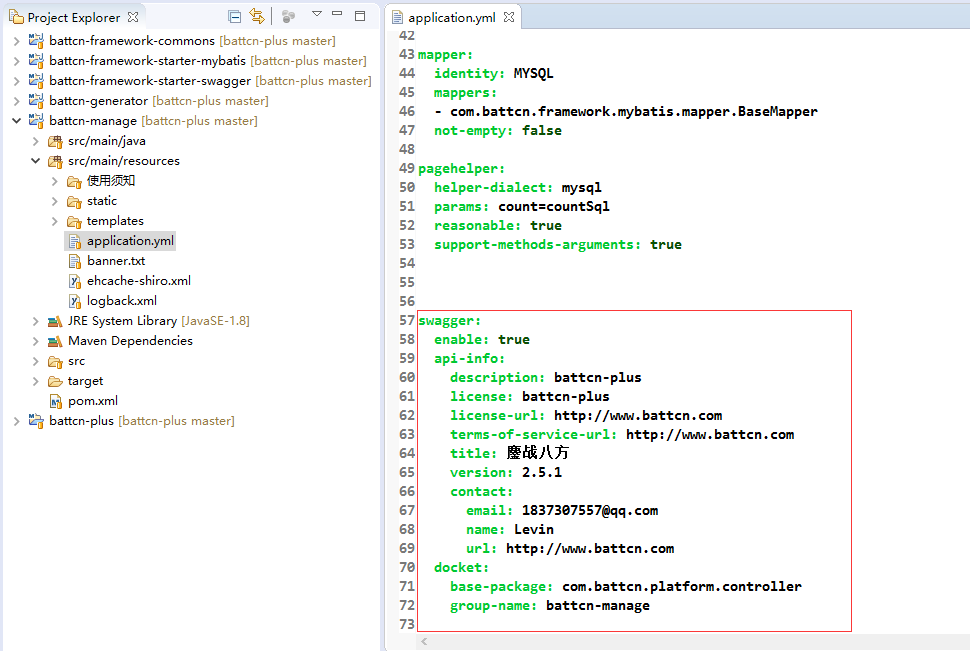
## 二：项目介绍及开发手册

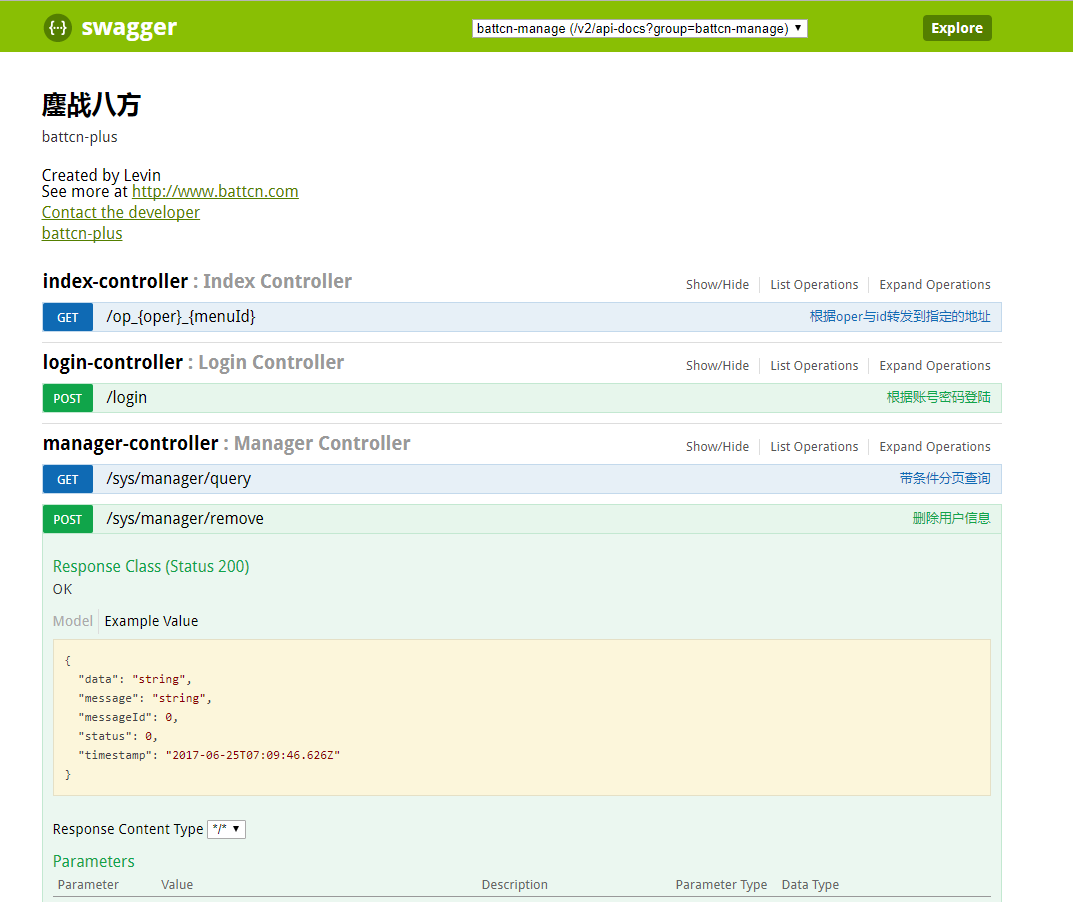
### 1）.项目介绍

#### 1.1）battcn-plus：根项目，父工程

battcn-manage：采用SpringBoot、Shiro、Thymeleaf、Bootstrap、Bootstrap Table、jQuery、Mybatis、PageHelper、OrderByHelper、Tk.mapper、Swagger 搭建的WEB系统、含颗粒度权限的实现，包含了用户、角色、操作、菜单、日志采集

battcn-generator：代码生成器，只适合使用了tk.mapper的项目或者本项目

battcn-framework-starter-swagger：swagger的自动装配工程，依赖该JAR后只需要简单的配置即可使用swagger，简化一个工程太多无需关注的代码

swagger-ui：http://localhost:8082/swagger-ui.html

battcn-framework-starter-mybatis：内置BaseMapper，BaseService、BaseServiceImpl、DataGrild（分页与排序对象）以及Ibatis的template方式操作DAO（不推荐）

