

# IDEA 插件 EasyYapi

- 功能
- 安装
- 配置
- 使用
- 总结
- 写在最后

## IDEA 插件 EasyYapi

通过 YAPI 的 [开放 API](#) , 首先我们会产生一个想法, 开发一个 IDEA 的插件, 产生这个想法以后, 就去先去插件市场里搜寻, 果然已经有很多大佬先一步开发好了. 这里先一步帮大家也筛选了一下, 选择了对代码 0 入侵的 [EasyYapi](#)

## 功能

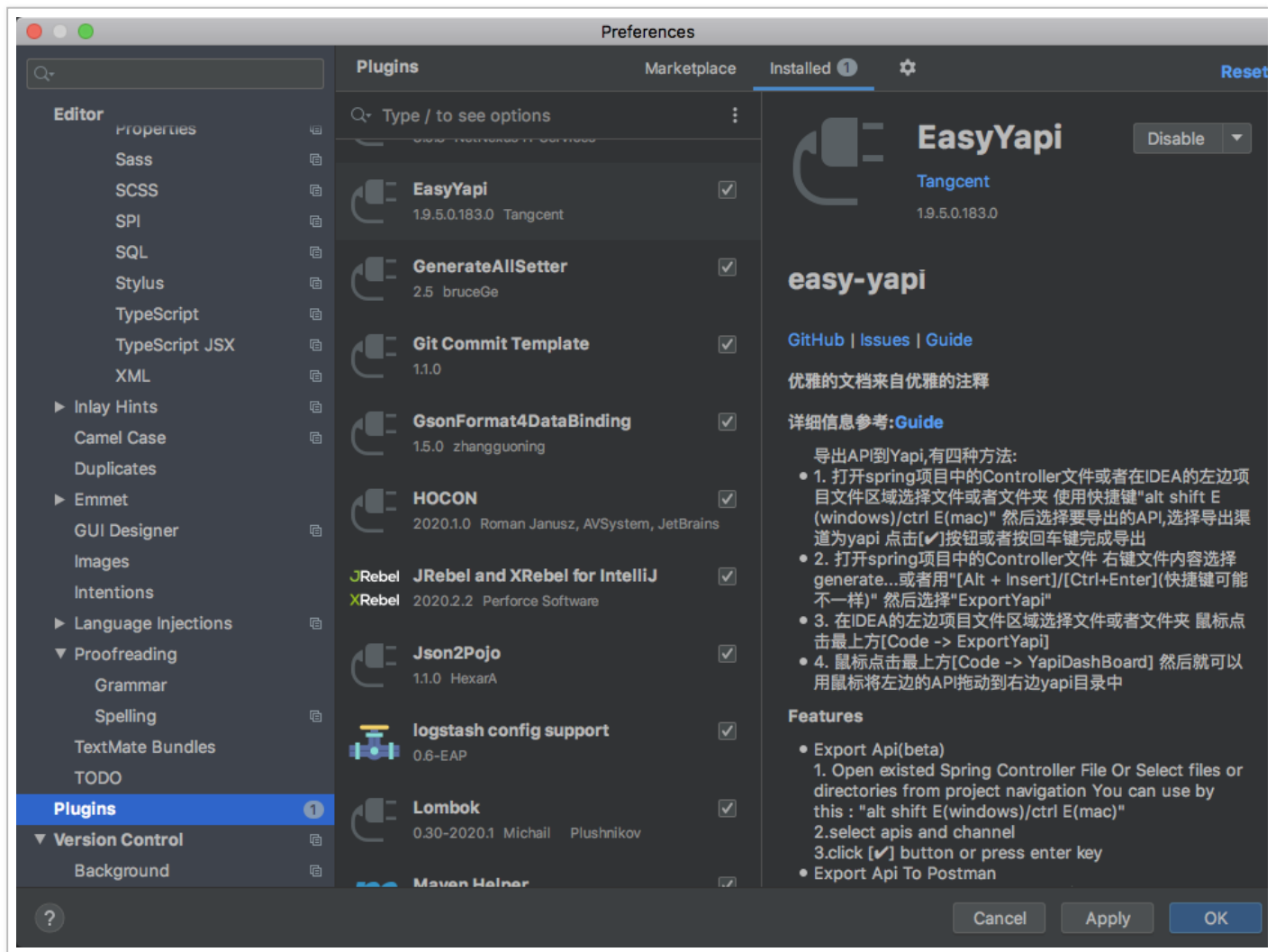
---

1. 导出 http 到
  1. Yapi
  2. Postman
  3. markdown
2. 导出 rpc 到
  1. Yapi
  2. markdown
3. Call api 调用 API

