

renren-fast开发文档_试看版

1. 介绍

1.1. 项目描述

- **renren-fast**是一个轻量级的 `Spring Boot` 快速开发平台，能快速开发项目并交付【接私活利器】
 - 完善的 `xss` 防范及脚本过滤，彻底杜绝 `xss` 攻击
 - 实现前后端分离，通过 `token` 进行数据交互
 - 推荐使用阿里云服务器部署项目，免费领取阿里云优惠券，请点击【[免费领取](#)】
-

1.2. 获取帮助

- Git地址：<http://git.oschina.net/babaio/renren-fast>
 - 代码生成器：<http://git.oschina.net/babaio/renren-generator>
 - 其他项目：<http://www.renren.io/open/>
 - 如需关注项目最新动态，请Watch、Star项目，同时也是对项目最好的支持
-

1.3. 官方QQ群

- 高级群：324780204(大牛云集，跟大牛学习新技能)
 - 普通群：145799952(学习交流，互相解答各种疑问)
-

2. 入门

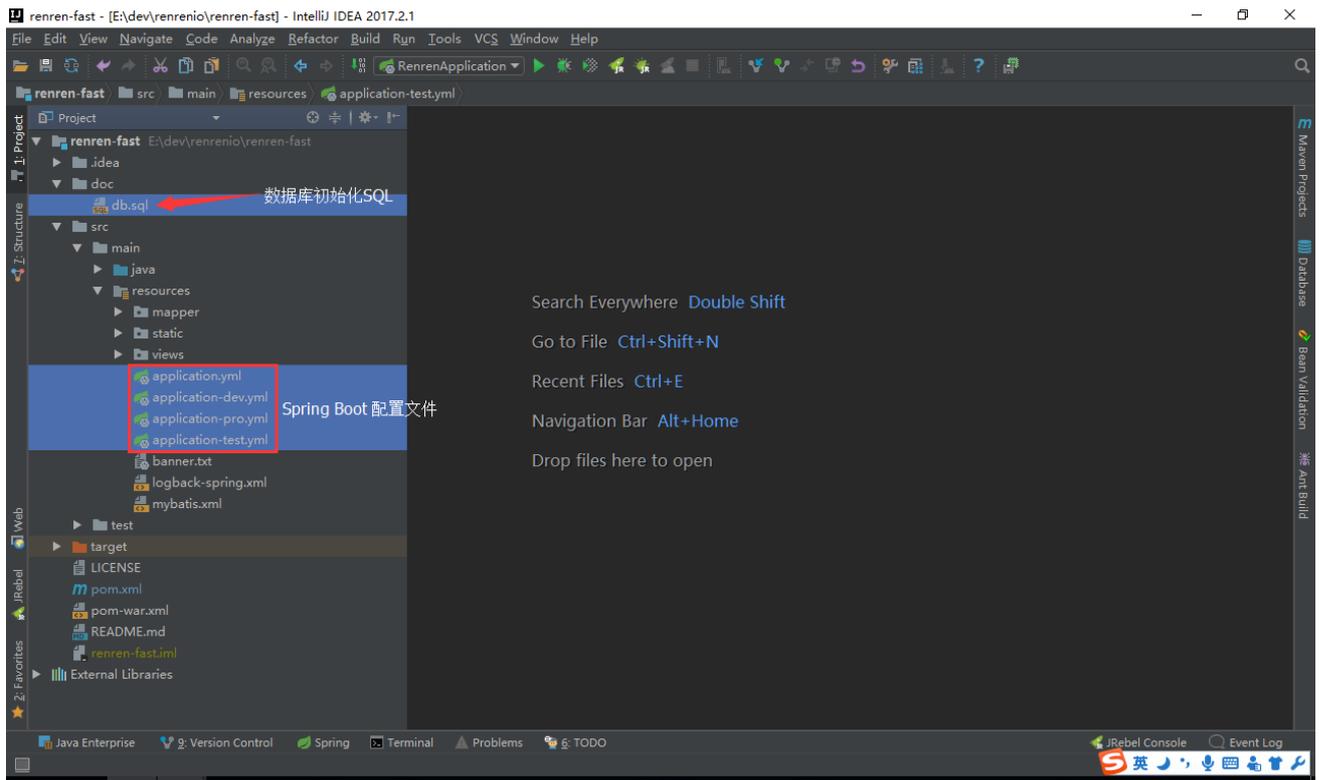
2.1. 快速开始

2.1.1. 本地部署

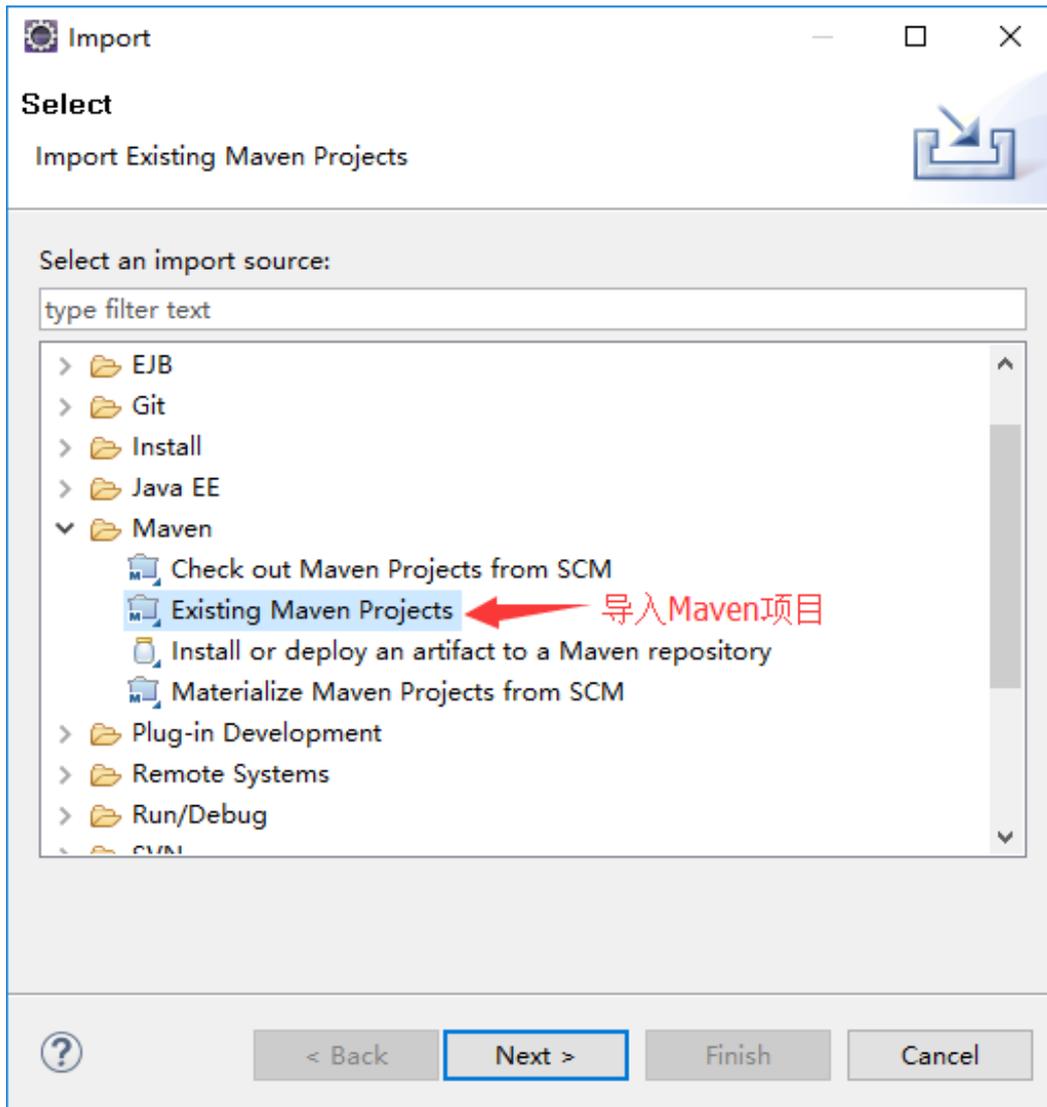
- 环境要求 JDK1.8、Tomcat8.0+、MySQL5.5+
- 通过 git ，下载renren-fast源码，如下：