battcn-framework-commons：工具类和通用异常BattcnException、BttcnLog收集日志的注解

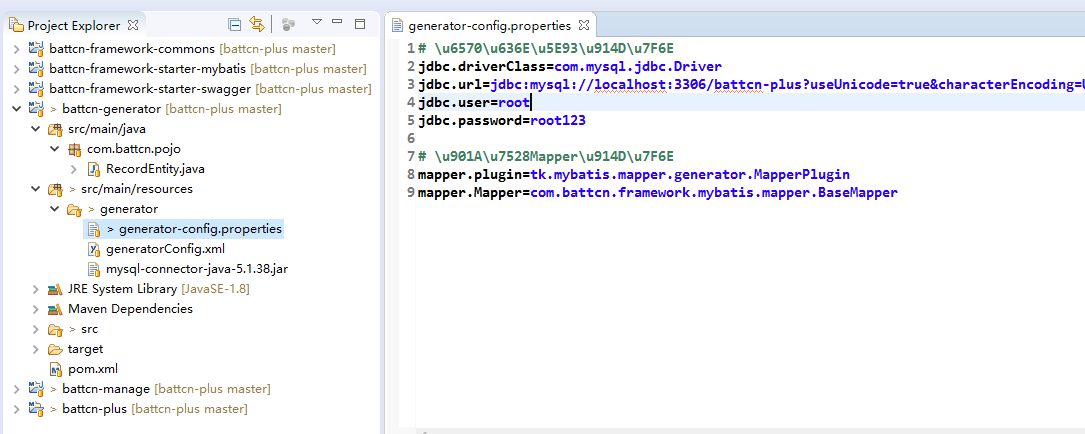
### 2）编码（编写一个简单的Hello Battcn）

#### 2.1）创建一张t\_hello表

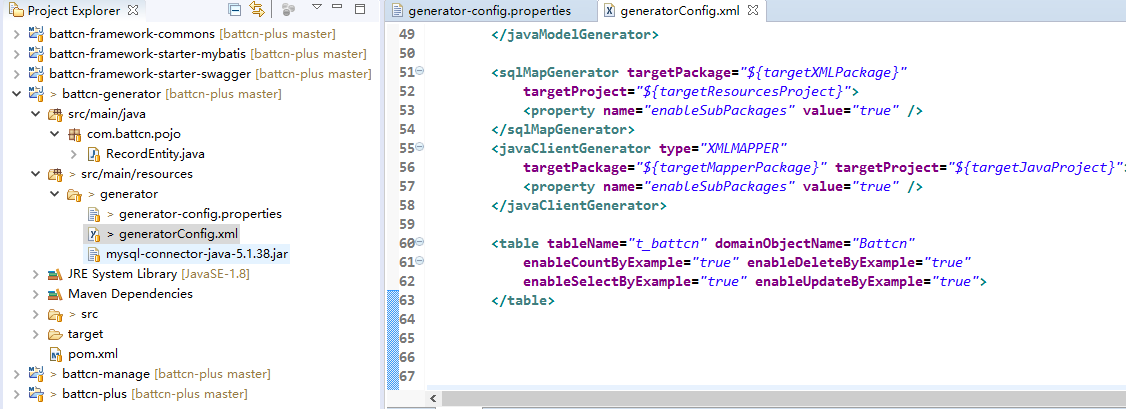
|  |
| --- |
| CREATE TABLE `t\_hello` (  `id` int(11) NOT NULL AUTO\_INCREMENT COMMENT '自增ID',  `name` varchar(50) DEFAULT NULL COMMENT '姓名',  `battcn` varchar(40) DEFAULT NULL COMMENT 'Q群',  `qq` varchar(28) DEFAULT NULL COMMENT 'QQ号',  `gmt\_create` datetime DEFAULT CURRENT\_TIMESTAMP COMMENT '创建时间',  `gmt\_modified` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP COMMENT '最后修改时间',  PRIMARY KEY (`id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8; |

