基于星火认知大模型的留学文书助手

姓名：邹翔 学号： 2020210120 班级： 2020211101

本项目通过星火大模型提供的开放API，将大模型成功接入到自己的网页中，为用户提供文书写作的指导，为没有留学文书写作经验的用户提供帮助。

1.1 总体设计

本项目基于Vue框架构建，通过WebSocket建立前后端通信。大模型API选用星火认知大型V3.1。

1.1.1 整体框架

整体框架如图1-1所示。



图1-1 整体框架

1.1.2 系统流程

系统流程如图1-2所示。包括用户选择面谱、输入问题、对话式回显、上下文联系以及开启新对话。



图1-2 系统流程

1.2 开发环境

本部分包括node.js安装、Vue.js(2.x)、大模型API申请。

1.2.1 安装node.js

Node.js 安装包及源码下载地址为：<https://nodejs.org/zh-cn/download/，如图1-3>所示。



图1-3 下载Node.js 安装包及源码

运行安装包界面如图1-4所示。

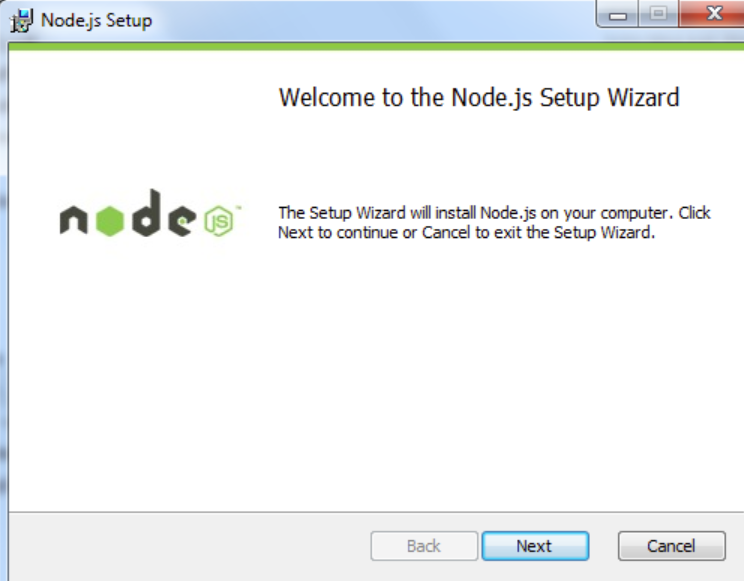


图1-4 运行安装包

选择接受协议选项，单击Next 按钮，如图1-5所示。

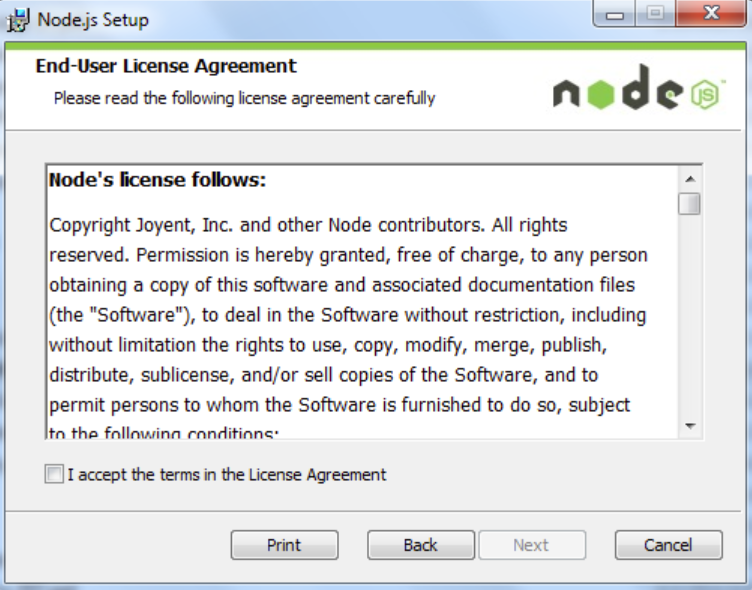


图1-5 接受协议选项

Node.js默认安装目录为 "C:\Program Files\nodejs\" , 也可以修改目录，并单击Next，如图1-6所示。