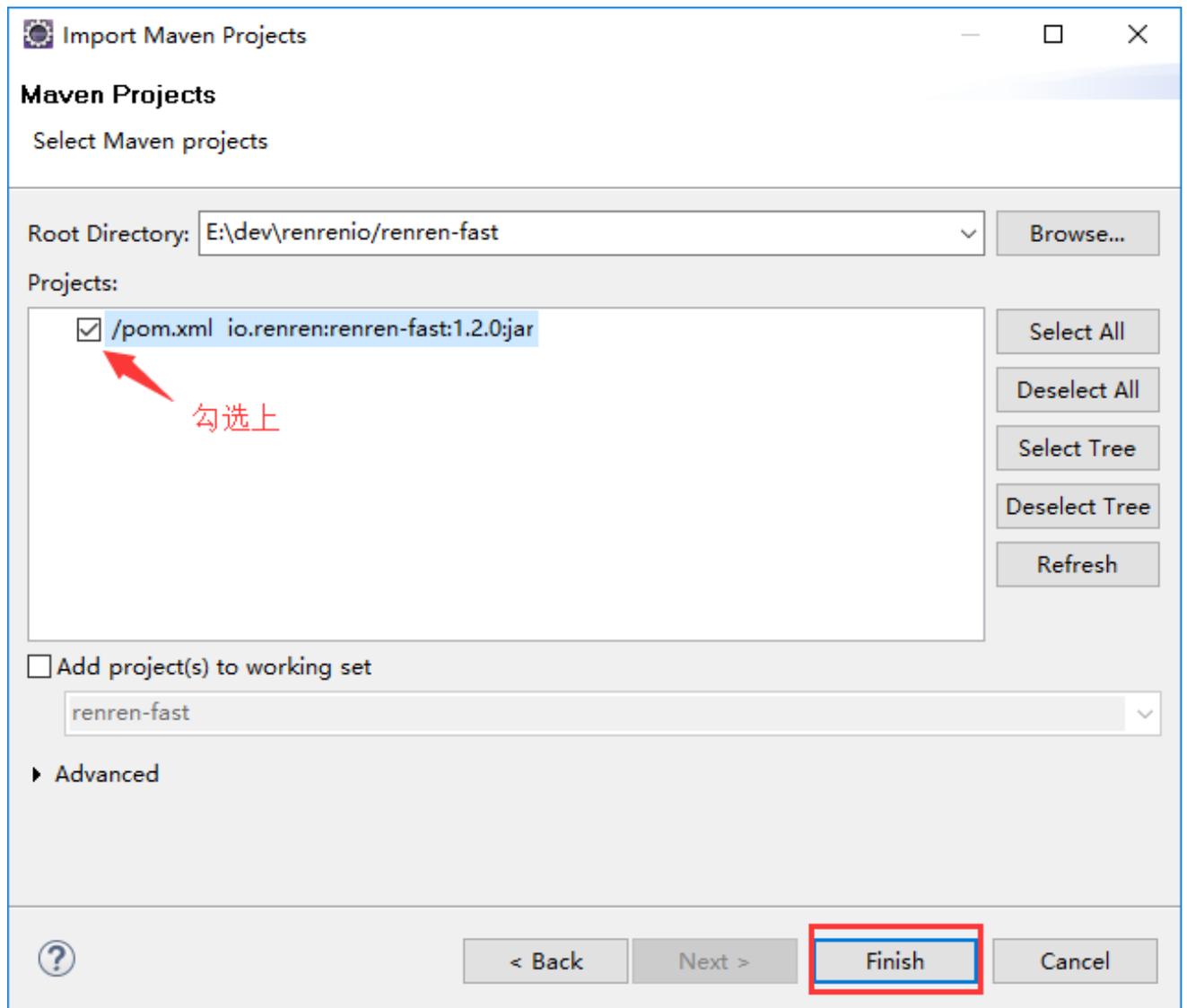
```
1. git clone https://git.oschina.net/babaio/renren-fast.git
```

- 用 idea 打开项目，File -> Open 如下图：



- 用 eclipse 打开项目，如下图：





- 创建数据库 `renren_fast` ，数据库编码为 `UTF-8`
- 执行 `doc/db.sql` 文件，初始化数据
- 修改 `application-dev.yml` ，更新MySQL账号和密码
- 运行 `io.renren.RenrenApplication.java` 的 `main` 方法，则可启动项目
- 项目访问路径：<http://localhost:8080/renren-fast>
- 账号密码：`admin/admin`
- Swagger路径：<http://localhost:8080/renren-fast/swagger/index.html>

