

本项目基于 HTML 结构内容,使用 CSS 进行样式设计、应用 JavaScript 建立执行逻辑,依靠百度智能云千帆大模型调用开放的 API,实现生成戏文的功能。

3.1 总体设计

本部分包括整体框架和系统流程。

3.1.1 整体框架

整体框架如图 3-1 所示。





教学资源

3.1.2 系统流程

系统流程如图 3-2 所示。



3.2 开发环境

本节主要介绍 VS Code、Node. js 和 Live Server 插件的安装过程,给出所需要的依赖环 境配置,并介绍大模型 API 的申请步骤。

3.2.1 安装 VS Code

安装 VS Code 参见 2.2.1 节。

除此以外,项目开发还需要安装 Live Server 插件,步骤如下:①运行 VS Code,单击"扩展"(5)按钮,如图 3-3 所示;②在搜索栏中输入 Live Server 进行安装,如图 3-4 所示。

3.2.2 安装 Node. js

安装 Node. js 参见 2.2.2 节。



图 3-3 运行 VS Code



图 3-4 安装 Live Server

3.2.3 环境配置

项目框架包括 LayUI、BootStrap 和 jQuery,均安装在本地./asset/lib 目录下。

(1) LayUI 遵循原生 HTML/CSS/JS 的书写与组织形式,属于轻量级框架,简单美观。 它适用于开发后端模式,在服务器端界面上有非常好的效果。

(2) LayUI 可以在官网首页或更新日志界面下载,如图 3-5 所示。

开始使用 Getting Started							
底层方法 Base Function	v2.9.3 2023-12-31						
模块系统 Modules ●	2023 收官。						
更新日志 Versions ●	• form						
布局	• 优化 input 组件圆角时后缀存在方框的问题 #1467 @bxjt123						
	• 优化 select 搜索面板打开逻辑,以适配文字直接粘贴触发搜索的情况#1498 @Sight-wcg						
11214 Layout	• table						
栅格 Grid	• 修复非常规列设置 field 表头选项时,导出 excel 出现合计行措位的问题 #1522 @Sight-wcg						
通用	treeTable						
新分 Color ●	• 修复 treeTable.updateNode() 更新已打开的节点时,未重新渲染表单元素的问题 #1468 @Sight-wcg						
	laydate						
按钮 Button ●	• 修复开启 rangeLinked 时,选择同一日期,某些情况时间会反选的问题#1517@Sight-wcg						
图标 Icon • upload							
	•加强 data 选项的函数写法,支持获取文件相关数据#1494@Sight-wcg						
	transfer						
表单	• 新增 dblclick 回调函数选项 #1491 @Sight-wcg						
表单组件 Form ●	・移动端的适配						
输入框 Input / Textarea	• 修复 rate, carousel —些小问题 #1469 @Sight-WCg						
选择框 Select	下载: layui-v2.9.3.zip						
复选框 Checkbox	v2.9.2 2023-12-18						

图 3-5 下载 LayUI

(3) LayUI 还可以在命令提示符(command, cmd)终端中通过 npm 指令下载或者使用 第三方内容分发网络(Content Delivery Network, CDN)方式引入,详情如下。

```
<!--引入 layui.css-->
<link href = "//unpkg.com/layui@2.9.3/dist/css/layui.css"rel = "stylesheet">
<!--引入 layui.js-->
< script src = "//unpkg.com/layui@2.9.3/dist/layui.js"></script>
```

BootStrap 是由 Twitter 公司开发的前端框架,用于快速开发 Web 应用程序和网站。 它使用的是基于 HTML、CSS 和 JavaScript 提供一系列组件,包括表单、按钮、导航等。同时,BootStrap 还有一套基于栅格系统的响应式设计,可以在不同设备上呈现出最佳效果。

jQuery 是一个库,包括 HTML 元素的选取、事件处理和 Ajax 操作。使用 jQuery 可以 更容易编写可维护的 JavaScript 代码,同时提高跨浏览器的兼容性。