#### 2.2）使用battcn-generator 生成 实体类和Mapper.java（mvn命令：mybatis-generator:generate）

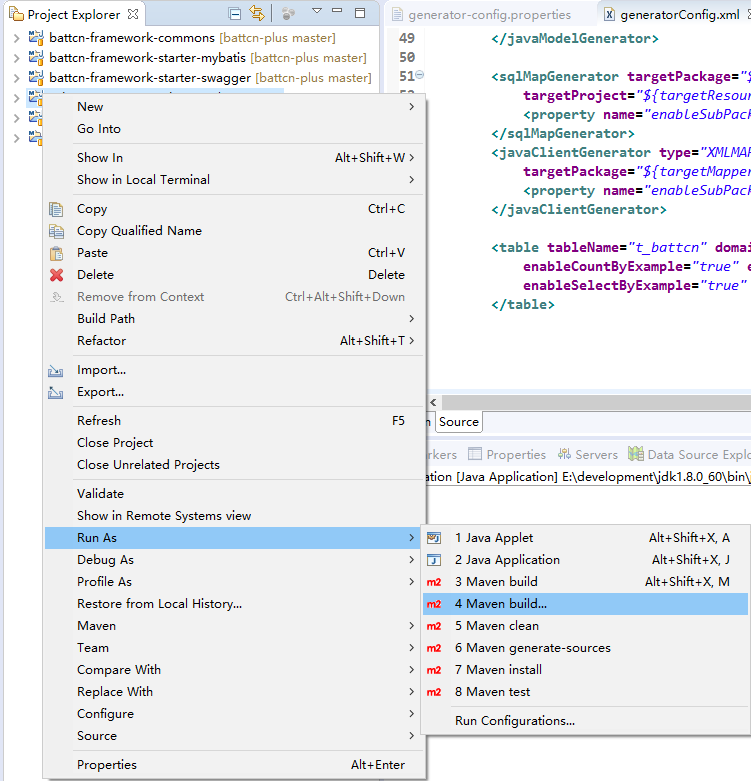
**此处需要注意将自己的JDBC修改正确**



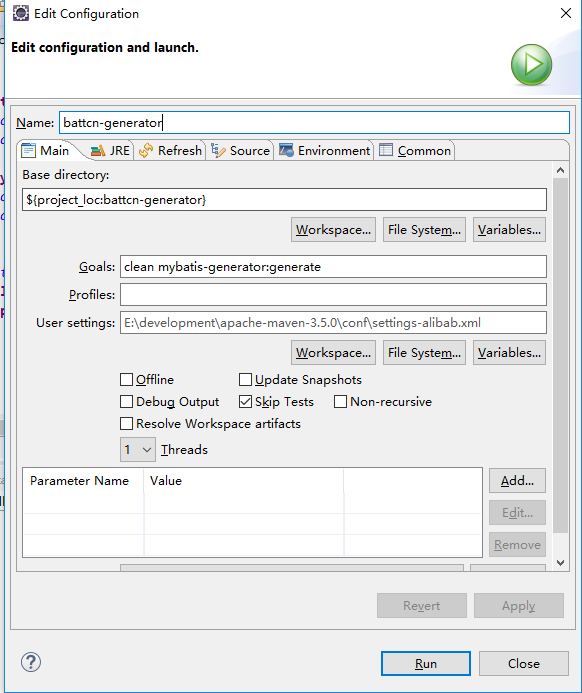
#### 2.3）编辑generatorConfig.xml 添加一个table项