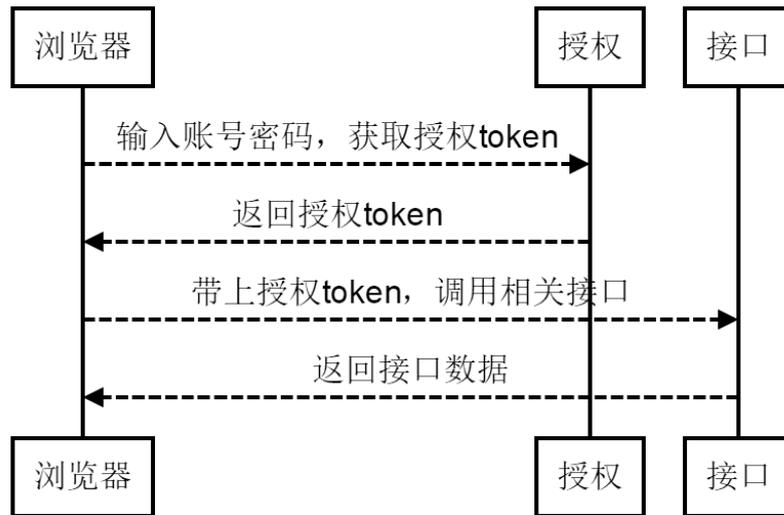
2.1.2. 配置文件

```
1. # Tomcat
2. server:
3.     tomcat:
4.         uri-encoding: UTF-8
5.         max-threads: 1000
6.         min-spare-threads: 30
7.         port: 8080 #tomcat端口号
8.         context-path: /renren-fast
9.
10. spring:
11.     # 环境 dev|test|pro
12.     profiles:
13.         active: dev
14.     # jackson时间格式化
15.     jackson:
16.         time-zone: GMT+8
17.         date-format: yyyy-MM-dd HH:mm:ss
18.     http:
19.         multipart:
20.             max-file-size: 100MB
21.             max-request-size: 100MB
22.             enabled: true
23.     resources: # 指定静态资源的路径
24.         static-locations: classpath:/static/,classpath:/views/
25.     redis:
26.         open: false # 是否开启redis缓存 true开启 false关闭
27.         database: 0
28.         host: redis.open.renren.io
29.         port: 16379
30.         password: # 密码（默认为空）
31.         timeout: 6000 # 连接超时时长（毫秒）
32.         pool:
33.             max-active: 1000 # 连接池最大连接数（使用负值表示没有限制）
34.             max-wait: -1 # 连接池最大阻塞等待时间（使用负值表示没有限制）
35.             max-idle: 10 # 连接池中的最大空闲连接
36.             min-idle: 5 # 连接池中的最小空闲连接
37.
38. # Mybatis配置
39. mybatis:
40.     mapperLocations: classpath:mapper/**/*.xml
41.     configFileLocation: classpath:mybatis.xml
42.
43. # 数据库配置
44. spring:
45.     datasource:
```

```
46.         type: com.alibaba.druid.pool.DruidDataSource
47.         driverClassName: com.mysql.jdbc.Driver
48.         druid:
49.             first: #数据源1
50.                 url: jdbc:mysql://localhost:3306/renren_fast?allowMulti
Queries=true&useUnicode=true&characterEncoding=UTF-8
51.                 username: renren
52.                 password: 123456
53.             second: #数据源2
54.                 url: jdbc:mysql://localhost:3306/bdshop?
allowMultiQueries=true&useUnicode=true&characterEncoding=UTF-8
55.                 username: renren
56.                 password: 123456
57.             initial-size: 10
58.             max-active: 100
59.             min-idle: 10
60.             max-wait: 60000
61.
```

2.2. 数据交互

- 一般情况下，web项目都是通过session进行认证，每次请求数据时，都会把jsessionId放在cookie中，以便与服务端保持会话
- 本项目是前后端分离的，通过token进行认证（登录时，生成唯一的token凭证），每次请求数据时，都会把token放在header中，服务端解析token，并确定用户身份及用户权限，数据通过json交互
- 数据交互流程：



