

调研与选型报告

一、开发环境

1、Windows+Mac 双平台

我们小组中有队友使用 mac 系统开发，同时为了适应不同操作系统下 web 网页端和小程序端，我们在 windows 下和 mac 都进行开发，这样方便对于一些网页显示的调整，而且在 mac 平台下有 sketch 这个界面设计软件，方便我们对界面进行初始化的构建，也便于后面代码的实现。

2、Docker 环境

Docker 是 Docker.Lnc 公司开源的一个基于 LXC 技术之上搭建的 Container 容器引擎，源代码托管在 Github 上，基于 Go 语言并遵从 Apache2.0 协议开源。Docker 属于 Linux 容器的一种封装，提供简单易用的容器使用接口。Docker 将应用程序与该程序的依赖，打包在一个文件里面。运行这个文件，就会生成一个虚拟容器。程序在这个虚拟容器里运行，就好像在真实的物理机上运行一样。有了 Docker，就不用担心环境问题。

总体来说，Docker 的接口相当简单，用户可以方便地创建和使用容器，把自己的应用放入容器。容器还可以进行版本管理、复制、分享、修改，就像管理普通的代码一样。

在我们的项目中，我们使用 Docker 用于后期的联合调试和打包上传服务器，这样我们可以在本地 Docker 环境中配好网站运行的环境，然后打包上传到服务器中，部署 Docker 环境后可以直接在服务器中部署网站，不需要再进行额外配置，也方便了以后环境的迁移和管理。

二、前端

1、编程语言

① JavaScript

Js 是浏览器唯一规定的编程标准，各大浏览器公司支持，有着庞大的社区，有着庞大而且优秀的知名度高的简单易用的第三方库。

主要特点：

- 1) 是一种解释性脚本语言（代码不进行预编译）。
- 2) 主要用来向 HTML（标准通用标记语言下的一个应用）页面添加交互行为。
- 3) 可以直接嵌入 HTML 页面，但写成单独的 js 文件有利于结构和行为的分离。
- 4) 跨平台特性，在绝大多数浏览器的支持下，可以在多种平台下运行（如 Windows、Linux、Mac、Android、iOS 等）。

5) Javascript 脚本语言同其他语言一样，有它自身的基本数据类型，表达式和算术运算符及程序的基本程序框架。Javascript 提供了四种基本的数据类型和两种特殊数据类型用来处理数据和文字。而变量提供存放信息的地方，表达式则可以完成较复杂的信息处理。

② HTML

主要特点：

- 1) 简易性：超级文本标记语言版本升级采用超集方式，从而更加灵活方便。
- 2) 可扩展性：超级文本标记语言的广泛应用带来了加强功能，增加标识符等要求，超级文本标记语言采取子类元素的方式，为系统扩展带来保证。
- 3) 平台无关性：虽然个人计算机大行其道，但使用 MAC 等其他机器的大有人在，超级文本标记语言可以使用在广泛的平台上，这也是万维网（WWW）盛行的另一个原因。
- 4) 通用性：另外，HTML 是网络的通用语言，一种简单、通用的全置标记语言。它允许网页制作人建立文本与图片相结合的复杂页面，这些页面可以被网上任何其他人浏览到，无论使用的是什么类型的电脑或浏览器

2、前端框架

① Vue

vue 是一款友好的、多用途且高性能的 JavaScript 框架，它能够帮助你创建可维护性和可测试性项目的代码库，vue 是渐进式的 JavaScript 框架，也就是说如果你已经有一个现成的服务端应用你可以将 vue 作为该应用的一部分嵌入其中，带来更加丰富的交互体验或者如果你希望将更多的业务逻辑放到前端来实现那么 vue 的核心库机器生态系统也可以满足你的各式需求。

与其他框架相同，vue 允许你讲一个网页分割成可复用的组件，每个组件都包含属于自己的 HTML、CSS、JavaScript 以用来渲染网页中相应的地方。

② Angular

Angular 框架是 Google 公司提供的一套开源项目，是基于 HTML 基础上进行拓展的开发工具，其拓展的目的就是希望能通过 HTML 标签构建动态 Web 应用。这个框架有着诸多特性，最为核心的是：MVVM、模块化、自动化双向数据绑定、语义化标签、依赖注入，以上这些全部都是属于 angular 特性。