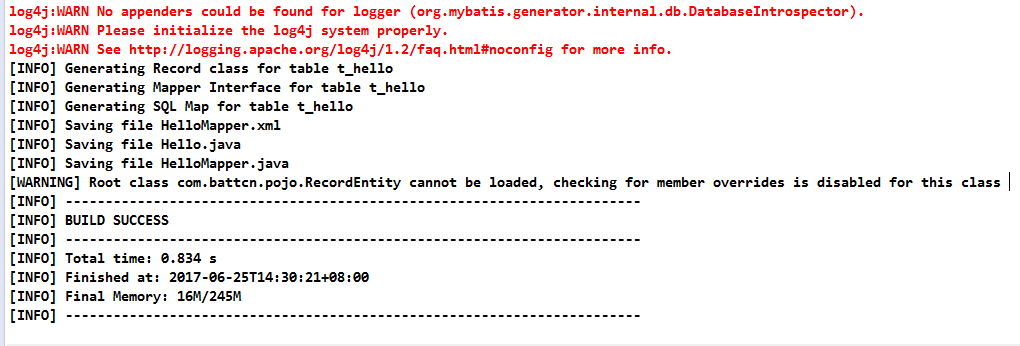
**右键项目：**



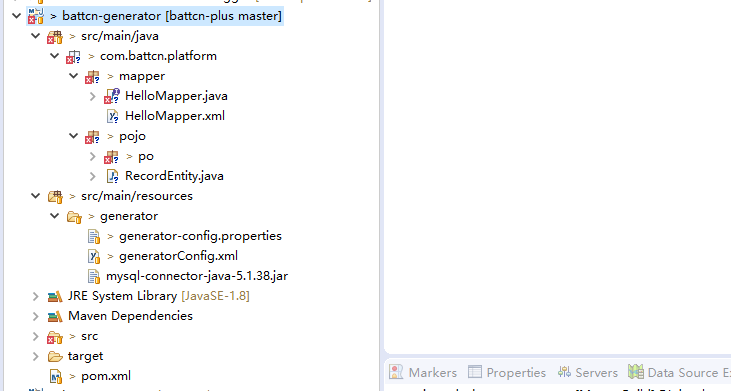
**填写：mybatis-generator:generate 然后点击Run**