3. 项目实战

3.1. 功能描述

我们来完成一个商品的列表、添加、修改、删除功能，熟悉如何快速开发自己的业务功能模块。

- 我们先建一个商品表tb_goods，表结构如下所示：

```

1. CREATE TABLE `tb_goods` (
2.     `goods_id` bigint NOT NULL AUTO_INCREMENT,
3.     `name` varchar(50) COMMENT '商品名',
4.     `intro` varchar(500) COMMENT '介绍',
5.     `price` decimal(10,2) COMMENT '价格',
6.     `num` int COMMENT '数量',
7.     PRIMARY KEY (`goods_id`)
8. ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COMMENT='商品管理';
  
```

- 接下来，我们利用代码生成器，帮我们生成基础代码，可以大大的节省重复工作量，代码生成器Git地址：<https://git.oschina.net/babaio/renren-generator>

3.2. 使用代码生成器

- 代码生成器是Spring Boot开发的项目，通过git clone把项目下载到本地后，直接运行main方法，就可以通过<http://localhost>打开，我们先来看看配置文件，如下所示：

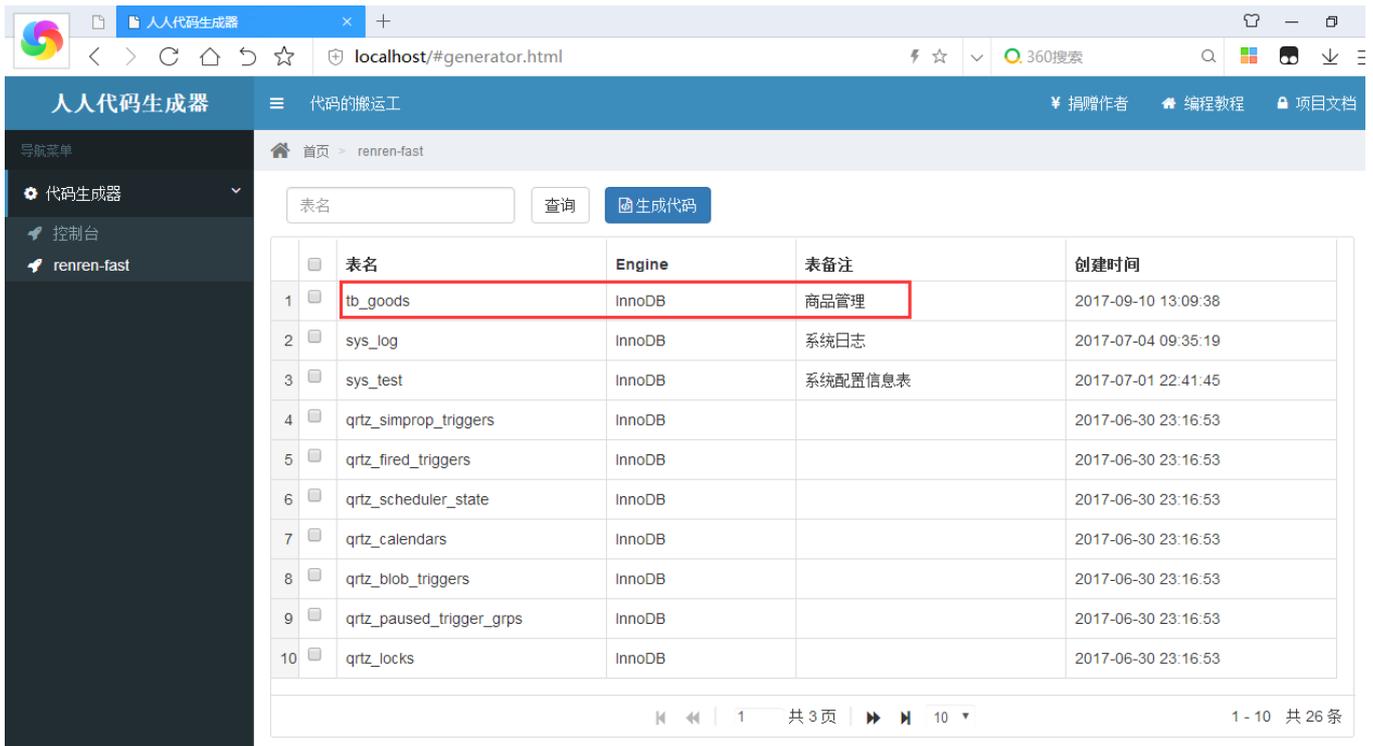
```
1.  #配置文件：generator.properties
2.
3.  #包名
4.  package=io.renren.modules.generator
5.  #作者
6.  author=chenshun
7.  #Email
8.  email=sunlightcs@gmail.com
9.  #表前缀(类名不会包含表前缀)
10. tablePrefix=tb_
11.
12. #类型转换, 配置信息
13. tinyint=Integer
14. smallint=Integer
15. mediumint=Integer
16. int=Integer
17. integer=Integer
18. bigint=Long
19. float=Float
20. double=Double
21. decimal=BigDecimal
22. bit=Boolean
23.
24. char=String
25. varchar=String
26. tinytext=String
27. text=String
28. mediumtext=String
29. longtext=String
30.
31. date=Date
32. datetime=Date
33. timestamp=Date
```