③ React

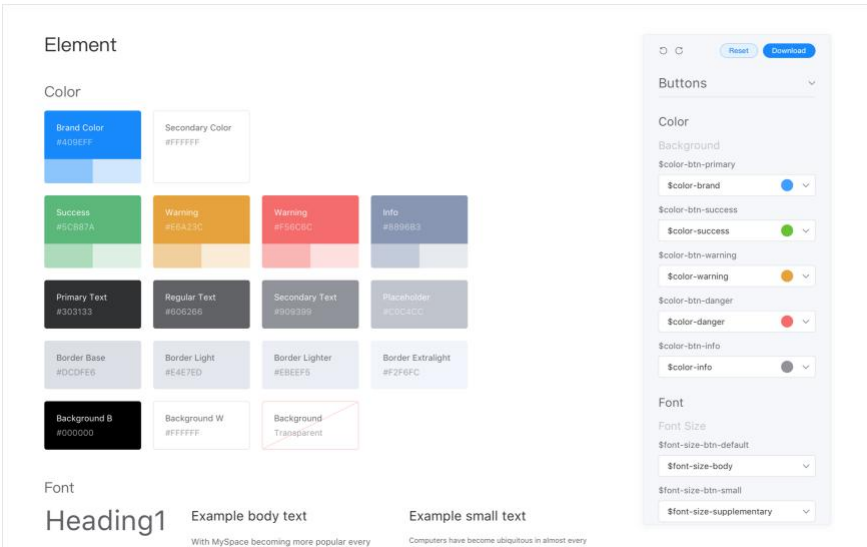
React(有时叫 React.js 或 ReactJS)是一个为数据提供渲染为 HTML 视图的开源 JavaScript 库。React 视图通常采用包含以自定义 HTML 标记规定的其他组件的 组件渲染。React 为程序员提供了一种子组件不能直接影响外层组件("data flows down")的模型,数据改变时对 HTML 文档的有效更新,和现代单页应用中组件之间 干净的分隔。

因为我们小组对 vue 都比较熟悉, 这里我们选用 vue 框架作为我们前端的框架使用。

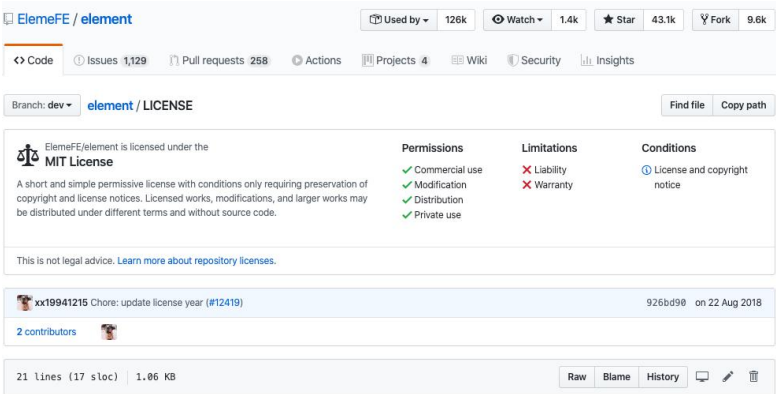
3、前端组件库

① Element-Ui

Element-UI 是饿了么前端团队推出的一款基于 Vue.js 2.0 的桌面端 UI 框架,手机端有对应框架是 Mint UI, 在后面小程序开发的时候也可以使用
官网:



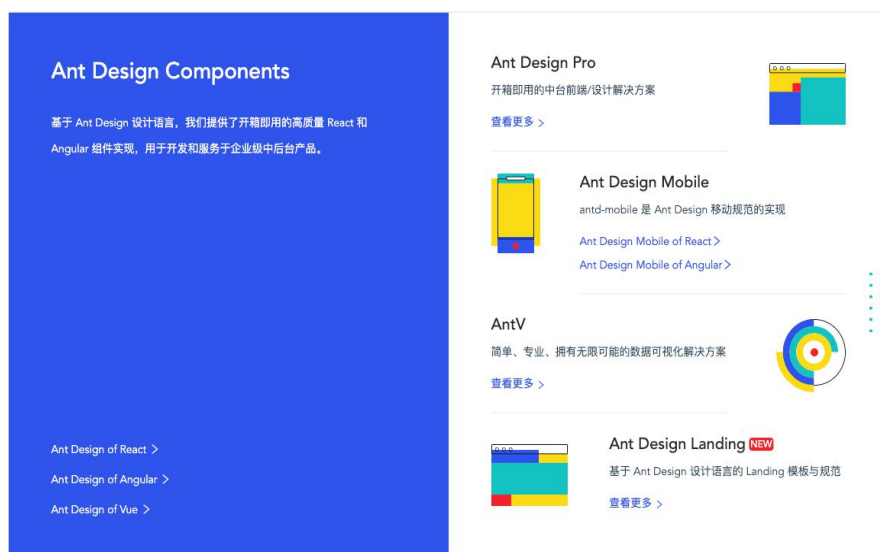
License:



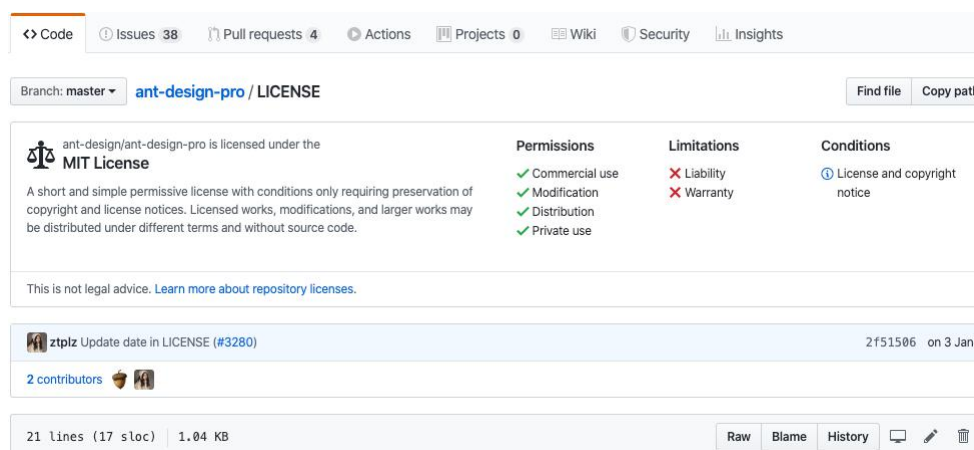
② Ant Designer

这也是一个提供 ui 设计的网站，里面包含了很多 ui 设计的组件库，也包含了网页端和移动端组件，相比于 element-ui 更加丰富。

官网：



License：



Element-ui 是基于 vue 的，使用起来更加方便，本项目使用 element-ui 组件库。

4、前端包管理

Npm：

NPM 是随同 NodeJS 一起安装的包管理工具，能解决 NodeJS 代码部署上的很多问题，常见的使用场景有以下几种：

- ①允许用户从 NPM 服务器下载别人编写的第三方包到本地使用。
- ②允许用户从 NPM 服务器下载并安装别人编写的命令程序到本地使用。
- ③允许用户将自己编写的包或命令程序上传到 NPM 服务器供别人使用。

因为 npm 在前端包管理中广泛使用，且在 GitHub 中数据显示也是更胜一筹，在第一个星期的学习中，我们小组的成员运用 npm 进行包管理，已经有了一定的经验，所以为了高效完成项目前端的开发，我们选择了学习成本更低、效率更高的 npm 进行开发。

5、测试框架

① mocha

mocha 是 JavaScript 的一种单元测试框架，既可以在浏览器环境下运行，也可以在 Node.js 环境下运行。并且 mocha 社区成熟、用的人多，测试各种东西社区都有示例。

