**成都锦城学院学生实验报告**

**课程名称： Web前后端综合开发**

**专 业： 前端技术开发方向 年 级： 2022级 班 级： 班**

**姓 名： 学 号：**

**实验组号： 九组 小组学生姓名： 邓军辉 梁航 王枭羽**

**实验日期： 2025 年 月 日 实验组织形式：□个人 ☑小组**

|  |
| --- |
| **一、实验名称：**  http模块应用  **二、实验目的：**   * 深入理解HTTP协议。 * 熟练使用Node.js http模块。 * 掌握Web API开发基础。 * 理解前后端分离模式。 * 提升数据处理和类型转换能力。   **三、实验要求：**   * API功能完整，前后端交互正确，数据格式规范，错误处理完善，代码结构清晰。 * 组织形式：以小组为单位开展，进行团队协作开发，完成后进行汇报、交流和讨论。实验结束后需提交个人实验报告，且相关代码和运行结果应以截图的方式在实验报告中呈现。 * 实验成绩评定要求： * 设计合理性（30分） * 方法实现的正确性和功能完整性（40分） * 代码的结构和可读性（20分） * 实验报告的规范性（10分）   **四、实验环境：**   * 环境：笔记本机房或教室，学生自带笔记本电脑，确保已安装Node.js运行环境 (建议使用最新的LTS版本)。 * 条件：代码编辑器，Visual Studio Code (推荐) 或其他常用代码编辑器。Web 浏览器，主流Web浏览器 (如Chrome, Firefox, Edge等)，用于测试前端页面和观察API响应。API测试工具，ApiPost，用于测试Web API。   **五、实验内容及方法步骤：**   * 实验内容   1、后端Web API开发 (Node.js)：   * + 使用http模块创建一个HTTP服务器，监听指定的端口。   + 编写请求处理函数，根据请求的URL路径和HTTP方法 (例如 GET 或 POST) 路由到不同的处理逻辑。   + 参数解析：     - * 对于GET请求，解析URL查询字符串中的参数。       * 对于POST请求，解析请求体中的参数 (例如，假设前端以application/x-www-form-urlencoded或application/json格式发送数据)。   + 算术运算逻辑：实现加、减、乘、除的运算函数，接收解析后的数值参数，并进行相应的计算。   + 结果封装：将运算结果封装成JSON格式的响应体，包括状态码 (例如成功或失败)、结果数据或错误信息。   + 设置响应头：设置正确的HTTP响应头，例如Content-Type: application/json。   + 错误处理：实现对非法参数 (非数值、缺少参数) 和运算错误 (除零) 的捕获和处理，并返回包含错误信息的JSON响应。   2、前端测试页面开发 (HTML/JavaScript)：   * 创建一个简单的HTML页面，包含两个数值输入框、一个运算符下拉选择框 (或一组单选按钮) 和一个提交按钮。 * 编写JavaScript代码，监听提交按钮的点击事件。 * 发送HTTP请求：使用fetch或XMLHttpRequest对象构建并发送HTTP请求到后端的API接口。可以选择GET或POST方法，并将用户输入的数值和运算符作为请求参数发送。 * 处理API响应：接收后端API返回的JSON响应。根据响应中的状态码或错误信息，在页面上显示运算结果或错误提示。 * 用户界面更新：将后端返回的运算结果或错误信息动态地显示在HTML页面上。 * 方法与步骤 * 创建后端项目：在VS Code中创建项目文件夹 (例如calculator-api)，并创建后端入口文件(api.js)。 * 编写后端API代码：在api.js中导入http和url(以及可能的 querystring或body-parser等模块)，实现 HTTP 服务器的创建、路由处理、参数解析、算术运算逻辑、结果封装和错误处理。 * 创建前端项目：在项目文件夹中创建前端相关的 HTML 文件(index.html)和JavaScript文件 (script.js) (可以放在一个 public或 frontend子目录下)。 * 编写前端测试页面代码：在index.html中设计用户界面元素 (输入框、选择框、按钮、结果显示区域)。在script.js中编 JavaScript代码实现事件监听、构建HTTP请求、发送请求以及处理和展示响应结果的逻辑。 * 运行后端API：在命令行终端中运行api.js(node api.js)。 * 在浏览器中打开前端页面：使用浏览器打开index.html文件。 * 进行测试：在前端页面输入数值和运算符，点击提交按钮，观察页面上显示的结果是否与预期一致，并测试各种错误情况。 * 调试与优化：根据测试结果调试前后端代码，优化代码结构和交互体验   **六、实验结果：**        **七、实验总结：**  本次实验，我没有使用额外的库，使用了node内置http模块实现了get和post操作。并且在开发过程中也发现了跨域问题，通过设置cors中间件，成功解决了跨域问题。本次实验收获满满。  **八、实验指导教师评定：**  在线提交实验报告，评定内容见在线课程对应任务点。 |