# 系统解决方案概要文档

## 开发运行环境

开发框架：Vue+Django

运行环境：

## 软件体系结构

### B/S架构

B/S架构也叫浏览器/服务器模式。统一客户端，将系统功能实现的核心部分集中到服务器上，简化系统的开发、维护和使用。客户机上只要安装一个浏览器，如Chrome、Safari、Microsoft Edge、Netscape Navigator或Internet Explorer，服务器安装SQL Server、Oracle、MYSQL等数据库。浏览器通过Web Server 同数据库进行数据交互，如下图所示。

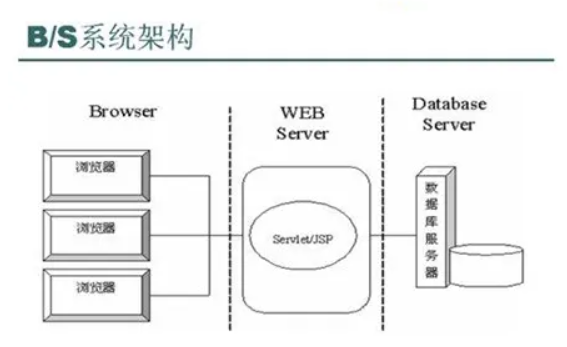


图1 B/S系统架构图

B/S架构相比于C/S架构的优点：

C/S架构软件（即客户机/服务器模式）分为客户机和服务器两层：第一层是在客户机系统上结合了表示与业务逻辑，第二层是通过网络结合了数据库服务器。简单的说就是第一层是用户表示层，第二层是数据库层。需要程序员自己写客户端。B/S 与C/S 的两层架构不同，它采取三层架构。只要有浏览器就可以打开。

### uWsgi+nginx

django是 wsgi web application 的框架，它只有一个简单的单线程 wsgi web server。产品正式上线运行的时候，由于性能需求，通常我们需要高效的 wsgi web server 产品，例如uwsgi，组成一个高效的后端服务。另外，在本次开发中，拟使用nginx代理层，管理服务器静态资源（科研成果附件），减轻django处理请求的压力。

