

OSPP 2022 项目结项报告

Apache APISIX Dashboard V3

项目摘要

Apache APISIX 即将迎来 V3 版本，而 Apache APISIX Dashboard 也需要对 V3 版本进行适配，借此机会，需要将 Dashboard 的前端部分进行重构，以解决历史问题，并带来更好的体验。

提案人：房森 wirth.fang@foxmail.com

导师：杨陶 iskyex@outlook.com

项目摘要

原开发进度表

社区联络期 (June 15 - June 30)

编码阶段一 (July 1 - July 15)

编码阶段二 (July 15 - Aug 30)

[编码阶段三 \(Sep 1 - Sep 23\)](#)

缓冲时间 (7 days)

[产出列表](#)

产出介绍

完成任务过程

为路由页面增加批量删除功能 [issue#2565](#)

[为批量删除增加e2e测试覆盖 issue#2562](#)

[提高e2e测试稳定性与修复重要bug issue#2515 && #2615](#)

[解决仪表盘 Docker 实现使用优化 issue#2446](#)

[为插件编辑器添加提示和预装插件模版 issue#1363](#)

[修复函数命名错误](#)

[使Dashboard支持Windows环境](#)

[重构:将Dashboard包管理器从yarn迁移到pnpm\(扩展性任务\) 相关旧pr #2459](#)

项目评估

项目补充

[遇到的问题及解决方案](#)

[为路由页面增加批量删除功能：](#)

[为插件编辑器添加提示和预装插件模版：](#)

[重构:将Dashboard包管理器从yarn迁移到pnpm\(扩展性任务\)](#)

[与导师沟通及反馈情况](#)

[导师寄语](#)

项目总结

原开发进度表

社区联络期 (June 15 - June 30)

这个阶段的任务是深入了解项目，并可能在此过程中解决一些可能会干扰进度的因素与预案。对所需的技术做一些研究，与社区开发人员和导师交谈，并改变一些技术步骤或计划。

编码阶段一 (July 1 - July 15)

编码阶段一有2周的时间，并在阶段结束后进行评估计划质量与风险。

第一阶段的目标是梳理项目结构，制定合理的更细节的重构计划

编码阶段二 (July 15 - Aug 30)

编码第二阶段有6周时间，阶段结束后进行可能的中期评估，有三个初步目标，至于其执行先后顺序有待商酌。

- ☒ 增加状态管理 // 更改为更合适的批量删除功能
- ☒ 增加e2e测试覆盖
- ☒ 解决有关issue

编码阶段三 (Sep 1 - Sep 23)

编码第三阶段有3周时间，如果任务进行顺利，我们将修复问题（issue）和对Dashboard性能进行优化。我们注意到社区中的一些问题与前端有关，也就是我们的项目控制台。我们可以考虑完成它们。

- ☒ 以ts代替有关js代码段
- ☒ 对Dashboard性能进行优化 // 主要是提高测试稳定性与修错

//后来又添加了个扩展性任务

- ☒ ~~重构:将Dashboard包管理器从yarn迁移到pnpm(扩展性任务)~~

一旦实现了本个阶段的目标，我们会检查代码并修复前面代码中的任何已知bug。编写代码应该被文档化。然后我们的工作基本完成了，然后是最后的评估。

缓冲时间 (7 days)