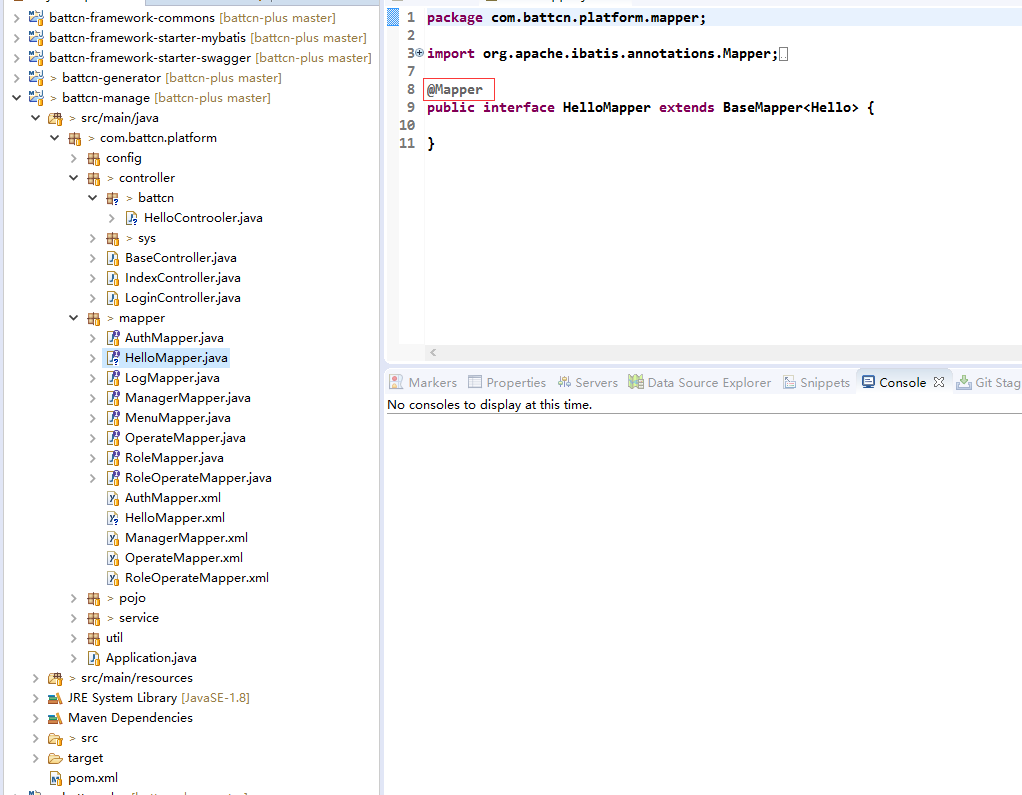
**成功的标志**

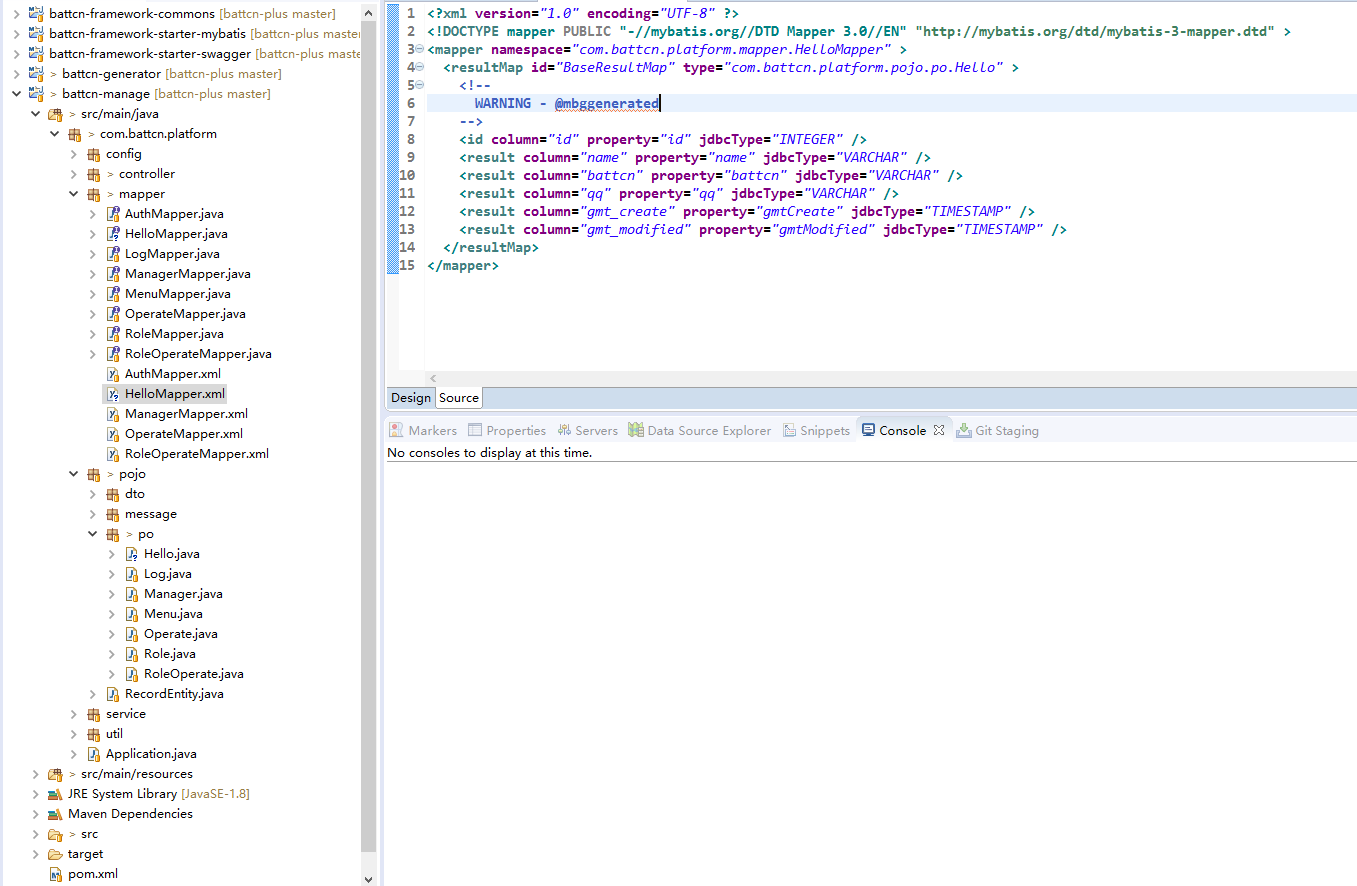


#### 2.4）刷新工程， battcn-generator 生成的代码会有红色错误，拷贝生成的代码到battcn-manage相应的目录下就好了，这个是因为battcn-generator 缺少JAR导致的，不用在意

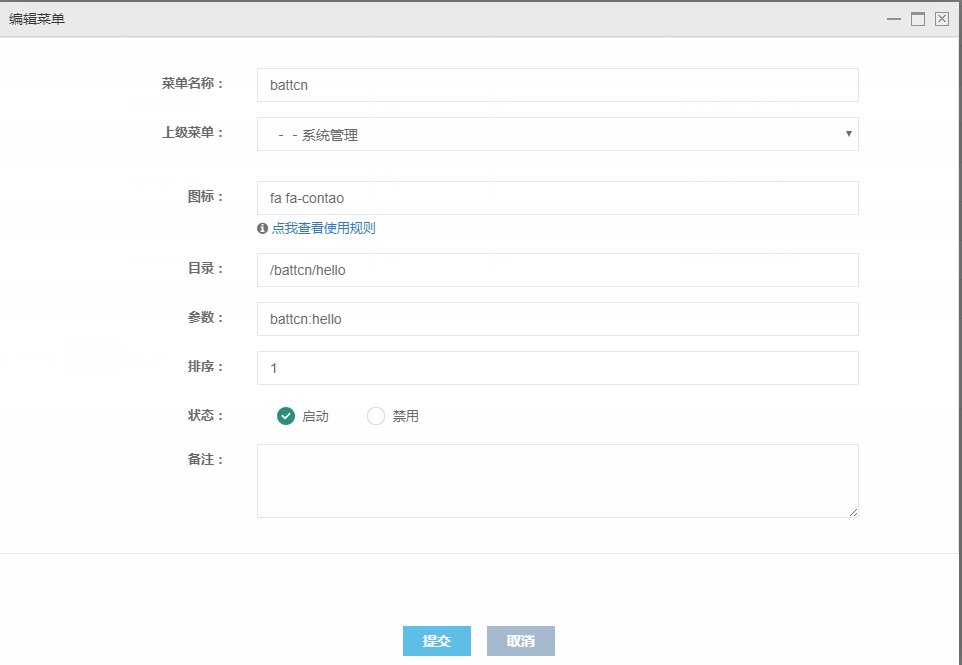


**注意：生成出来的代码是无该注解的，需要自己手动添加**

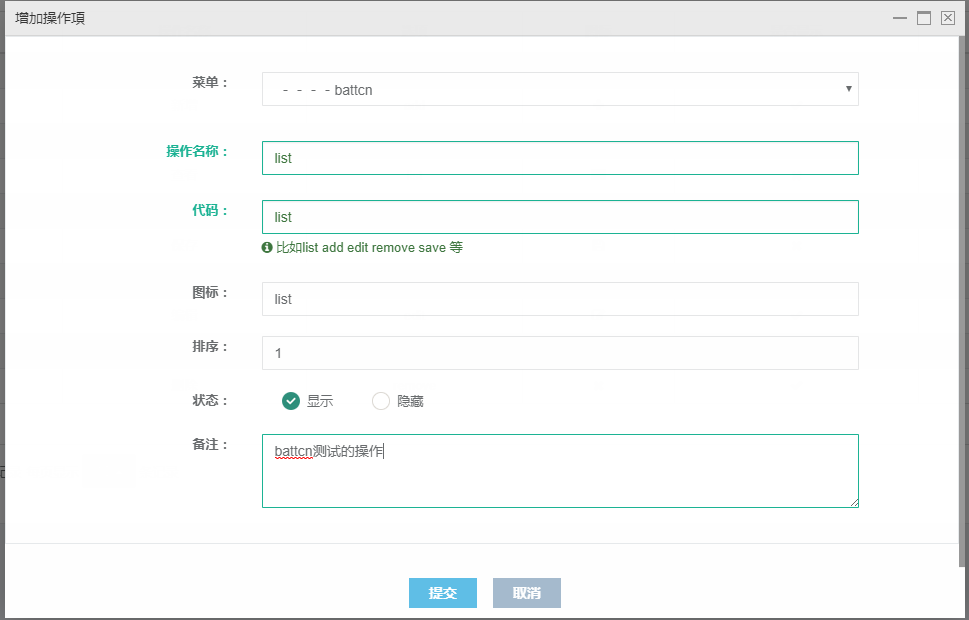


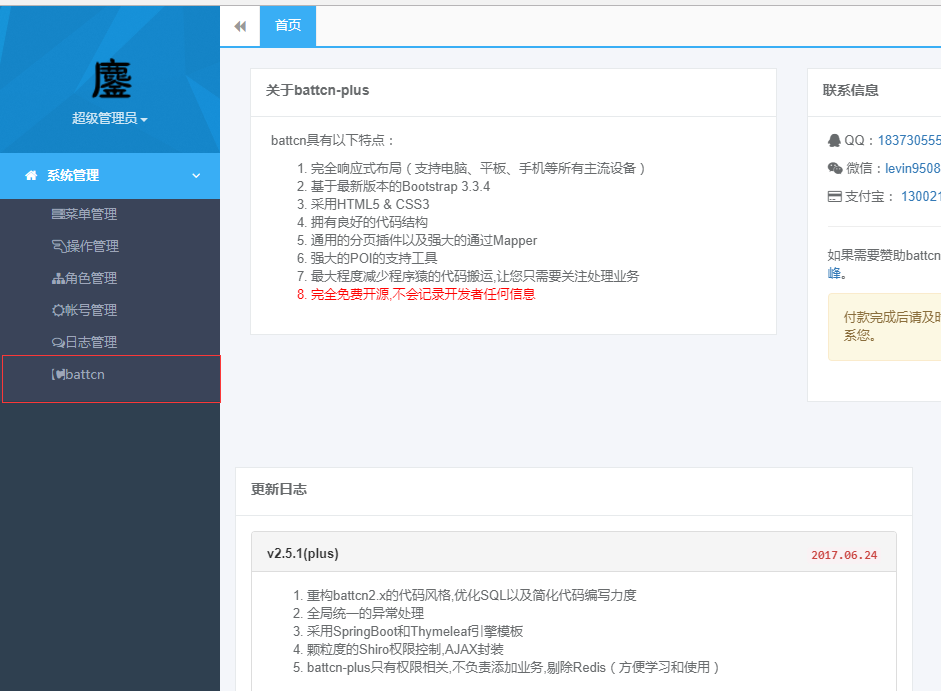