使用 mocha，我们就只需要专注于编写单元测试本身，然后，让 mocha 去自动运行所有的测试，并给出测试结果。

mocha 的特点主要有：

- 1) 既可以测试简单的 JavaScript 函数，又可以测试异步代码，因为异步是 JavaScript 的特性之一；
- 2) 可以自动运行所有测试，也可以只运行特定的测试；
- 3) 可以支持 before、after、beforeEach 和 afterEach 来编写初始化代码。

② Jest

Jest 是 Facebook 开源的一款 JS 单元测试框架，基于 Jasmine 至今已经做了大量修改添加了很多特性。

主要特点：

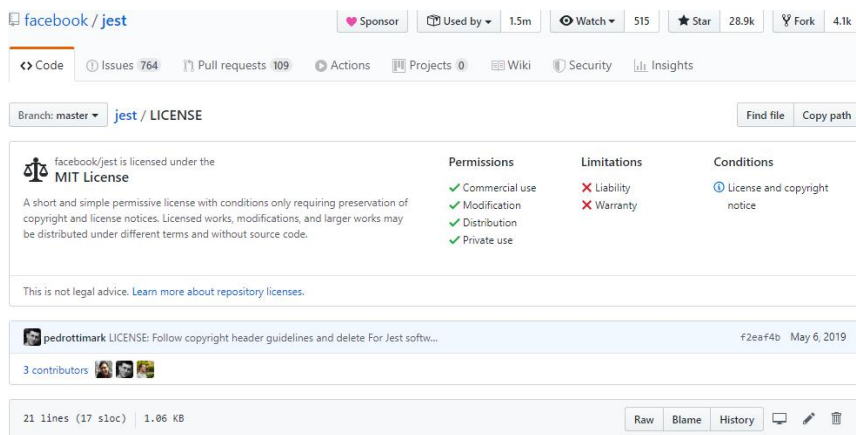
- 1) 基于 Jasmine 至今已经做了大量修改添加了很多特性
- 2) 开箱即用配置少，API 简单
- 3) 支持断言和仿真
- 4) 支持快照测试
- 5) 在隔离环境下测试
- 6) 互动模式选择要测试的模块
- 7) 优雅的测试覆盖率报告，基于 Istanbul
- 8) 智能并行测试(参考)
- 9) 较新，社区不十分成熟
- 10) 全局环境，比如 describe 不需要引入直接用
- 11) 较多用于 React 项目(但广泛支持各种项目)

Jest 比起 mocha 更加方便快捷，且更易使用。所以采用 Jest 作为前端测试框架更合适。

License:

The screenshot shows the GitHub repository page for mocha (mochajs/mocha) under the master branch. The file selected is LICENSE. The page displays the MIT License text, which is a short and simple permissive license. It includes a table with Permissions (Commercial use, Modification, Distribution, Private use) and Limitations (Liability, Warranty). The Conditions section mentions the license and copyright notice. At the bottom, it shows the file was updated by josephlin5555 on Feb 7, 2018, and has 10 contributors. The file size is 1.09 KB and it contains 22 lines of code.

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	① License and copyright notice
✓ Modification	✗ Warranty	
✓ Distribution		
✓ Private use		