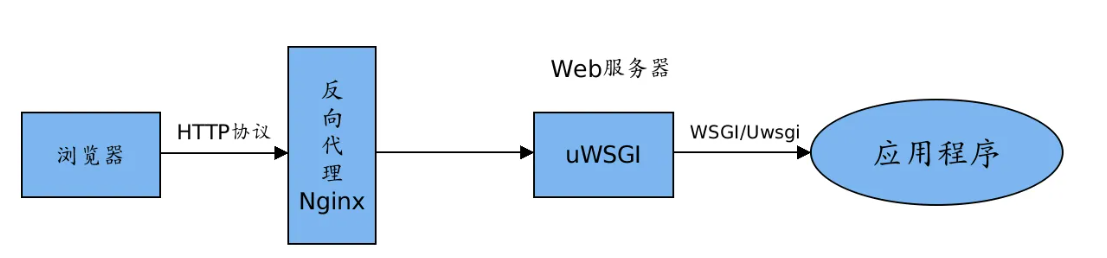


图2 项目流程图

配置方法如下图所示：

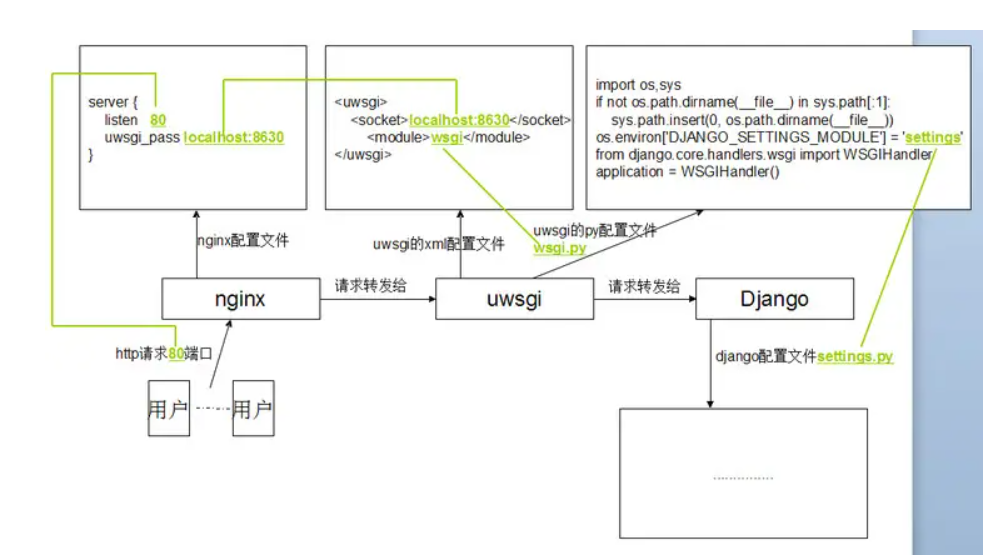


图3 后端服务器配置图

## Vue

Vue是基于MVVM架构模式设计的。MVVM由Model，View，ViewModel三部分构成。

M(Model)：数据模型（Vue的data）

V(View)：视图，即UI，用来展示数据（Vue的el）

ViewModel：是一个对象，用来同步视图View和模型Model。

其结构示意图如下：

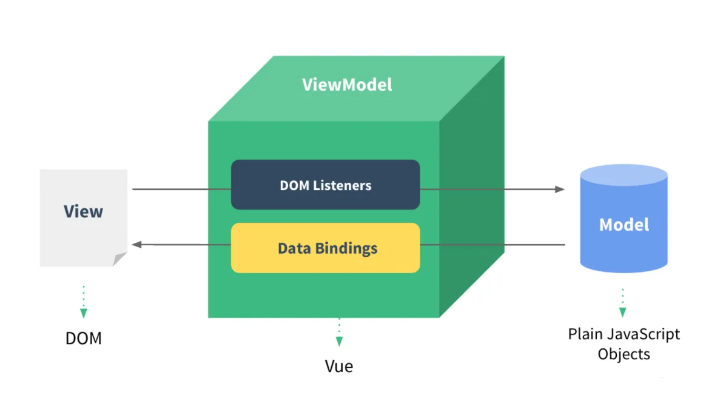


图4 Vue框架结构示意图

## Django

基于django的MTV模型，即M：「Model」 模型，数据管理，T：「Template」 模板，网页展示，V： 「View」 视图，逻辑控制

它的响应模式如下：

1，中间件接收到一个http请求

2，django在url路由系统中查找对应的视图来处理http请求

3，视图函数调用对应的model模型来存储数据、调用相应的模板来展示页面

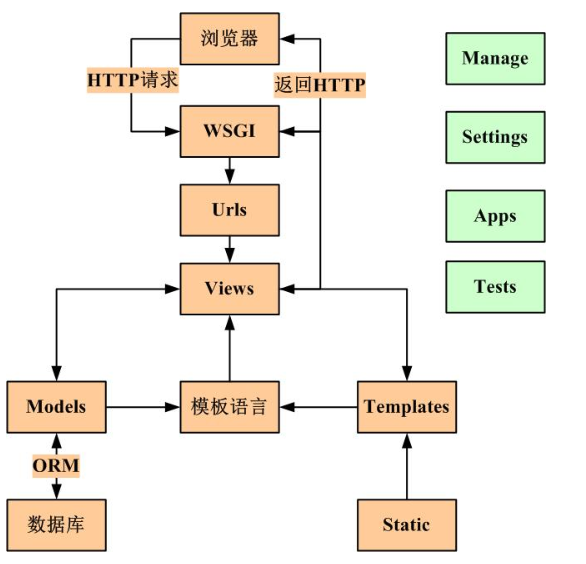


图4 Django的MTV模型

## 关键技术分析

### 科研成果上传和获取

科研成果附件上传到指定服务器地址后，对于静态资源的获取通过nginx转发路由实现，可以减少django程序的运行压力，并且可以对请求统一加请求头，解决跨域和下载问题。

Nginx配置示例如下图所示：



图5 nginx配置

自定义请求头方法如下：

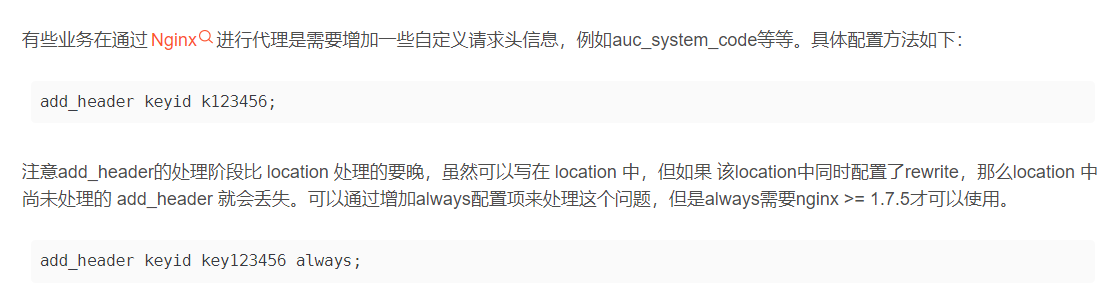


图6 nginx自定义请求头

### 文章附件预览

针对能够在线预览科研成果的附件，我们拟通过在服务器上安装文件预览服务，在需要时调用。

kkfileView预览服务项目指南网址：

<https://kkfileview.keking.cn/zh-cn/docs/production.html>

实现方法：计划在Dcker容器环境下运行服务：先拉取镜像，然后设置访问端口用命令运行，浏览器访问相应容器端口就可以看到项目演示首页。当项目内需要预览文件时，只需要调用浏览器打开本项目的预览接口，并传入须要预览文件的url。