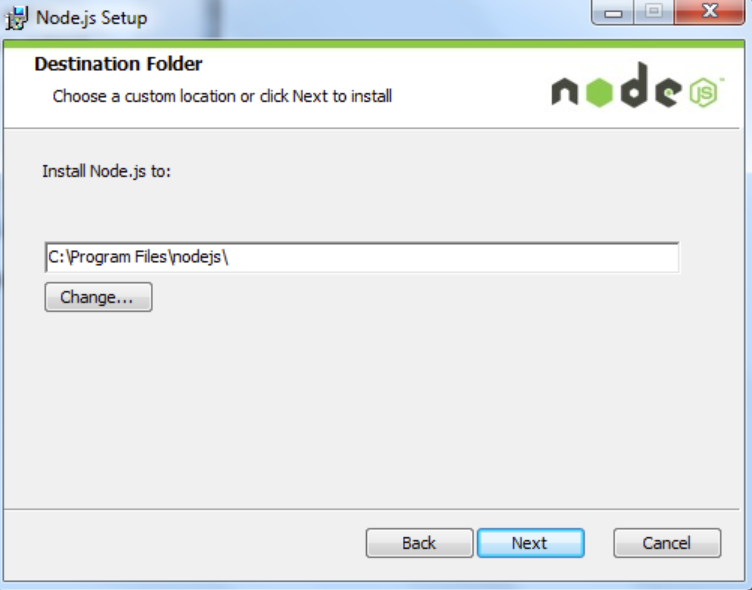


图1-6 安装目录

单击树形图标选择需要的安装模式 , 然后单击Next，如图1-7所示。

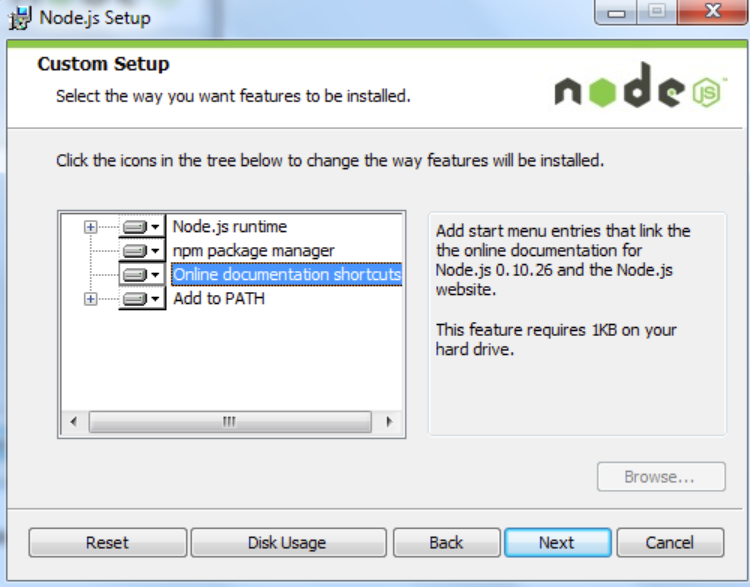


图1-7 选择安装模式

单击 Install开始安装Node.js，也可以单击 Back修改先前的配置，然后单击Next，如图1-8所示。

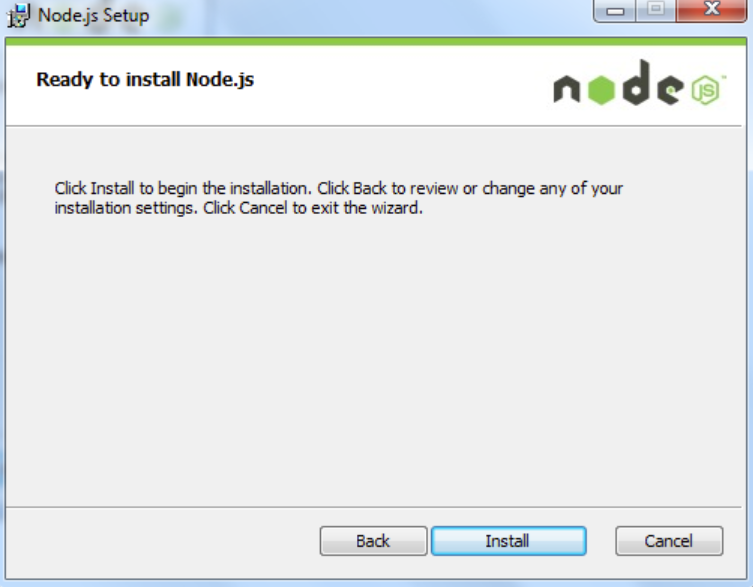


图1-8 安装Node.js

安装过程如图1-9所示。

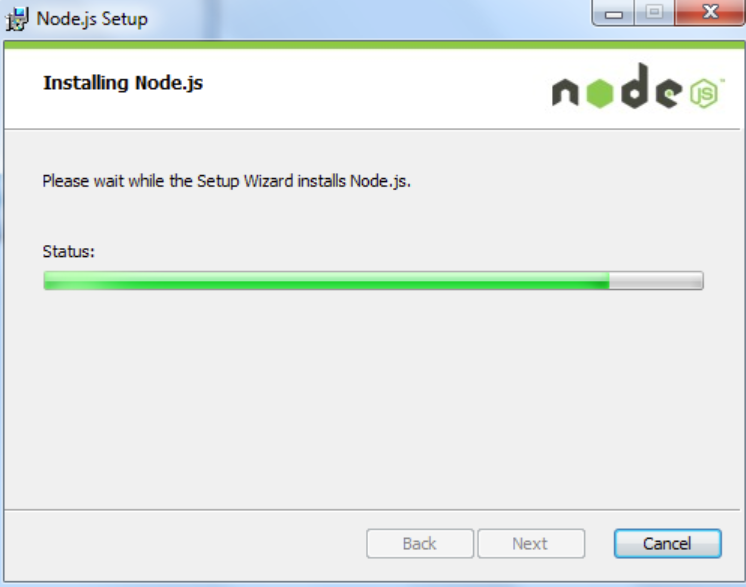


图1-9 安装过程

单击 Finish按钮退出安装向导，如图1-10所示。

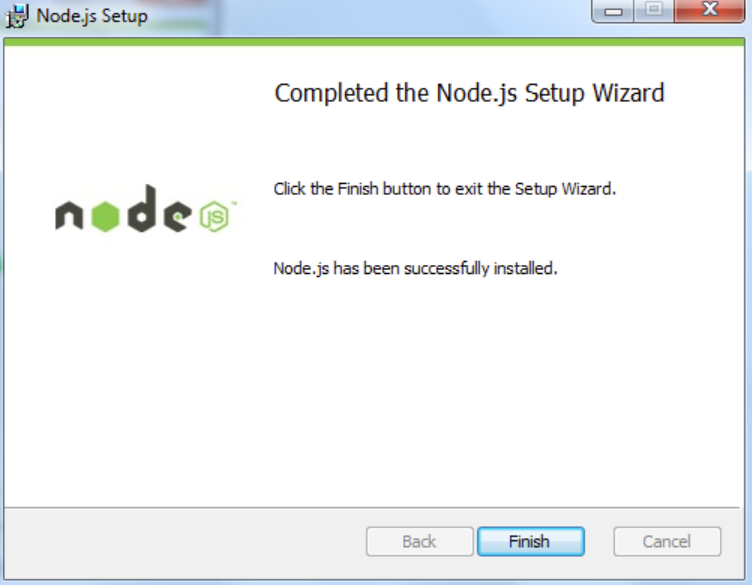


图1-10 安装向导

1.2.2 安装Vue.js（本项目使用Vue2）

打开CMD，输入指令”npm install vue -g”，如图1-11-1所示。

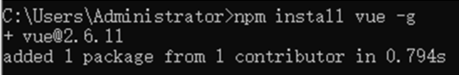


图1-11-1 安装Vue

输入指令”npm info vue”检查确认安装完成，如图1-11-2所示。

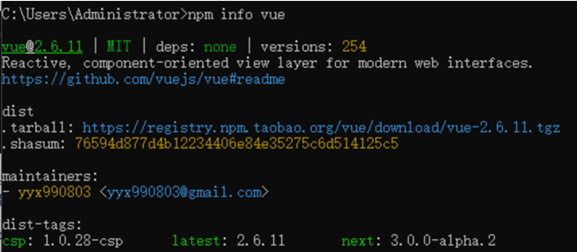


图1-11-2 检查确认Vue安装完成