6、小程序前端框架

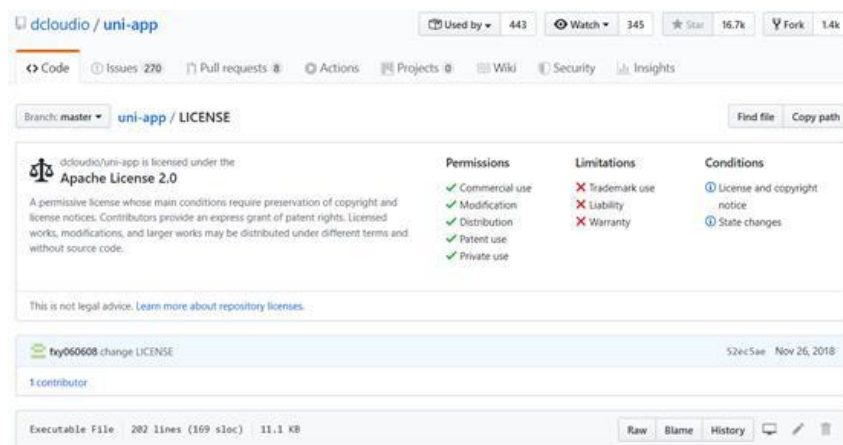
① mpvue

美团团队开源的一款使用 Vue.js 开发微信小程序的前端框架。使用此框架，开发者将得到完整的 Vue.js 开发体验，同时为 H5 和小程序提供了代码复用的能力。mpvue 在发布后的几天间获得 2.7k 的 star, 上升速度飞起, 截至目前为止已经有 13.7k 的 star。

② uni-app

uni-app 是一个使用 Vue.js 开发所有前端应用的框架，开发者编写一套代码，可发布到 iOS、Android、H5、以及各种小程序等多个平台。uni-app 已经是业内最风靡的应用框架，支撑着 6 亿手机用户的庞大生态。

License:



部分小程序框架比较：

部分 app 在各端的兼容性：

平台	微信原生	wepy	mpvue	taro	uni-app	chameleon
微信小程序	○	○	○	○	○	○
支付宝小程序	✗	✗	○	○	○	下拉刷新无法停止
百度小程序	✗	✗	○	○	○	下拉刷新无法停止
头条小程序	✗	✗	○	○	○	✗
H5端	✗	✗	✗	上拉加载/下拉刷新失效	○	上拉加载/下拉刷新失效
App端	✗	✗	✗	上拉加载失效	○	列表无法滚动，无法测试上拉加载/下拉刷新

教学课程方面：

	官方课程	三方课程
wepy	无	无
mpvue	无	腾讯课堂、慕课网、网易云课堂均有三方课程
taro	掘金小册	慕课网有三方课程
uni-app	官方1小时视频	腾讯课堂、网易云课堂较多三方课程。腾讯课堂官方为uni-app录制了课程
chameleon	官方5分钟视频	无

技术支持、社区活跃度：

	官方技术支持	社区活跃度
wepy	官网微信已无法添加	无法入群，无法判断活跃度
mpvue	较少	9个微信群。气氛活跃
taro	较及时	9个微信群。气氛活跃
uni-app	较及时，客服机器人完善。社区沉淀较多经验分享	30多个QQ、微信群，气氛活跃
chameleon	较及时	1个微信群，1个QQ群，气氛活跃

从以上各种数据来看，uni-app 相比于 mpvue 更加实用方便，所以我们决定采用 uni-app 作为小程序前端框架。

三、后端

1、编程语言

Python：

Python 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言，Python 的设计具有很强的可读性，相比其他语言经常使用英文关键字，其他语言的一些标点符号，

它具有比其他语言更有特色语法结构。Python 相比于 C++ 这种语言来说入门更加简单，使用也更加方便。

2、后端框架

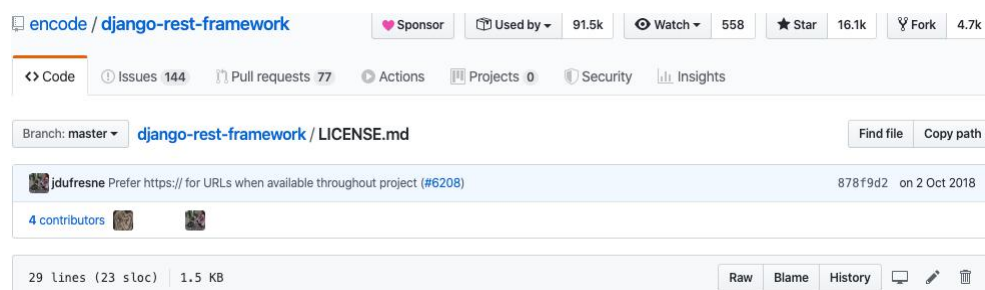
① Django:

Django 是 MVT 设计模式的一种后端框架，相比于轻量级的 Flask，Django 原生提供了众多的功能组件，同时 Django 也提供项目工程管理的自动化脚本工具数据库 ORM 支持（对象关系映射，英语：Object Relational Mapping），Admin 管理站点文件管理，认证权限，session 机制等等功能，让后端开发更加简便快捷。