## 安装

---

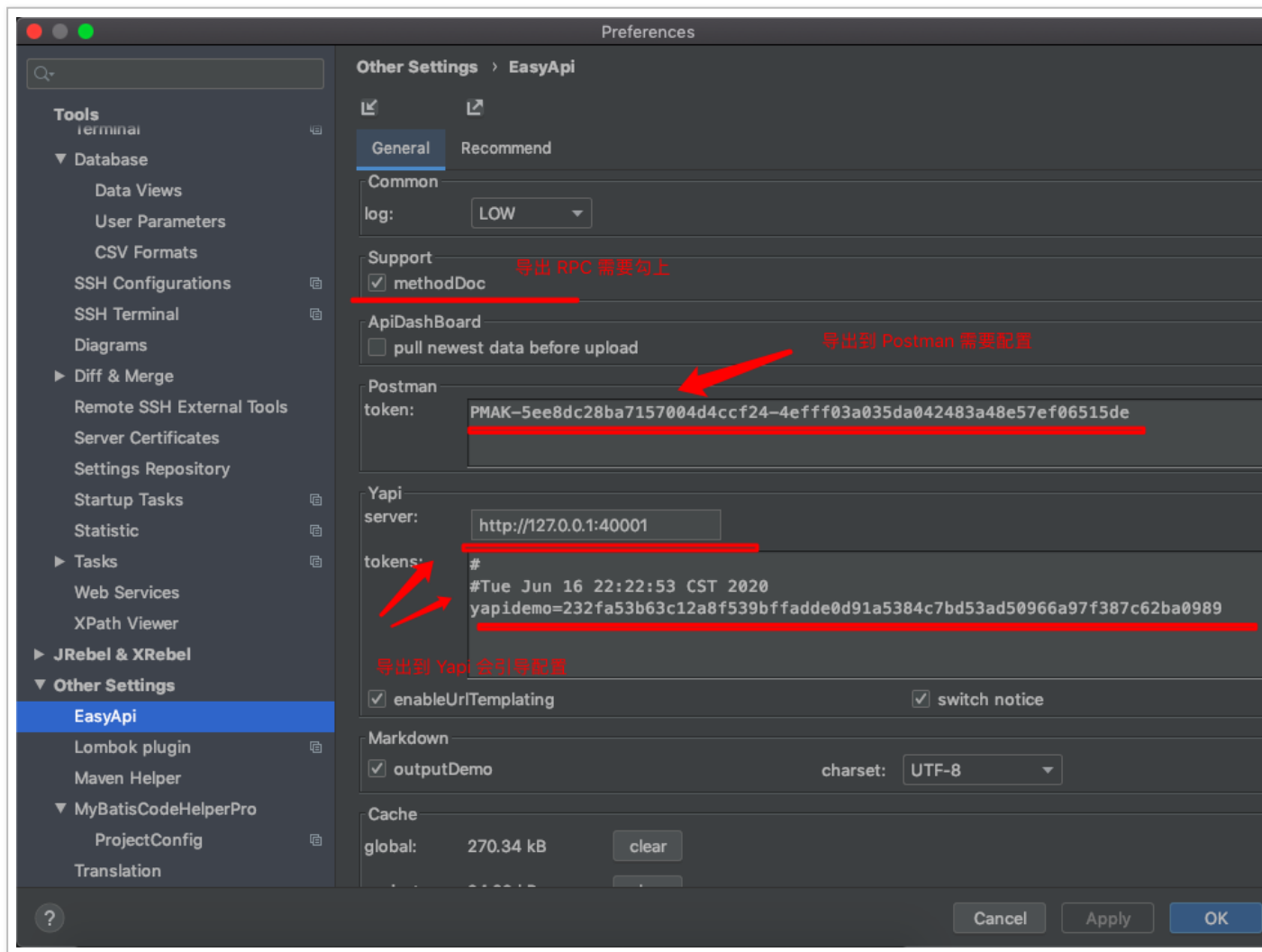
Preferences → Plugins → Marketplace → 搜索 "EasyYapi" → 安装该插件 → 重启 IDE



## 配置

进入 `Preferences (Settings) > Other Settings > EasyApi > Support` , 主有三项:

- Postman token 配置
- Yapi 配置 , 第一次导出的时候会引导配置, 这里不用手动配置
- RPC : Support >[v] methodDoc

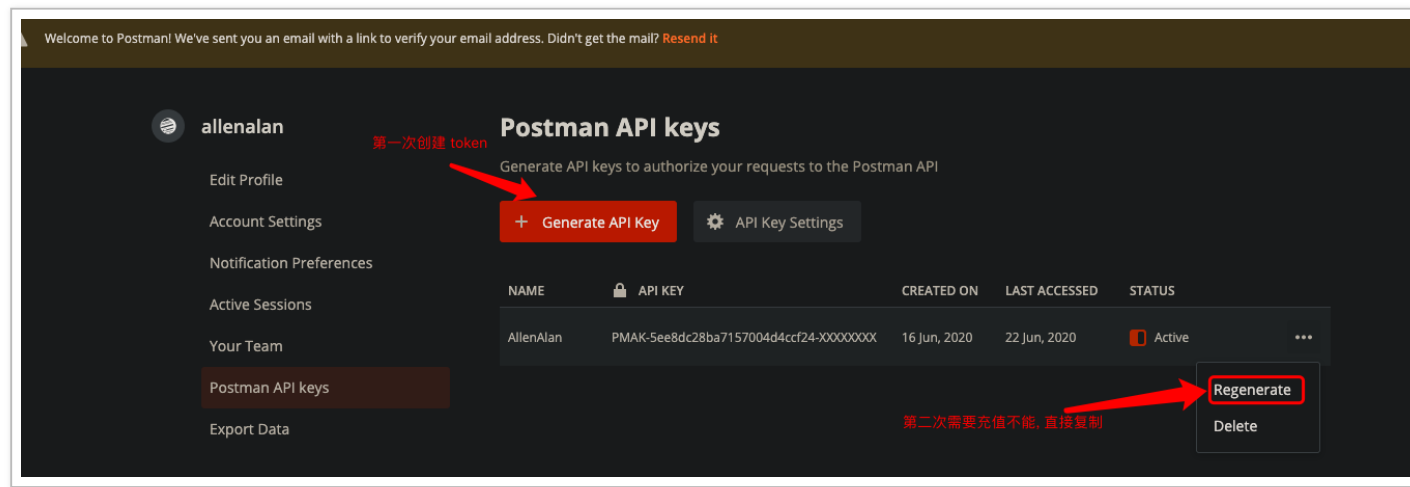


## YAPI token 获取



# Postman token 获取

Postman token 获取页面



# 使用

## 1. 导出

选中 一个或多个 API 文件 (Controller 或是 RPC Interface), 使用快捷键 `alt shift E(windows)/ctrl E(mac)`

