# 调研和选型报告

我们对前端框架,前端样式工具,后端框架,python代码风格检查工具这四个方面进行了调研与选型。在调研过程中,我们搜索查询了许多与我们的项目相关的框架,并对其优缺点和可行性进行了仔细的对比分析,最终,确定了最适合我们本次项目的框架结构。

### (一)前端框架

在经过仔细分析之后我们确定了三种比较适合我们本次项目的可行方案:分别是 Angular, React+Redux 和 vue.js。下面是我们对三种框架的分析,最终确定将 vue.js 作为我们前端的框架。

Angular

GitHub Star 39,370 Fork 9,595

LICENSE PERMITTED

Angular 的最大优势在于它的流行程度。也有人认为它和 Google 密切相关的名字,会影响团队使用它。

是否采用:

不采用理由: Angular 框架着重于在单个页面应用程序中创建用户界面并没有处理构建完整的 web 应用这个更大的关注点,如果不及早确定下来,这将会导致整个项目难以维护,在实际项目中,运行时提供不属于核心框架的技术往往让人觉得不可思议,这大大降低了 TypeScript 对最终开发者的价值。

#### React + Redux

GitHub Star 13,516 Fork 2,141

#### LICENSE PERMITTED

React 和 Redux 的最大优势在于它们相对简单和专注。做一件事情并把它做好是非常困难的,但这两个库都很有效地完成了它们的目标。虽然对于某些状态容器方法可能是外部的,但大多数开发人员还是可以轻松掌握概念,并了解单向数据体系结构的好处,简化大量的用户界面应用程序。

# 是否采用: 🖀

不采用理由:要构建一个功能丰富的 Web 应用程序,需要许多功能,一旦 脱离 React 和 Redux 和其他一些库的核心,你将发现一个非常分散的社区,拥有无数的解决方案和模式,不容易整合在一起。虽然 React 和 Redux 都是非常专注的库,但缺乏经验的团队还是会很容易地生成不可维护的解决方案,而不是意识到他们所做的选择会导致性能不佳或错误。

### Vue.js

GitHub Star 110,432 Fork 15,421

#### LICENSE PERMITTED

渐进式构建能力是 vue.js 最大的优势, vue 有一个简洁而且合理的架构, 使得它易于理解和构建。

# 是否采用:<mark>是</mark>

采用理由:适合传统的 web 应用程序;有强壮稳健的应用程序层;功能足够满足需求;相对轻便;易于学习和上手;LICENSE PERMITTED

(二)前端样式

在前端样式工具的选择上,我们对比了 Element UI 和 Bootstrap 两种工

具包,最终确定将 Element UI 作为我们前端样式的设计工具。

• Element UI

Bootstrap

选择: Element UI

理由:较 Bootstrap 而言, Element UI 更加符合本项目设计理念。同时,

Element UI 较易上手,有详细的方便阅读的操作指南,便于我们小组成员快速

上手开始开发项目。

(三)后端框架

我们在后端方面经过对比分析,确定了两种比较适合我们项目的框架,分

别为 Django, Ruby on Rails 和 Symfony。经过以下分析,我们最终确定了

Django 作为我们的后端选型。

可行方案:

Django

GitHub Star 35,645 Fork 15,212

LICENSE PERMITTED

基于 MVC 的开源框架模式,编程语言为 Python,主要目的是简便、快速

地开发数据库驱动的网站。强调代码复用;有许多功能强大的第三方插件,具

有很强的可扩展性。

是否采用:<mark>是</mark>

### 采用理由:

该框架用于开发数据库驱动网站,符合项目需求;团队成员有相关开发基础;GitHub 星标和 fork 数最多;LICENSE PERMITTED。

## Symfony

Github Star 18,249 Fork 6,351

LICENSE PERMITTED

基于 MVC 模式的面向对象的 PHP5 框架,允许在 web 应用中分离事务控制,服务逻辑和表示层。

是否采用:<mark>否</mark>

否定理由:不熟悉该框架及 PHP ,需要大量时间学习;框架繁重,不适合小型项目。

## • Ruby on Rails

Github Star 40,490 Fork 16,338

LICENSE PERMITTED

网页的全栈开发框架,结合了 PHP 体系的优点(快速开发)和 Java 体系的优点(程序规整),在业内受到广泛关注。

是否采用:

否定理由:成员不熟悉 Ruby 语言,且该框架非常规,可能需要较长的熟悉时间。

## (四) Python 代码检查工具

- Pylint
- Flake8

鉴于两者相似,选择其中之一,Pylint,即可。

### (五)包管理工具

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:

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。 允许用户从 NPM 服务器下载并安装别人编写的命令行程序到本地使用。

允许用户将自己编写的包或命令行程序上传到 NPM 服务器供别人使用。

### (六)前端测试工具

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。

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: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;可以自动运行所有测试,

也可以只运行特定的测试;可以支持 before、after、beforeEach 和 afterEach 来编写初始化代码。