② Django Rest Framework:

Django REST framework 是一套基于 Django 框架编写 RESTful 风格 API 的组件。相比于 Django，Django REST framework 支持 Serializers（序列化）、Authentication（身份认证）、ViewSet and Routers（视图集和路由）。在实际开发中。可以作为 Django 的辅助框架，帮助我们规范开发和使用，同时支持对一些数据的管理和身份验证。这里对于仓储物流的数据管理，使用序列化的数据访问更加合理，相比于 Django 来说 Django Rest Framework 更适合我们这个项目。

License:



3、包管理工具:

因为后台开发需要安装各种各样的包和依赖，所以我们需要一个合适包管理工具来管理我们的 Python 环境，我们主要有一下两个选择

① pip:

pip 是一个安装和管理 Python 包的工具，python 安装包的工具有 easy_install, setuptools, pip, distribute 等。distribute 是 setuptools 的替代品，而 pip 是 easy_install 的替代品。pip 是首选的 Python 包管理工具，主要用于安装 PyPI 上托管的软件包(第三方库、应用、框架等)。在 Python3 的开发中应该尽量使用 pip。pip 依赖于 pip 仓库，默认为: <http://pypi.python.org/>，只要里面有就可以安装，较为方便。

②conda:

conda 也是 Python 的一个包管理工具，相比于 pip，conda 可以管理更多的环境，可以针对每一个项目进行不同的包管理，而且 conda 有图形界面和命令行两种操作方法，免除了很多命令行命令，而且 conda 自带的环境也包含很多科学计算的库，可以开箱即用，我们项目中因为不涉及这一块，所以我们可以使用 minianaconda(conda 的一个精简版)包管理。

4、测试框架

①Django 单元测试(testcase):

Django 支持 Python 的 unittest，且 Django 框架自带了单元测试工具，利用该工具，我们可以方便地对单元进行错误检查，以提高项目的质量，可以通过测试接口对每一个用例进行测试。

②PyTest

PyTest 是一个成熟的全功能的 Python 测试工具，可以帮助编写更好的程序。支持断言、单元测试。使用 PyTest 可以快速地编写小型的测试，甚至是复杂的功能测试。

Django 自带的单元测试框架使用十分方便，功能也挺全面，对于本项目来说 testcase 可以满足任务要求。

5、前后端参数传递框架

① Ajax

Ajax 全称为“Asynchronous JavaScript and XML”（异步 JavaScript 和 XML），是一种创建交互式网页应用的网页开发技术，通过在后台与服务器进行少量数据交换，AJAX 可以使网页实现异步更新。这意味着可以在不重新加载整个网页的情况下，对网页的某部分进行更新。

优点：

- 1) 最大的优点就是页面无需刷新，在页面内与服务器通信，非常好的用户体验。
- 2) 使用异步的方式与服务器通信，不需要中断操作。
- 3) 可以把以前服务器负担的工作转嫁给客户端，减轻服务器和带宽，可以最大程度减少冗余请求。
- 4) 基于标准化的并被广泛支持的技术，不需要下载插件或者小程序。

缺点：

- 1) AJAX 干掉了 Back 和 History 功能，即对浏览器机制的破坏。

在动态更新页面的情况下，用户无法回到前一个页面状态，因为浏览器仅能记忆历史记录

中的静态页面。一个被完整读入的页面与一个已经被动态修改过的页面之间的差别非常微妙；用户通常会希望单击后退按钮能够取消他们的前一次操作，但是在 Ajax 应用程序中，这将无法实现。

2) 安全问题技术同时也对 IT 企业带来了新的安全威胁，ajax 技术就如同对企业数据建立了一个直接通道。这使得开发者在不经意间会暴露比以前更多的数据和服务器逻辑。ajax 的逻辑可以对客户端的安全扫描技术隐藏起来，允许黑客从远端服务器上建立新的攻击。还有 ajax 也难以避免一些已知的安全弱点，诸如跨站点脚本攻击、SQL 注入攻击和基于 credentials 的安全漏洞等。