3.2.4 大模型 API 申请

百度智能云千帆大模型 API 申请参见 1.2.4 节。

3.3 系统实现

本项目使用 VS Code 开发环境搭建 Web 项目,文件结构如图 3-6 所示。



图 3-6 文件结构

3.3.1 头部< head >

定义文档字符和设置网页样式的相关代码如下。

3.3.2 样式< style >

定义网页样式的具体描述及相关代码见"代码文件 3-1"。

3.3.3 主体< body>

设置网页主体的相关代码如下。

```
< center >
   < div class = "box"><!-- 包含左右两个子盒子和其他元素的容器 -->

 <div class = "left"></div><!--样式定义在 CSS 中,位于左侧两个具有不同背景图像的盒</li>

子-->
     <div class = "right"></div><!-- 位于右侧的盒子 -->
     <!-- < img src = "../assets/images/同光.png"alt = ""class = "front_img"/>-->

    < div class = "front img"><!-- 包含一个 iframe 元素,用于显示京剧介绍, iframe 的 src 属</li>

性指向 introduce. html 文件 -->
       < iframe name = "fm" src = "./introduce. html" frameborder = "0" style = "width: 100%;</pre>
height: 100 %; "></iframe>
     </div>
     <div class = "content"><!-- 主要内容区域包含网页的标题、键盘控制区域、输入表单和生成
按钮 -->
       <h1 style = "text - align: center; font - size: 40px;">戏文生成系统</h1>
       < div class = "keyboard"><! -- 包含描述锣鼓点键盘的文本, 一个用于播放音频的 div
(<div id = "Audio"></div>),另一个用于显示键盘(<div id = "Keyboard"></div>)-->
         <锣鼓点键盘:</p>
         < div id = "Audio"></div>
         < div id = "Keyboard"></div>
       </div>
       <div class = "top"><!-- 包含输入问题、选择框的区域及用于生成文本的按钮 -->
         <div style = "display: flex"><!-- 使用 Flex 布局,将子元素横向排列 -->
           < div class = "problem">我想牛成一段京剧</div>
           < div class = "select" style = "margin - right: 20px">
             < select name = "" id = "select">
              < option value = "">下拉选择</option>
              < option value = "慢板">慢板</option>
              < option value = "原版">原版</option>
              <option value = "流水">流水</option>
              <option value = "快板">快板</option>
             </select>
           </div>
           < div class = "message">
             表达大意为:
             < input type = "text" id = "content" />
           </div>
         </div>
         < div class = "btn">
           <br/>
<br/>
button class = "btn" onclick = "requestMessage()">生成</button>
<!-- 单击时调用名为 requestMessage 的 JavaScript 函数 -->
         </div >
       </div>
        < div style = "border: 1px solid # aaa; padding: 10px; font - size: 25px;" id =</pre>
"resultContent">
<!-- 用于显示生成文本结果的区域,边框、内边距和字体大小的样式,初始文本在这里产生-->
         回答将在这里产生...
       </div>
     </div>
   </div>
 </center>
```

3.3.4 主体< body>脚本

处理 RequestMessage 的步骤如下:①获取用户在网页中输入的值,包括一个文本输入 框的值和一个下拉选择框的值;②构建一个包含用户问题的数据对象,将问题的开头、选择 框的值和输入框的值组合在一起。如果用户输入的值为空,函数会立即返回,以确保不会发 送空的请求;③创建一个新的 div 元素,用于显示加载中的提示信息,并将这个元素添加到 界面的特定区域(ID 为 ResultContent 的元素);④函数调用名为 Request 的其他函数,将 用户问题的数据对象转换为 JSON 字符串,并通过某种方式将该请求发送到后端进行处理。 相关代码如下。

```
function RequestMessage() {
```

```
var content = document.getElementById('content').value;
      var select = document.getElementById('select').value;
      var problem = "我想生成一段京剧";
      if (!content) return;
      var loadingDiv = document.createElement('div');
      loadingDiv.innerText = '加载中...';
      document.getElementById('resultContent').innerHTML = '';
      document.getElementById('resultContent').appendChild(loadingDiv);
      var data = {
        messages: [
          {
            role: "user",
            content: problem + select + "表达大意为" + content,
          }.
        ],
      };
      request(JSON.stringify(data));
}
```