有9天的缓冲时间，以防前几周发生的事情没有按计划进行。我认为一些工作可能会在两周内完成，所以我们实际上会进展得更快，并拥有更多的容错能力。

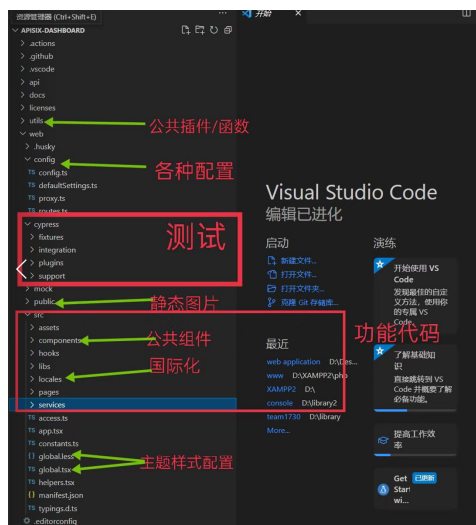
产出列表

使用 TypeScript 作为主要的编程语言

- 梳理项目结构，制定合理的重构计划
- 完成重构方案中路由与页面的重构工作：
 1. 为路由页面增加批量删除功能
 2. 增加e2e测试覆盖
 3. 对Dashboard性能进行优化，主要提高测试稳定性与修复重要bug
 4. 以ts代替有关js代码段
- 解决项目中现存的 issue：
 5. 解决仪表盘 Docker 实现使用优化
 6. 为插件编辑器添加提示和预装插件模版
 7. 修复相关函数错误
- 使Dashboard支持Windows环境
- 重构：将Dashboard包管理器从yarn迁移到pnpm（扩展性任务）

产出介绍

Apache Dashboard是一个基于umi2构建的网关管理系统，其文件目录如下图：



在和社区及导师深入交流之后，制定了两个阶段，八个目标，及完成情况如下：

第一阶段：

- ☒ 为路由页面增加批量删除功能
- ☒ 增加e2e测试覆盖
- ☒ 对Dashboard性能进行优化，主要提高测试稳定性与修复重要bug
- ☒ 以ts代替有关js代码段

第二阶段：

- ☒ 解决仪表板 ~~Docker~~ 实现使用优化
- ☒ 为插件编辑器添加提示和预装插件模版
- ☒ 修复相关函数错误

第三阶段：

- ☒ 使Dashboard支持Windows环境
- ☒ 重构：将Dashboard包管理器从yarn迁移到pnpm（扩展性任务）

完成任务过程

为路由页面增加批量删除功能 [issue#2565](#)

情况介绍：希望添加批量删除路由的操作，很经常都是批量处理路由，但是只能批量导入，不能批量删除，有多选功能，支持批量删除应该很友好

解决方法：我在路由页面的列表添加了此项功能

解决PR: <https://github.com/apache/apisix-dashboard/pull/2502>

为批量删除增加e2e测试覆盖 [issue#2562](#)

情况介绍：批量删除路由的功能需要用cypress e2e测试工具写前端测试用例

解决方法：我在”批量删除“pr里一起添加了，主要使用了cypress，上手挺快的

解决PR: <https://github.com/apache/apisix-dashboard/pull/2502>

提高e2e测试稳定性与修复重要bug issue#2515 && #2615

情况介绍：Dashboard的前端e2e不够稳定经常出错，并且出现了一个影响很多开发者的bug。

解决方法：我在“为插件编辑器添加提示和预装插件模版”pr里一起添加了

解决PR: <https://github.com/apache/apisix-dashboard/pull/2581>

解决仪表板 Docker 实现使用优化 issue#2446

情况介绍：apisix-dashboard登录成功，然后访问 apisix/管理员/标签/路由或 apisix/管理员/标签/消费者，遇到 401 未经授权的错误代码，这意味着仪表板授权的 jwt 令牌无效。最后在社区的帮助下，发现原因是1个以上的仪表板实例。使用多实例时每个实例生成并持有一个 jwt令牌，将导致验证冲突，不符合apisix本身的高可用性

解决方法：优化文档以强调仪表板实例数量不超过1

解决PR: <https://github.com/apache/apisix-dashboard/pull/2472>

为插件编辑器添加提示和预装插件模版 issue#1363

情况介绍：我们需要清楚地告诉用户应该填充数据的哪一部分，我们需要紫色部分

Take route as an example (the use of service is the same method), enable the limit-req plugin on the specified route.

```
curl http://127.0.0.1:9080/apisix/admin/routes/1 -H 'X-API-KEY: edd1c9f834335f136f87ad84b625c8f1' -X PUT -d '{
  "methods": ["GET"],
  "uri": "/index.html",
  "plugins": {
    "limit-req": {
      "rate": 1,
      "burst": 2,
      "rejected_code": 503,
      "key": "remote_addr"
    }
  },
  "upstream": {
    "type": "roundrobin",
    "nodes": {
      "39.97.63.215:80": 1
    }
  }
}
```