1.2.3 大模型API申请

注册讯飞认知星火大模型账号，链接如下：<https://xinghuo.xfyun.cn/>，如图1-12所示。



图1-12 讯飞注册页面

注册界面如图1-13所示。



图1-13 注册界面

单击API测试，如图1-14所示。



图1-14 星火认知大模型首页

单击API测试申请，如图1-15所示。



图1-15 API测试申请界面

填写申请工单信息，如果没有创建应用，先重建应用然后填写对应的APP ID，如图1-16所示。



图1-16 申请工单界面

等待工单处理完成，如图1-17所示。



图1-17 工单列表

申请通过之后，可单击左侧项目栏中的星火认知大模型v1.5，查看服务接口认证信息。将APP ID、APISecret和APIKey填写到main.js的requestObj变量中，如图1-18所示。

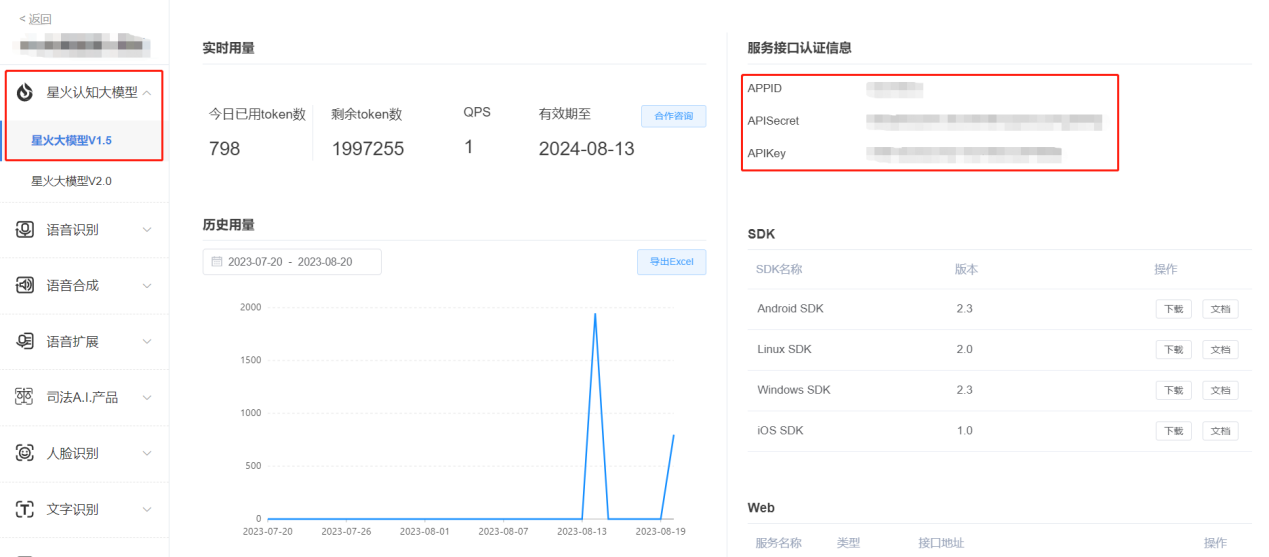


图1-18 星火认知大模型服务信息界面

1.3 系统实现

本项目主要结构，如图1-19所示。

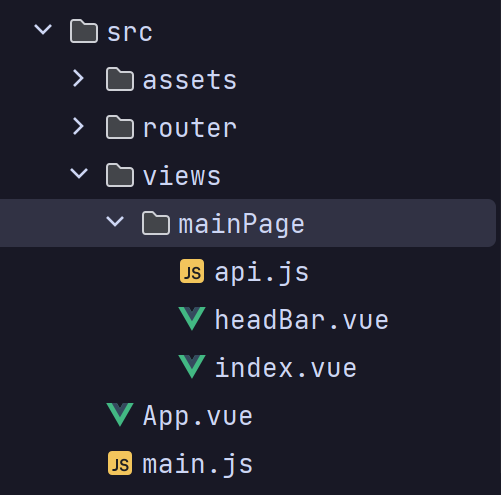


图1-19 本项目主要结构

1.3.1 api.js

主要用于生成鉴权URL

export const getWebsocketUrl = () => {  
 return new Promise((resovle, reject) => {  
 let url = "wss://spark-api.xf-yun.com/v3.1/chat";  
 let host = "spark-api.xf-yun.com";  
 let apiKeyName = "api\_key";  
 let date = new Date().toGMTString();  
 let algorithm = "hmac-sha256"  
 let headers = "host date request-line";  
 let signatureOrigin = `host: ${host}\ndate: ${date}\nGET /v3.1/chat HTTP/1.1`;  
 let signatureSha = CryptoJs.HmacSHA256(signatureOrigin, requestObj.APISecret);  
 let signature = CryptoJs.enc.Base64.stringify(signatureSha);  
  
 let authorizationOrigin = `${apiKeyName}="${requestObj.APIKey}", algorithm="${algorithm}", headers="${headers}", signature="${signature}"`;  
  
 let authorization = base64.encode(authorizationOrigin);  
  