定义一个名为 Request 的函数,用于向百度 AI 开放平台发送请求。步骤如下:①得到 访问命令 access_token;②构建请求 URL,其中包含 API 的地址、版本号、自定义的 AI 工 作室、聊天补全资源及访问命令;③通过 Axios 库发送 HTTP 请求,使用 POST 方法将用 户的问题数据发送到服务器端。发送成功时,在响应中提取聊天结果,并替换其中换行符为 HTML 的标签;④将处理后的结果插入 HTML 界面的指定元素中。在请求失败时,会输 出错误信息到控制台。相关代码如下。

```
function request(data) {
    var access_token = "24.b2323fce8b9a9c9c2e1a86826d0a0765.2592000.1706840780.282335 -
42303922";
    var url = `https://aip. baidubce. com/rpc/2. 0/ai_custom/v1/wenxinworkshop/chat/
completions?access_token = $ {access_token}`;
    axios({
        url,
        data,
        method: "post",
        headers: {
            "Content - Type": "application/json",
        }
    }
}
```

```
},
},
}).then(function(response) {
    var resultContent = response.data.result.replace(/\n/g, "<br>");
    document.getElementById('resultContent').innerHTML = resultContent;
}).catch(function(error) {
    console.error('Error:', error);
});
}
```

API在线调试平台如图 3-7 所示。

自助服务	🖨 专业工程间帮您实现AI股据转换!		
自助工具中心 大模型专区 へ	人工智能 AI 语音技术 文字识别 人脸识别	制 人体分析 岡像识别 岡像院家	(2) 建設用 副像環想与特效 内容容板 自然局有处理 机器模器 GayOL 繁荣的作平台 <u>千代大規型平台</u>
示例样板间 HOT	千帆大模型 产品详情	ERNIE-Bot-4	技术文档 示例代码 调试结果 对话询试 云上部署
示例代码	请输入 Q	鉴权参数 ③ 获取鉴权参数	
Token计算器	~ 鉴权认证机制	ak_sk access_token	ERNIE-Bot-4 自国生 XX ~
示例代码中心 ^	硬取AccessToken ERNIE-Bot-4	应用列表: ③	
人工智能 AI	ERNIE-Bot-8K	戏文生成 🗸	
Reterior BOS	ERNIE-Bot	+ client_id:③ ℃	
ANIKIT IN DOG	ERNIE-Bot-turbo-Al原生应用	VW8Vy3****PnaLbd	
内容分发网络 CDN	BLOOMZ-7B	* client_secret: ⑦ □	C
简单消息服务 SMS	Qianfan-BLOOMZ-7B-compre	g7qWlo****y8huP8	
音视频处理 MCP	> Llama-2 Qianfan-Chinese-Llama-2-7B	Header Content-Type:	8
视频创作分发平台	Qianfan-Chinese-Llama-2-13B ChatGLM2-68-32K	application/json	文本翻译 FAQ 百科问答情感分析 阅读理解 内容创作 代码编写
媒体内容分析 MCA	XuanYuan-708-Chat-4bit	Body	游船入 🛅 tokent 甘語 🌘
音视频直播 LSS	ChatLaw	* messages: ()	