You can open dashboard with a browser: <http://127.0.0.1:9080/apisix/dashboard/>, to complete the above operation through the

```
"plugins": {
  "limit-req": {
    "rate": 1,
    "burst": 2,
    "rejected_code": 503,
    "key": "remote_addr"
  }
}
```

解决方案：在相应模态框添加预览的相关提示内容，为编辑器添加预制插件模板

解决PR: <https://github.com/apache/apisix-dashboard/pull/2581>

修复函数命名错误

情况介绍：apisix-dashboard的路由相关功能函数命名有一些错误

解决方法：修复命名

解决PR: <https://github.com/apache/apisix-dashboard/pull/2599>

使Dashboard支持Windows环境

情况介绍：Windows下启动不了，以为是什么高级原因，原来是启动命令掺杂了cp（Linux命令），我当时因为这个非常迷惑，还以为是什么脚本不能在windows环境使用，没想到就个这，影响我好长时间，不过我现在把他解决了。

解决方法：使用node.js来实现复制命令，而不是cp，因为cp也可以在Windows上运行。

解决PR: <https://github.com/apache/apisix-dashboard/pull/2619>

重构:将Dashboard包管理器从yarn迁移到pnpm(扩展性任务) 相关旧pr [#2459](#)

情况介绍：此 PR 的目的是介绍高级和流行的 pnpm 包管理工具，并使用 pnpm 更快地下载依赖项包。

效果：pnpm将成为仪表板的下一代包管理工具。将包管理器从原始 yarn 迁移到 pnpm，以减少依赖项安装和编译时间。

解决方法：package.json、pnpm-lock.yaml、相关的 GitHub 工作流文件。相关文档已更新，已针对使用中可能出现的问题编写了解决方案，并且工作流正常运行。

解决PR: <https://github.com/apache/apisix-dashboard/pull/2604>

项目评估

基本上在做的时候已经考虑的比较完善了，不完善的已经有很多其他维护者的建议来修正。

项目补充

遇到的问题及解决方案

为路由页面增加批量删除功能：

这是第一个遇到的问题，主要是在这个适应阶段我没很快进入适应项目，我去网上查的Dashboard教程都是部署方法！！！而不是教开发者如何开发。以前的经验也用不了，Windows下启动不了，以为是什么高级原因，原来是启动命令掺杂了cp（Linux命令）....我当时不知道从哪来的cp，以为是构建脚本无法在Windows使用，没想到竟然是如此简单的任务....

我浪费了很多时间在其中，结果问了杨导10分钟就解决了，我以后如果去大厂/学术岗，一定要咨询mentor帮我把环境配好，让他介绍一下相关情况，这是最惨痛的教训.....