#### 2.5）添加一条操作菜单



#### 2.6）添加操作项，然后刷新页面





编写HelloService.java 继承BaseService

|  |
| --- |
| **package** com.battcn.platform.service;  **import** com.battcn.framework.mybatis.service.BaseService;  **import** com.battcn.platform.pojo.po.Hello;  **public** **interface** HelloService **extends** BaseService<Hello>{  } |

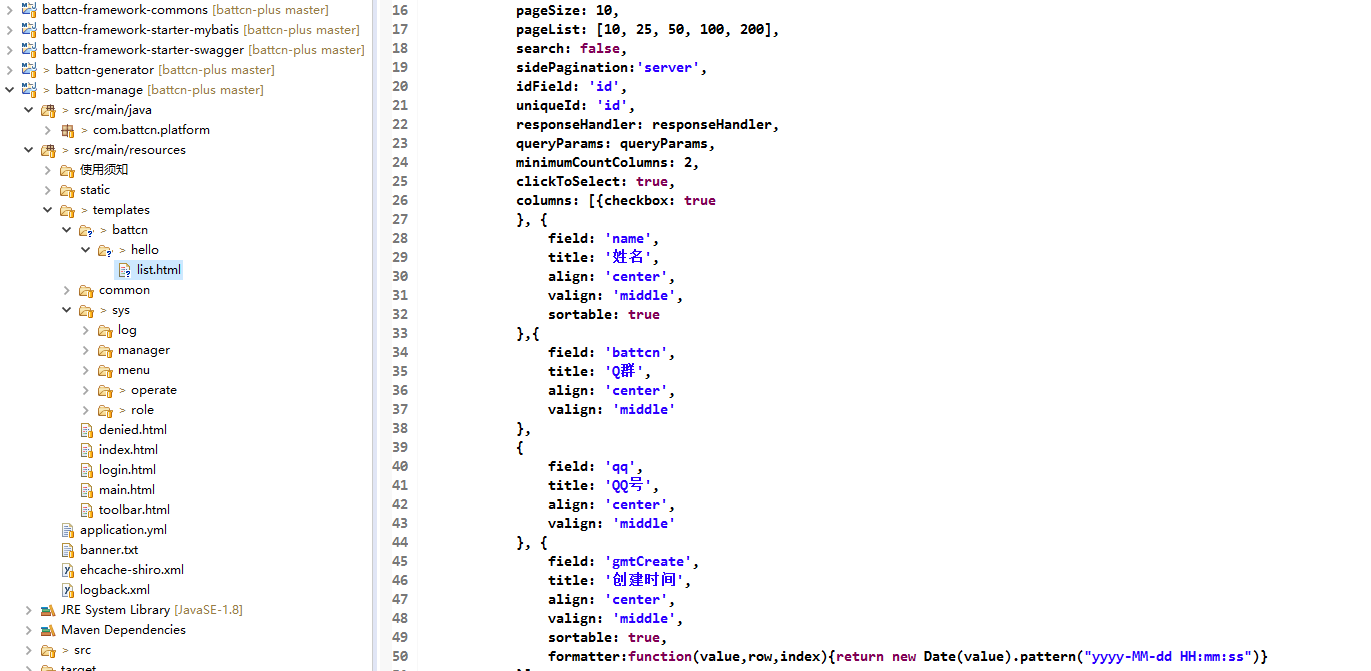
编写HelloServiceImpl.java 继承BaseServiceImpl 实现 HelloService 接口

|  |
| --- |
| **package** com.battcn.platform.service.impl;  **import** org.springframework.stereotype.Service;  **import** com.battcn.framework.mybatis.service.impl.BaseServiceImpl;  **import** com.battcn.platform.pojo.po.Hello;  **import** com.battcn.platform.service.HelloService;  @Service  **public** **class** HelloServiceImpl **extends** BaseServiceImpl<Hello> **implements** HelloService{  } |