图 3-7 API 在线调试平台

3.3.5 其他界面设计

主界面设计成一个包含链接和图像的网页,用户可以通过单击链接跳转到新的界面。 网页布局分为左侧和右侧两部分。主要内容如下。①左侧有顶部、中部和底部3个区域。 顶部区域有1个链接,打开此链接后可以看到图像、标题和文本;中部区域还可以细分为两 个区域,1个是创作的相关链接,1个是分类的相关链接;底部区域包括多个图像;②右侧 有多个区域,每个区域都有1个图像。相关代码如下。

```
function AddAudio() {
    var lenkb = drumconfig.Keyboard.length
    var audiostr = ""
    for (var i = 0; i < lenkb; i++) {
        var filename = drumconfig.Filename[i]
        /*编写鼓音频的 HTML 文档 */
        var audiopart = "< audio id = \"" + filename + "\">";
        audiopart = "< audio id = \"" + filename + "\">";
        audiopart += "< source src = \"" + "./assets/" + "drum/" + filename + "\""
        audiopart += " type = \"audio/wav\"/></audio >"
        audiostr += audiopart
    }
    document.getElementById("Audio").innerHTML = audiostr
}
通过循环遍历 drumconfig.Keyboard 中定义的键盘配置,为每个键盘按钮创建对应的 Audio 元素,该
```

```
元素包含对应音频文件的路径.
function InitializeKeyboard() {
    var KBPosition = GUIconfig.Keyboards.KBPosition
    var Scale = GUIconfig.Keyboards.Scale
    var kbstr = ""
    var kbabspos = document.getElementById("Keyboard").getBoundingClientRect()
    var kbapleft = kbabspos.left
    var kbaptop = kbabspos.top
    for (key in KBPosition) {
        var relp = KBPosition[key]
        var x = relp[0] * Scale + 20 + kbapleft;
        var y = relp[1] * Scale + 10 + kbaptop;
        var kbid = "Keyboard - " + key
        var engid = "Eng - " + key
        var chnid = "Chn-" + key
        var charidx = drumconfig.EnglishName.indexOf(key)
        var Chnchar = ""
        var keyclass = "SilentKeyboard"
        var Engclass = "EnglishCharacterNoChinese"
        if (charidx != -1) {
            Chnchar = drumconfig.ChineseName[charidx]
            Engclass = "EnglishCharacter"
        }
        /*键盘元素*/
        kbpart = "<div class = \"" + keyclass + "\" id = \"" + kbid</pre>
        kbpart += "\"style = \"position:absolute;left:" + x + "px;top:" + y + "px\" "
        kbpart += "onmousedown = \"ResponseMouseDown()\" onmouseup = \"ResponseMouseUp()\">"
        / * 英语字符元素 * /
        kbpart += "< span id = \"" + engid + "\" class = \"" + Engclass + "\">" + key +
"</span>"
        /*汉语字符*/
        kbpart += "< span id = \"" + chnid + "\" class = \"ChineseCharacter\">" + Chnchar +
"</span>"
        kbpart += "</div>\n"
        /*存放到 document 中*/
        kbstr += kbpart
    }
    document.getElementById("Keyboard").innerHTML = kbstr
}
```

初始化键盘根据图形用户界面(Graphical User Interface,GUI)的 config. Keyboards 中 定义的键盘位置、缩放比例等信息,来创建每个键盘按钮的 HTML 元素,并放置在界面上, 如图 3-8 所示。



图 3-8 锣鼓点键盘

```
< img src = "./assets/images/同光.png"alt = "同光" class = "front img"/>
< div class = "index_body_content">
       <div class = "index side box">
       < div class = "super_width_box">
 <!-- ul 界面 -->
             <!----> --> <!---->
                 </div>
     </div>
 </div>
< script >
$ (document).ready(function () {
 var ulWidth = $ (".ul_one").width();
  var animationDuration = 30000;
 function scrollText() {
    $ (".ul_one").animate({ "margin - left": - ulWidth }, animationDuration, "linear",
function() {
      $ (this).css("margin - left", 0);
     scrollText();
   });
  }
  scrollText();
  $ (".ul one").hover(
   function() {
      $ (this).stop();
   },
   function() {
      scrollText();
    }
 );
});
```

```
</script>
```

网页顶部京剧简介排布需要通过嵌入网页来实现,文字显示在图像之上向左滚动播放, 如图 3-9 所示。



图 3-9 顶部京剧简介

其他界面相关代码见"代码文件 3-2"。

3.4 功能测试

本部分包括运行项目、发送问题及响应。

3.4.1 运行项目

项目使用 VS Code 编写 HTML、CSS 和 JavaScript 的代码,实现相关功能并优化前端,项目文件夹 Demo 单独出现在 VS Code 右侧,运行界面如图 3-10 所示。