3) 对搜索引擎的支持比较弱。如果使用不当，AJAX 会增大网络数据的流量，从而降低整个系统的性能。

②Axios

Axios 是一个基于 promise 的 HTTP 库，可以用在浏览器和 node.js 中。Axios 主要是用于向后台发起请求的，还有在请求中做更多是可控功能。

axios 特点：

- 1) 从浏览器中创建 XMLHttpRequests
- 2) 从 node.js 创建 http 请求
- 3) 支持 Promise API
- 4) 拦截请求和响应（就是有 interceptor）
- 5) 转换请求数据和响应数据
- 6) 取消请求
- 7) 自动转换 JSON 数据
- 8) 客户端支持防御 XSRF

③Fetch

Fetch 号称是 Ajax 的替代品，是在 ES6 出现的，使用了 ES6 中的 promise 对象。Fetch 是基于 promise 设计的。Fetch 的代码结构比起 Ajax 简单多了，参数有点像 JQuery Ajax。但是，一定记住 Fetch 不是 Ajax 的进一步封装，而是原生 js，没有使用 XMLHttpRequest 对象。

优点：

- 1) 符合关注分离，没有将输入、输出和用事件来跟踪的状态混杂在一个对象里
- 2) 更好更方便的写法
- 3) 更加底层，提供的 API 丰富（request, response）
- 4) 脱离了 XHR，是 ES 规范里新的实现方式

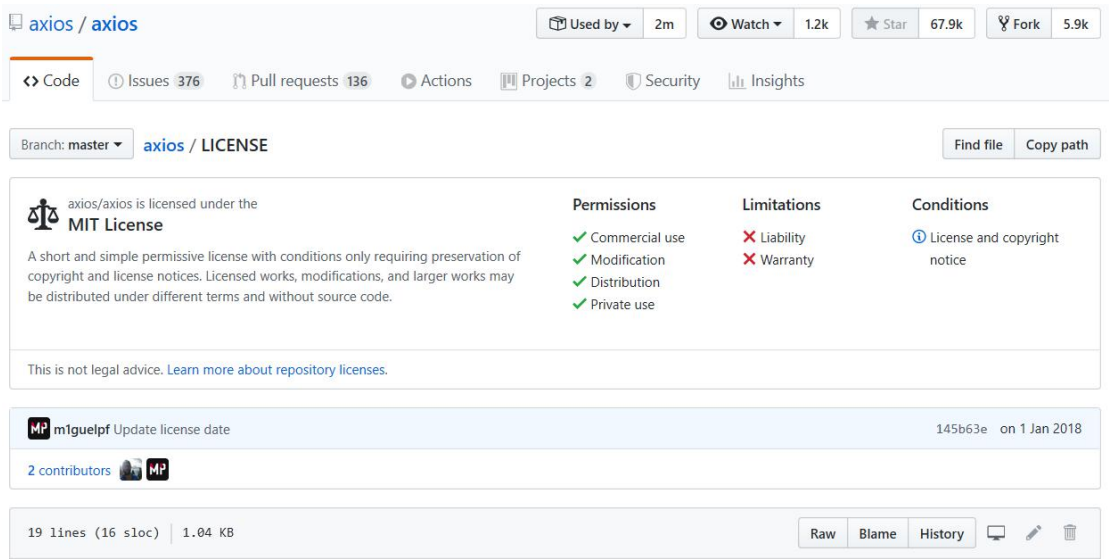
缺点：

- 1) fetch 只对网络请求报错，对 400，500 都当做成功的请求，需要封装去处理
- 2) fetch 默认不会带 cookie，需要添加配置项
- 3) fetch 不支持 abort，不支持超时控制，使用 setTimeout 及 Promise.reject 的实现的超时控制并不能阻止请求过程继续在后台运行，造成了量的浪费
- 4) fetch 没有办法原生监测请求的进度，而 XHR 可以

由于 fetch 在国内使用限制过大，ajax 配置过于繁琐，axios 既提供了并发的封装，

也没有 fetch 的各种问题，而且体积也较小，当之无愧现在最应该选用的请求的方式。

License:



The screenshot shows the GitHub repository page for `axios/axios`. The file `LICENSE` is selected, showing the MIT License text. The license text states: "A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code." The page also displays a table of permissions and limitations, and a commit history section.