上面的配置文件，可以配置包名、作者信息、表前缀、类型转换。其中，类型转换是指，MySQL中的类型与JavaBean中的类型，是怎么一个对应关系。如果有缺少的类型，可自行在generator.properties文件中补充。

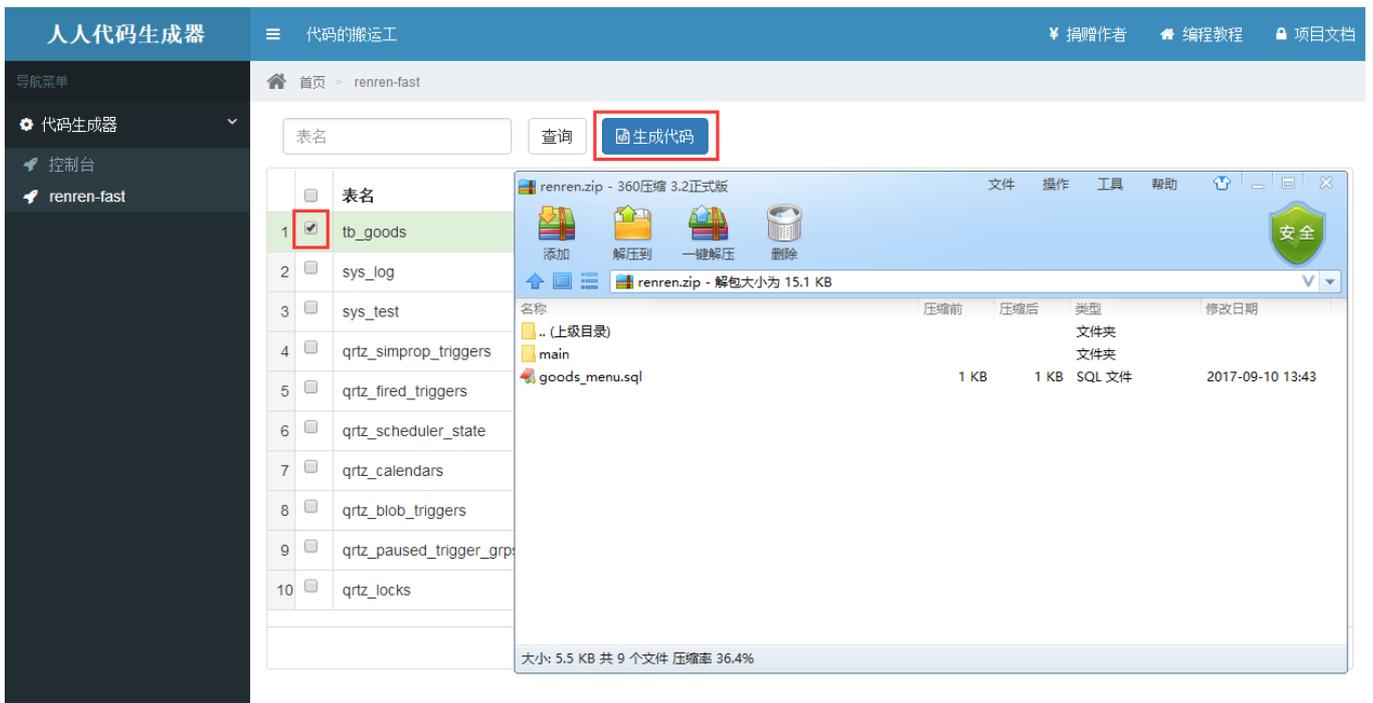
- 再看看application.yml配置文件，我们只要修改数据库名、账号、密码，就可以了。其中，数据库名是指待生成的表，所在的数据库。

```
1. # Tomcat
2. server:
3.     tomcat:
4.         max-threads: 10
5.         min-spare-threads: 10
6.         port: 80
7.
8. # mysql
9. spring:
10.    datasource:
11.        type: com.alibaba.druid.pool.DruidDataSource
12.        driverClassName: com.mysql.jdbc.Driver
13.        url: jdbc:mysql://localhost:3306/renren_fast?useUnicode=true&ch
14.        aracterEncoding=UTF-8
15.        username: renren
16.        password: 123456
17.    jackson:
18.        time-zone: GMT+8
19.        date-format: yyyy-MM-dd HH:mm:ss
20.    resources:
21.        static-locations: classpath:/static/,classpath:/views/
22. # Mybatis配置
23. mybatis:
24.    mapperLocations: classpath:mapper/**/*.xml
```