图 3-10 运行界面

在界面中,右击选择 Open with Live Server,如图 3-11 所示。

	intro	duce.html	opera_write.html	⇔index.html ×				
	inde	x.html > 🛛 I	html > 🛛 body > 📿 div.mai	in > ∅ div.left > ∅ div.top > ∅ a		转到定义	F12	
	14	<iink r<="" th=""><th>el="stylesheet" href=</th><th>"/assets/css/base.css"></th><th></th><th>結判引用</th><th>Shift+F12</th><th></th></iink>	el="stylesheet" href=	"/assets/css/base.css">		結判引用	Shift+F12	
		<link r<="" td=""/> <td>el="stylesheet" href=</td> <td><pre>"/assets/lib/bootstrap.css"></pre></td> <td></td> <td>1133300</td> <td></td> <td></td>	el="stylesheet" href=	<pre>"/assets/lib/bootstrap.css"></pre>		1133300		
	16	<link r<="" td=""/> <td>el="stylesheet" href=</td> <td>"/assets/css/index.css"></td> <td></td> <td>快速宣誓</td> <td>`</td> <td></td>	el="stylesheet" href=	"/assets/css/index.css">		快速宣誓	`	
_	17	<link r<="" td=""/><td>el="stylesheet" href=</td><td>#4607 to allow</td><td>Chife . Ale . 513</td><td></td>	el="stylesheet" href=	#4607 to allow	Chife . Ale . 513			
	18				直线所有与用	Shill+Ait+F12		
	19	<body st<="" td=""><td>yle="background-color</td><td></td><td>言心女性已</td><td>E2</td><td></td></body>	yle="background-color		言心女性已	E2		
	20 <div style="neight: 150px;"></div>							
	21		alass= main >			更以所有匹配项	Ctrl+F2	
	22	VUIV	div class= lert 2			格式化文档	Shift+Alt+F	
	23	1111	(a) href-""			重构	Ctrl+Shift+R	
	25		<pre> <img alt="</td" src="/assets/</pre></td><td>(images/京剧图片/珠帘塞.ipg"/><td></td><td></td><td></td><td></td></pre>					
	26		<pre><div class="txt"></div></pre>	Berly Martin I and Martin B		剪切	Ctrl+X	
	27		<h1>个人中心</h1>	h1>		1974	Ctrl+C	
	28		<				curve	
	29		· 点击	进入		粘點	Ctrl+V	
	30		<i class="</td><td>'iconfont icon-arrow-right"></i>	•	Onen with Live Comm	Abul AbuQ		
						Open with the Server	AIL+L AIL+U	
					┕┿	Stop Live Server	Alt+LAlt+C	
			漸变背景 mask</p					
	34		<pre><div class="mask"></div></pre>			命令面板	Ctrl+Shift+P	
	35							
	36	<pre></pre>	:/div>					
		<pre></pre>	div class="center">					
	38		<div class="b</td <td>00x"></td> <td></td> <td></td> <td></td> <td></td>	00x">				
	39		<img alt="'</td><td>*></td><td></td><td></td><td></td></tr><tr><td></td><td>40</td><td></td><td></div></td><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td>41</td><td></td><td><pre><div class=" box"="" src="/asset</td><td>:s/images/京剧图片/旦.jpg"/> <	(img_src="/assets/images/京剧图户	ī/巴班.jpg" alt	="">>		
	42		<div class="center</td><td>2"></div>					

图 3-11 选择界面

依次打开 VS Code→OPERA→index 文件,右击选择 Open With Live Serve,主页效果 如图 3-12 所示。



图 3-12 主页效果

单击创作进入主要功能界面,如图 3-13 所示。



图 3-13 主要功能界面

3.4.2 发送问题及响应

在"我想生成一段京剧"右侧对话框中输入"流水",表达大意为"我很开心",单击"生成" 按钮后,收到的答案显示在文本框内,如图 3-14 所示。



图 3-14 发送问题及响应