Permissions	Limitations	Conditions
✓ Commercial use	✗ Liability	① License and copyright notice
✓ Modification	✗ Warranty	
✓ Distribution		
✓ Private use		

四、团队协作

1、代码版本管理

①SVN:

SVN 是 Subversion 的简称,目前是 Apache 项目底下的一个开放源代码的版本控制系统,它的设计目标就是取代 CVS,SVN 是集中式管理。

1) 优点

- 1、集中式管理，管理方式在服务端配置好，客户端只需要同步提交即可，使用方便，操作简单，很容易就可以上手。
- 2、在服务端统一控制好访问权限，利用代码的安全管理。
- 3、所有的代码已服务端为准，代码一致性高。

2) 缺点

- 1、所有操作都需要通过服务端进行同步，这会导致服务器性能要求比较高。如果服务器宕机了就无法提交代码了。

2、分支管理不灵活，svn 分支是一个完整的目录，且这个目录拥有完整的实际文件，这些操作都是在服务端进行同步的，不是本地化操作，如果要删除分支，也是需要将远程的分支进行删除，这会导致大家都得同步。

3、需要联网。如果无法连接到 SVN 服务器，就无法提交自己的代码，更别说还原、对比等操作了。如果在内网还好，网速比较稳定，同步相对较快，如果是通过外网同步，有可能就需要同步很久。

②GIT:

git 是 Linus Trovalds 大神的作品，是一个开放源码的版本控制软件。与 SVN 最大的区别，就是分布式的管理。

1) 优点

1、分布式开发时，可以 git clone 克隆一个本地版本，然后在本地进行操作提交，本地可以完成一个完整的版本控制。在发布的时候，使用 git push 来推送到远程即可。

2、git 分支的本质是一个指向提交快照的指针，速度快、灵活，分支之间可以任意切换。都可以在本地进行操作可以不同步到远程。

3、冲突解决，多人开发很容易就会出现冲突，可以先 pull 远程到本地，然后在本地合并一下分支，解决好冲突，在 push 到远程即可。

4、离线工作，如果 git 服务器出现问题，也可以在本地进行切换分支的操作，等联网后再提交、合并等操作。

2) 缺点

1、git 没有严格的权限控制，一般是通过系统设置文件的读写权限来做权限控制。

2、工作目录只能是整个目录，而 svn 可以单独 checkout 某个有权限的目录。

3、git 上手可能没有 svn 那边顺手，需要经过学习一下。

总结

由于我们的是分布式，多人开发，版本迭代比较快的项目，所以选择使用 git 进行版本管理

2、代码规范管理:

① 前端

ESLint:

ESLint 是一个以可扩展、每条规则独立的，被设计为完全可配置的 lint 工具，一个 QA 工具，用来作为静态代码检查,避免低级错误和统一代码的风格。

主要有以下特点：

- 1) 默认规则包含所有 JSLint、 JSHint 中存在的规则， 易迁移；
- 2) 规则可配置性高： 可设置「 警告」、「 错误」 两个 error 等级， 或者直接禁用；
- 3) 包含代码风格检测的规则（ 可以丢掉 JSCS 了）；
- 4) 支持插件扩展、 自定义规则。

②后端:

PyLint:

PyLint 是 Python 源代码分析器，可以分析 Python 代码中的错误，查找不符合代码风格标准和有潜在问题的代码，是一个可以用于验证多个文件的模块和包的工具。缺省情况下，PyLint 启用许多规则。它具有高度可配置性，从代码内部处理程序控制它。另外，编写插件添加到自己的检查中是可能的。

PEP8:

PEP8 一开始是一篇关于 Python 编码风格的规范，它提出了保持代码一致性的细节要求，Johann C.Rocholl 开发了一个 PEP8 应用程序来自动检测代码是否符合 PEP8 规范。