为插件编辑器添加提示和预装插件模版：

这个我遇到了一些问题，

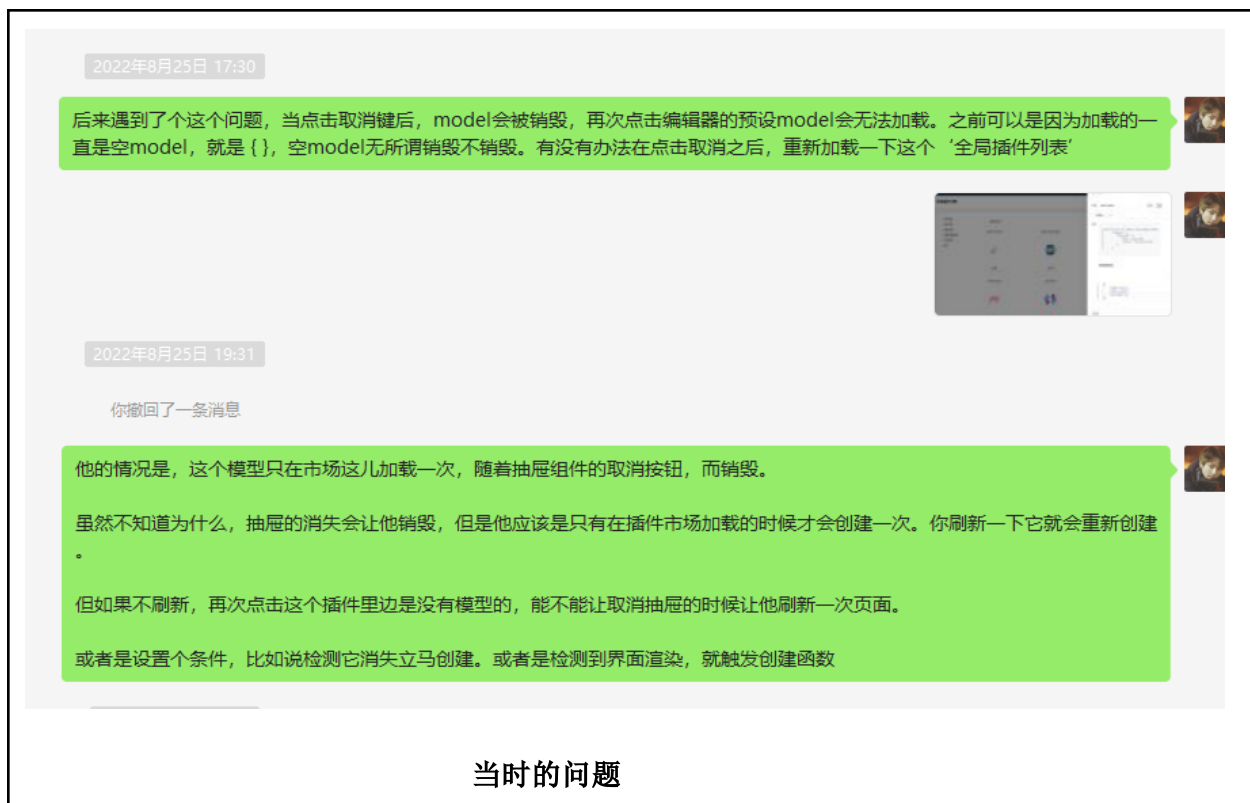
- schema的分别？
- 如何匹配？是否需要上面的函数？需不需要重新写一个匹配？
- Model如何加载，从哪修改？
- json具体如何导入？
- uri和名字用哪个？
- uri如何使用
- schema要不要改？如何改？
- 模型重复如何解决，是否要改模型？
- debug阶段
- 调整使用的方法

.....

Model最后也需要我们销毁，这里分两种情况，假如是通过createModel创建的Model，那么我们需要手动销毁，但是如果是monaco默认创建的，则不需要，在调用实例的销毁方法时，会顺带销毁默认创建的Model。

```
1. model.dispose();
```

我当时查到的图片



我一个一个查找教程，一个一个查找用例，最后也在杨的帮助下完成了。这个让我了解了Dashboard的工作环境，插件用法，和相关文档，对项目的理解加深了很多。

重构:将Dashboard包管理器从yarn迁移到pnpm(扩展性任务)

这个我是在原来的开发者已经做了一些的基础上完成，很多线上环境和线下环境的情况不一样，我在这个过程中学习了很多Github工作流和线上的知识，对Github的工作方式了解更深了。

与导师沟通及反馈情况

一句话：我爱杨陶，我爱杨导❤️❤️😘😘💕💕

沟通很好，帮助很多，我深深体验到大厂培养成本的含义了，确实很多习惯得让人帮助培养，很多细节，很多东西都是技术之外的东西，如果我之前能问更多可能会好很多，但是如果再来一次，我说不定还是会自己琢磨。不过下次就知道没必要了，很多知识没必要了解，赶紧让人带你上路更重要。😂😂😂

项目总结

项目的产出基本符合项目的目标和要求，并且增加了两三项任务，总体上超出计划预期。在未来的一段时间内，我会继续关注APISIX的发展，推动Dashboard的发展，自发的参与到其发展中。期望在未来的一年内成为Apache顶级项目的committer。

这段开源经历对我的职业生涯来说是一个良好的开始，也是一段有价值、值得铭记的经历。不仅会成为我生涯的闪光点，也会成为我人生的加速器，推动我迈入更高的阶梯。