- 在数据库renren_fast中，执行建表语句，创建tb_goods表，再启动renren-generator项目(运行RenrenApplication.java的main方法即可)，如下所示：



- 我们只需勾选tb_goods，点击【生成代码】按钮，则可生成相应代码，如下所示：



3.3. 测试项目

- 我们先在renren_fast库中，执行goods_menu.sql语句，这个SQL是生成菜单的，SQL语句如下所示：

```

1.  -- 菜单SQL
2.  INSERT INTO `sys_menu` (`parent_id`, `name`, `url`, `perms`, `type`, `i
   con`, `order_num`)
3.      VALUES ('1', '商品管理', 'modules/generator/goods.html', NULL, '1',
   'fa fa-file-code-o', '6');
4.
5.  -- 按钮父菜单ID
6.  set @parentId = @@identity;
7.
8.  -- 菜单对应按钮SQL
9.  INSERT INTO `sys_menu` (`parent_id`, `name`, `url`, `perms`, `type`, `i
   con`, `order_num`)
10.     SELECT @parentId, '查看', null, 'goods:list,goods:info', '2', null,
   '6';
11. INSERT INTO `sys_menu` (`parent_id`, `name`, `url`, `perms`, `type`, `i
   con`, `order_num`)
12.     SELECT @parentId, '新增', null, 'goods:save', '2', null, '6';
13. INSERT INTO `sys_menu` (`parent_id`, `name`, `url`, `perms`, `type`, `i
   con`, `order_num`)
14.     SELECT @parentId, '修改', null, 'goods:update', '2', null, '6';
15. INSERT INTO `sys_menu` (`parent_id`, `name`, `url`, `perms`, `type`, `i
   con`, `order_num`)
16.     SELECT @parentId, '删除', null, 'goods:delete', '2', null, '6';

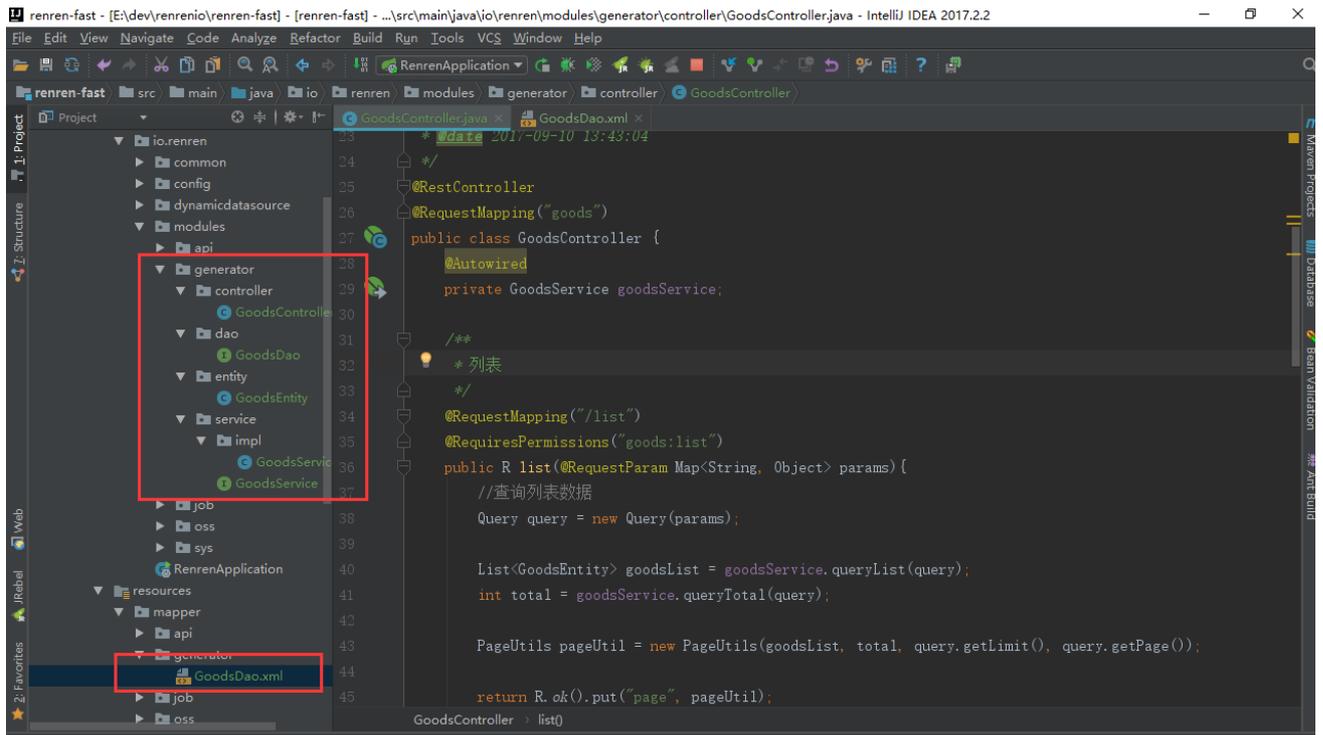
```