选导出类型以及导出的接口, 确定后回车

导出到 YAPI 服务



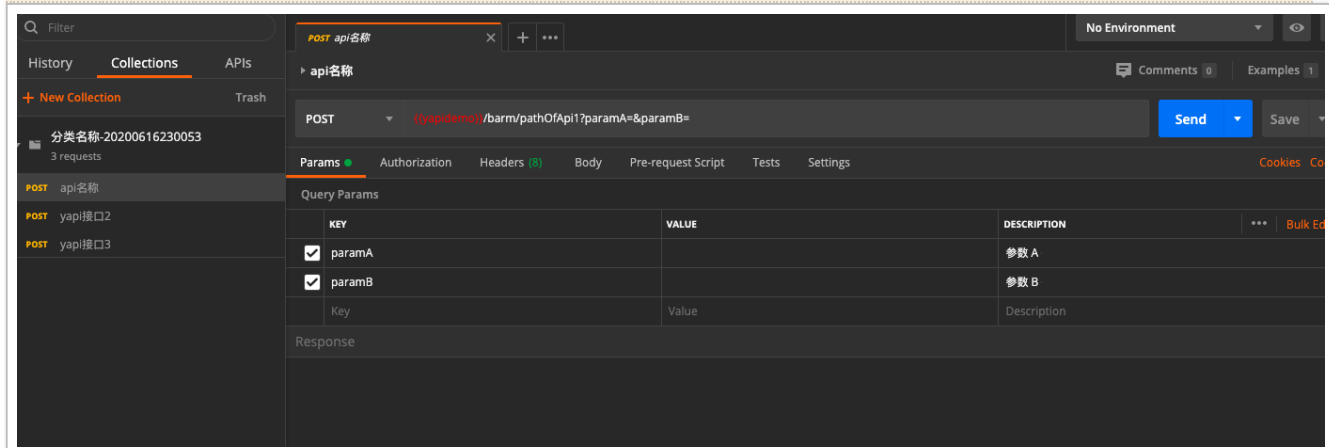
导出到 PostMan

PostMan 是给功能强大的 工具, 应该有很多人跟我一样接口测试会习惯性依赖 PostMan

- 1. `postman` 的 `token` , 可以从 `Postman Integrations Dashboard` 获得
- 2. 可以在 `Preferences(Settings) > Other Settings > EasyApi` 中 `手动配置`

3. 在 API 文件类中使用 快捷键 `Ctrl+Enter` 选择 ExportPostman.

4. 反应会慢一点. 之后就会在你的 PostMan 中出现



导出到 Markdown

文件

Desktop

easy-api.md

easy-api.md

Query:

name	value	required	desc
paramA		NO	参数 A
paramB		NO	参数 B

RESPONSE

Header:

name	value	required	desc
content-type	application/json;charset=UTF-8	NO	

Body:

name	type	desc
code	integer	编码
message	string	信息

Response Demo:

```
{  "code": 0,  "message": ""}
```

yapi接口2

# api名称

## BASIC

Path: /barm/pathOfApi1

Method: POST

Desc:

api名称  
api描述

## REQUEST

Headers:

name	value	required	desc
Content-Type	application/x-www-form-urlencoded	YES	

Query:

name	value	required	desc
paramA		NO	参数 A
paramB		NO	参数 B

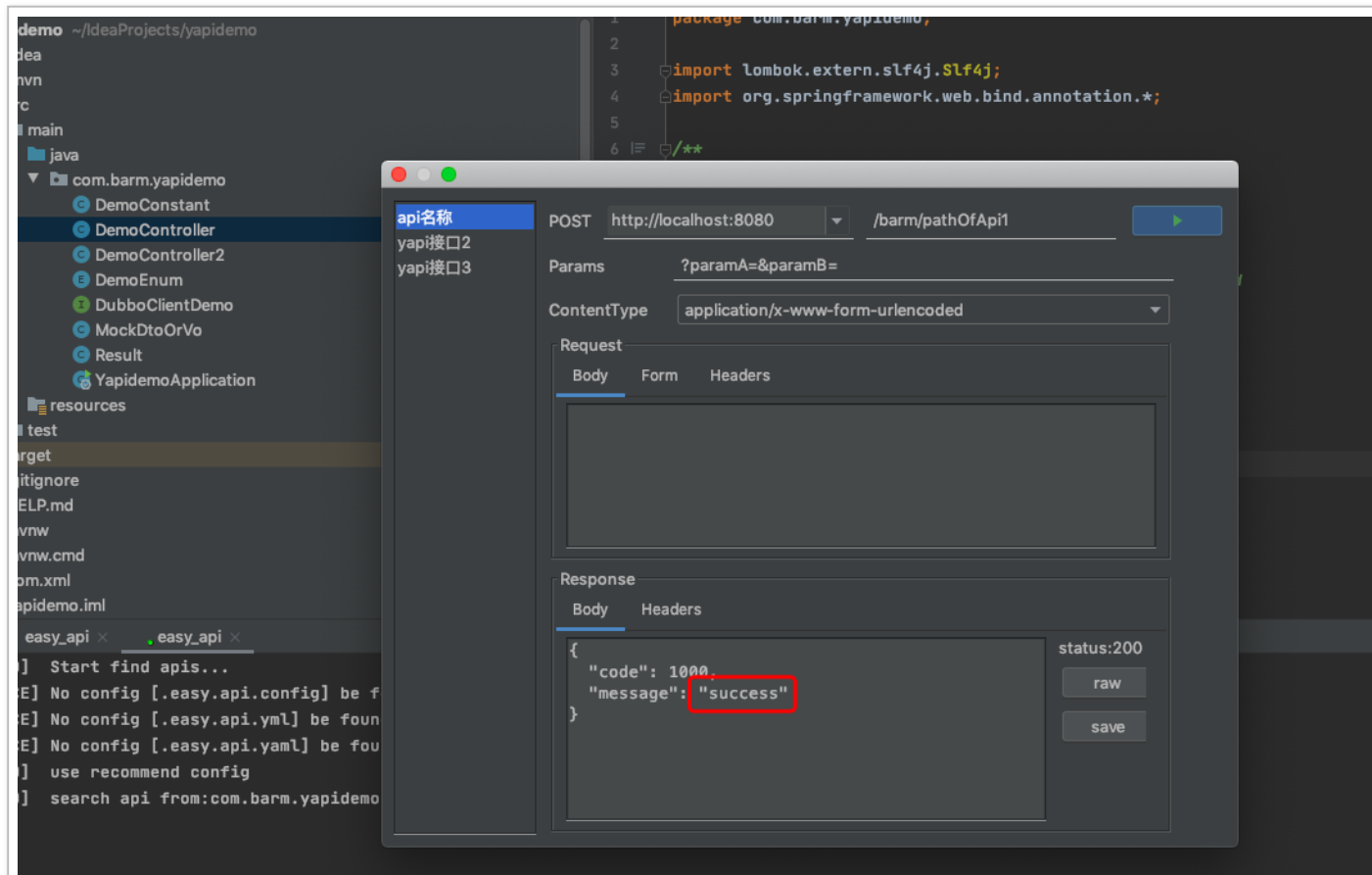
## RESPONSE

Header:

name	value	required	desc
------	-------	----------	------

### 2. Call API

在 api/rpc 文件中 使用 快捷键 `Ctrl+Enter` 选择 Call API , 发起对当前文件中的 API 的请求 (有点鸡肋的功能, 不过在没有 yapi 和 postMan 的情况下可以使用)



## 总结

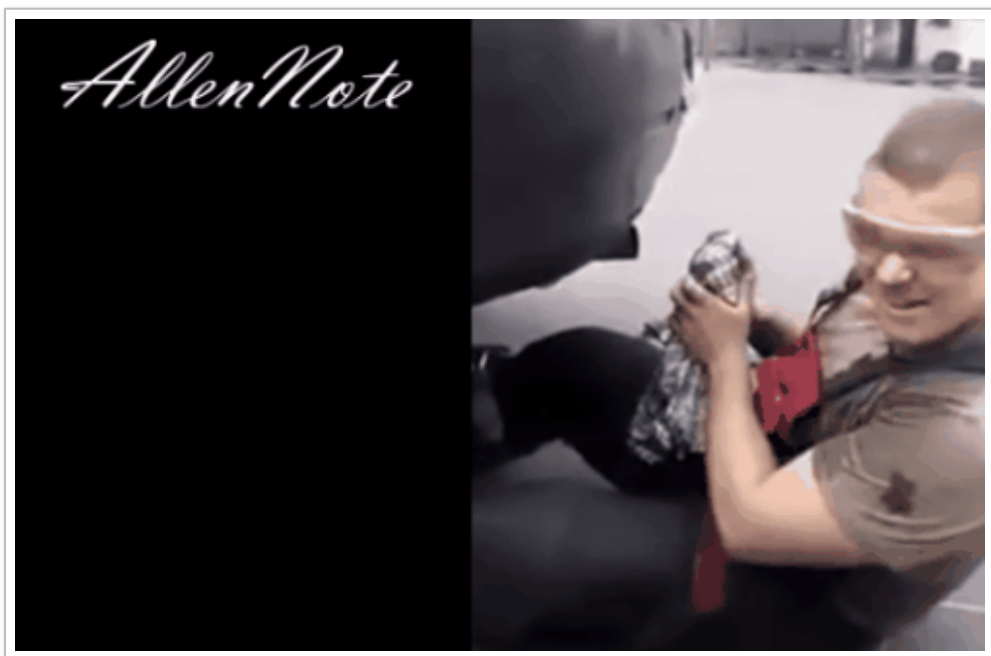
可以看到 EasyYapi 提供了丰富的 API 导出功能, 即使不依赖 YAPI, PostMan 也可以满足基本需求. 另外 EasyYapi 对注释规范要求很高. 不过这另一方面也提高了对编码的注释规范, 有利有弊.

按照 YAPI + EasyYapi 的方式书写 API. 我们一个字段的注释. 在先编写代码再导出到 YAPI 的方式, 我们只需要写一遍注释即可.

先写完接口即可, 后续逻辑导出后再补充也不迟.

## 写在最后

还是老三样. 欢迎 点赞, 转发, 评论 ~



---

全文完

---

本文由 简悦 SimpRead 转码，用以提升阅读体验，原文地址