预览效果如下图所示：



图7 文件预览效果

### 推荐算法

系统社交相关功能中“根据科研者的阅读、follow、成果和研究兴趣，提出或者关注的问题等信息推荐科研人员其感兴趣的研究成果、科研人员或者科研问题”，需要根据科研人员的一些兴趣以及行为，在系统中的研究成果、科研人员或者科研问题中都要做一些推荐，首先关键词是推荐，我们需要一个能够契合当前系统的推荐算法，这里我们对于研究成果和科研人员的推荐使用基于标签的推荐。

基于标签的推荐可以解决系统的冷启动问题，即一个系统刚刚开始运行，不知道用户喜欢什么东西，可以根据从用户获得的信息以及用户自愿填写的标签等，知晓用户的一些喜好，从而提供与用户标签最契合的物品标签进行推荐。具体一点可能是用向量点积等方式获得一个分数，最后按分数进行排序，拿到分数最高的几个研究成果、科研人员进行推荐。这其中应当设计好数据库的存取方案与该算法的融合，避免时间复杂度过大。

我们可以根据其阅读科研成果的标签定位该用户最感兴趣的研究方向，从而对研究成果进行推荐；可以对自己以及其follow人员的标签对感兴趣的科研人员进行推荐。

对于科研问题，我们可能没有最直接的标签，也可以文本内容进行提炼，利用包括但不限于TF-IDF算法进行标签的获取，然后找到最契合标签的问题进行推荐。其中TF-IDF算法是一种用于信息检索与文本挖掘的常用加权技术。TF-IDF是一种统计方法，用以评估一个字/词对于一个文档集或一个语料库中的其中一份文档的重要程度。字词的重要性随着它在文档中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。简单的解释为，一个单词在一个文档中出现次数很多，同时在其他文档中出现此时较少，那么我们认为这个单词对该文档是非常重要的。 解决方法可以用Sklearn、Gensim等库进行编码，需要对机器学习的知识有一些了解。

## 非功能需求

### 浏览器兼容需求

所谓的浏览器兼容性问题,是指因为不同的浏览器对同一段代码有不同的解析,造成页面显示效果不统一的情况。

科研交流平台的主要使用场景是个人电脑，一般安装的浏览器有： Chrome（最新稳定版，94.0.4606.54） 、Safari（最新稳定版，14.1.2） 、Edge（使用chromium内核，最新稳定版，MacOS：92.0.902.84，Windows：94.0.992.31） 、Firefox（最新稳定版，92.0） 对于不在上述浏览器列表中的浏览器及版本，不保证网页效果、功能完全适配。对于过早版本或其他浏览器，引导用户切换到上述浏览器。

上述浏览器中，Chrome适配优先级最高，开发中首先保证Chrome的适配。由于各大浏览器会有自身的默认样式，并且不尽相同，所以为了尽可能的保证样式的统一性，前端在开发项目之前需要先进行样式格式化。

浏览器兼容解决方案参考：<https://blog.csdn.net/weixin_43638968/article/details/109177674>

### 系统可靠性需求

1. 数据库备份：任何数据库都需要备份，备份数据是维护数据库必不可少的操作。备份就是为了防止原数据丢失，保证数据的安全。当数据库因为某些原因造成部分或者全部数据丢失后，备份文件可以帮我们找回丢失的数据。此外也有很多非数据丢失的应用场景：特殊应用场景下基于时间点的数据恢复、开发测试环境数据库搭建、相同数据库的新环境搭建、数据库或者数据迁移
2. 确保响应性能。可以通过优化后端算法，提高系统性能，减少响应时长。也可以通过压缩数据传输量，减少带宽占用，能同时处理更多请求，响应速度也更快。

### 网络安全需求

1. 前后端接口要求设置合理，后端接收数据时需要进行数据检验。例如修改密码接口不可以仅仅通过传入用户名和新密码进行处理，还需要进行权限校验，否则可以通过伪造请求修改他人密码。再例如对于数据查询请求，需要避免传入sql关键字获取用户不该拿到的数据，防止sql注入。Sql注入一般发生在拼接sql语句的场景，从而对数据库进行越权访问或其他危险操作，危害极其严重，使用django的orm查询可以有效避免sql注入问题，也可以通过参数化查询和模糊查询处理请求。
2. Url加密，对于获取静态资源的直接网址需要进行加密，以免将服务器的所有静态文件访问地址直接暴露在url上，被越权访问。