 // 将空格编码  
 url = `${url}?authorization=${authorization}&date=${encodeURI(date)}&host=${host}`;  
  
 resovle(url)  
 })  
}

1.3.2 headBar.vue

头部栏，包含基础功能：作者联系方式以及开启新聊天。

<template>  
 <div class="head-bar">

//个人信息展示  
 <el-button type="text" @click="dialogVisible = true">Contact Me</el-button>  
 <el-dialog  
 title="Contact Me"  
 :visible.sync="dialogVisible"  
 width="30%"  
 >  
 <div class="contact-items">  
 <span>Zoe</span>  
 <span>2020210120</span>  
 <span>Github:https://github.com/Zouu-X</span>  
 </div>  
 <span slot="footer" class="dialog-footer">  
 <el-button @click="dialogVisible = false">取 消</el-button>  
 <el-button type="primary" @click="dialogVisible = false">确 定</el-button>  
 </span>  
 </el-dialog>  
 <span>留学文书小助手</span>

//新聊天的按钮  
 <el-button class="style-change" @click="newChatEvent">New Chat</el-button>  
 </div>  
</template>

1.3.3 index.vue

本项目的主体部分，负责面谱选择、接收输入，并回显输入输出。

<template>  
 <main>  
 <div class="main-body">

//聊天框部分  
 <div class="chat-container">  
 <div class="chat-messages">  
 <div  
 v-for="(message, index) in messages"  
 :key="index"  
 class="msg-group"  
 v-if="!message.invisible" //部分信息不可见  
 >  
 <div class="msg-head"> //判断消息是来自于哪方  
 <span v-if="message.role === 'user'">user</span>  
 <span v-else-if="message.role === 'assistant'">AI</span>  
 </div>  
 <div  
 class="message"  
 :class="{ 'my-message' : message.role === 'user'}"  
 >  
 {{ message.content }}  
 </div>  
 </div>  
 </div>  
 </div>

//输入框  
 <el-input  
 style="width: 650px; margin-top: 20px;"  
 type="text"  
 placeholder="请输入内容"  
 v-model="textInput"  
 @keyup.enter.native="Submit"  
 >  
 <el-button slot="append" icon="el-icon-thumb" @click="Submit" />  
 </el-input>  
 <div class="mod-choose">

//面谱选择  
 <el-radio-group  
 v-model="radio"  
 @input="inputListener"  
 style="margin-top: 30px;"  
 >  
 <el-radio-button label="全部文书"></el-radio-button>  
 <el-radio-button label="简历"></el-radio-button>  
 <el-radio-button label="动机信"></el-radio-button>  
 <el-radio-button label="推荐信"></el-radio-button>  
 </el-radio-group>  
 <el-button class="mod-button" @click="modConfirm">确认</el-button>  
 </div>  
  
 </div>  
 </main>  
  
</template>

1.3.4 App.vue

整合组件后的完整启动文件，网页的整体呈现。

<template>  
 <div class="main" id="mainPage">  
 <head-bar @newChatEvent="handleChatEvent" />  
 <main-page :goNewChat="goNewChat" />  
 </div>  
</template>

1.4 功能测试

启动指令：npm run serve。

初始页面，如图1-20所示。



图1-20 初始页面

选择面谱：，如图1-21所示。



图1-21 面谱选择

输入输出回显，如图1-22所示。



图1-21 输入输出回显

1.5 问题及解决

1.不知道该接口要传入的参数。

阅读官方接口文档：<https://www.xfyun.cn/doc/spark/Web.html>。

2.面谱设计组件选择困难。

最后选用最成熟的Element UI。

3. 颜色搭配困难。

查阅北邮官网，使用北邮官方配色。