编写HelloController.java

|  |
| --- |
| **package** com.battcn.platform.controller.battcn;  **import** org.springframework.beans.factory.annotation.Autowired;  **import** org.springframework.stereotype.Controller;  **import** org.springframework.web.bind.annotation.RequestMapping;  **import** org.springframework.web.bind.annotation.ResponseBody;  **import** com.battcn.framework.common.exception.BattcnException;  **import** com.battcn.framework.mybatis.page.DataGrid;  **import** com.battcn.platform.controller.BaseController;  **import** com.battcn.platform.pojo.po.Hello;  **import** com.battcn.platform.service.HelloService;  **import** com.github.pagehelper.PageInfo;  **import** springfox.documentation.annotations.ApiIgnore;  @Controller  @RequestMapping("/battcn/hello")//此处填写t\_sys\_menu的channel  @ApiIgnore//不加入swagger的扫描中去  **public** **class** HelloControoler **extends** BaseController{    @Autowired  HelloService helloService;  @RequestMapping(value = "/list")  **public** String list() **throws** BattcnException {  **return** "battcn/hello/list";  }  @RequestMapping(value = "/query")  @ResponseBody  **public** PageInfo<Hello> query(DataGrid grid) {  **return** **this**.helloService.listForDataGrid(grid);  }  } |

编写battcn/hello/list.html（直接copy旧的，比如sys/log/list.html）只需要修改columns 里面的字段就好了，其他的一概不需要修改，除非特殊情况

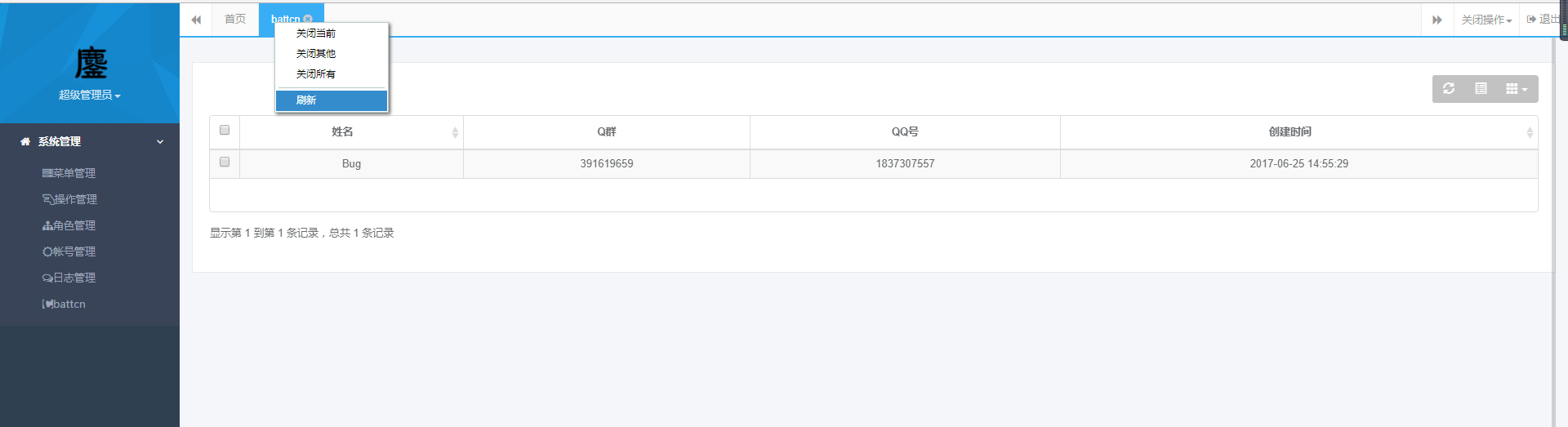


讲了这么多，能不能装逼成功还要看Application.java运行的效果

通过日志我们可以分析出来，查询成功无数据，我们模拟一条数据



通过SQL添加一条数据，查询成功了，同理增删改就留给各位了



## 三：说点什么

1. battcn秉承开源精神，只要不拿着battcn的代码挂淘宝就心满意足了，该源码简化了CRUD的操作，同时具备重要日志记录，权限，以及面向对象，代码干净整洁，也欢迎各位给出宝贵的意见，如果您喜欢这套代码，点下watcg，start即可欢迎fork欢迎捐助（PS：主要是演示地址服务器要钱）
2. 花半个小时熟悉他，以后简单CRUD5分钟包括前后端
3. 静态资源：请参考项目中的README.me文件，自行下载即可