- 然后把生成的main目录覆盖renren-fast的main目录，再重启renren-fast，效果如下所示：

菜单ID	菜单名称	上级菜单	图标	类型	排序号	菜单URL	授权标识
4	菜单管理	系统管理	☰	菜单	3	modules/sys/menu.h...	
5	SQL监控	系统管理	🔍	菜单	4	druid/sql.html	
6	定时任务	系统管理	🕒	菜单	5	modules/job/schedul...	
27	参数管理	系统管理	⚙️	菜单	6	modules/sys/config....	sys:config:list,sys:...
30	文件上传	系统管理	📁	菜单	6	modules/oss/oss.html	sys:oss:all
31	商品管理	系统管理	📦	菜单	6	modules/generator/g...	
32	查看	商品管理		按钮	6		goods:list,goods:info
33	新增	商品管理		按钮	6		goods:save
34	修改	商品管理		按钮	6		goods:update
35	删除	商品管理		按钮	6		goods:delete
29	系统日志	系统管理	📄	菜单	7	modules/sys/log.html	sys:log:list



- 我们就操作了这几步，就把查询、新增、修改、删除就完成了，是不是很快啊，下面就是我们才生成的代码，如下所示：



4. 源码分析

4.1. 多数据源

4.1.1. 实现多数据源

4.1.2. 测试多数据源

4.1.3. 增加多数据源

4.1.4. 移除多数据源

4.2. 核心模块

4.2.1. 前后端分离

4.2.2. 权限设计思路

4.2.3. XSS脚本过滤

4.2.4. SQL注入

4.2.5. Redis缓存

4.2.6. 异常处理机制

4.2.7. 后端效验机制

4.2.8. 系统日志

4.2.9. 添加菜单

4.2.10. 添加角色

4.2.11. 添加管理员

4.3. 定时任务模块

4.3.1. 新增定时任务

4.3.2. 源码分析

4.4. 云存储模块

4.4.1. 七牛的配置

4.4.2. 阿里云的配置

4.4.3. 腾讯云的配置

4.4.4. 源码分析

4.5. API模块

4.5.1. API的使用

4.5.2. 源码分析

5. 生产环境部署

5.1. 常规部署

5.2. war包部署

5.3. docker部署

6. 捐赠获取完整版

捐赠【50元】，即可获取【完整版】的开发文档，捐赠时，需备注邮箱地址，通过邮件的形式，发送到你邮箱

备注：相信有了你们的支持，人人开源会越来越好，项目也会越来越完善。

通过微信或支付宝，扫描下面的二维码，即可捐赠，如下图所示：

微信



支付宝



