

# 基于多层架构的计算应用系统

## 一、系统介绍

该系统构建了一个多层架构的计算应用系统，包含基础数学运算、统计计算、矩阵计算和线性回归分析等功能，可完成多种计算的需求，项目主要包含三个核心部分：

### 1. C++计算核心模块：

- 实现了基础数学运算（加减乘除、幂运算）
- 统计计算功能（均值、中位数、标准差）
- 矩阵运算（乘法、转置、行列式计算）
- 线性回归分析
- 通过 pybind11 将 C++功能暴露给 Python

### 2. Python 后端 API（基于 FastAPI）：

- 封装 C++计算模块提供 RESTful API 接口
- 实现错误处理和配置管理
- 为前端提供计算服务

### 3. React 前端（TypeScript）：

- 现代化用户界面，分为四个主要功能区
- 基本计算器界面
- 统计计算器界面
- 矩阵计算器界面
- 回归分析可视化界面
- API 服务调用封装

利用 C++的高性能计算能力，Python 的灵活性和快速 API 开发，以及 React 的现代前端用户体验，构建一个完整的计算应用系统。

## 二、三层架构接口设计

### 1. 前端与后端之间的接口

前端通过 RESTful API 与后端进行通信，使用 HTTP 协议和 JSON 格式交换数据。

- 前端接口设计：

#### 1. API 服务层（frontend/src/services/api.ts）

前端定义一个 `calculatorApi` 服务对象，封装了与后端的所有通信逻辑；使用 `Axios` 库发送 HTTP 请求；定义请求和响应的 TypeScript 接口，确保类型安全。

提供了四个主要方法：`basicOperation`: 基本数学运算；`statisticalCalculation`: 统计计算；`matrixOperation`: 矩阵运算；`linearRegression`: 线性回归分析。

## 2. 类型定义

为每种操作定义请求和响应的接口类型，例如：`BasicOperationRequest`、`BasicOperationResponse` 等，这些类型定义确保前后端数据交换的一致性。

### 2. 后端与前端之间的接口

后端使用 FastAPI 框架提供 RESTful API 接口，接收前端请求并返回 JSON 格式的响应。

- 后端 API 接口设计：

1. 路由定义 (`backend/app/api/routes.py`)

定义四个主要 API 端点：`/api/basic`: 基本数学运算、`/api/stats`: 统计计算、`/api/matrix`: 矩阵运算、`/api/regression`: 线性回归分析，每个端点都使用 POST 方法，接收 JSON 请求体。

2. 数据验证

使用 Pydantic 模型定义请求和响应的数据结构，实现自动数据验证和类型转换，通过验证器 (validator) 添加了自定义验证规则，例如：检查操作类型是否有效、数组不能为空等。

3. 响应模型

每个 API 端点都定义了明确的响应模型，确保返回给前端的数据格式一致。

### 3. 后端与 C++ 计算模块之间的接口

后端通过 Python 绑定调用 C++ 计算模块，实现高性能计算。

- 后端服务层设计：计算服务类 (`backend/app/services/calculator.py`)

定义 `CalculatorService` 类，封装对 C++ 模块的调用，提供了与 API 路由对应的四个主要方法，实现了错误处理和异常转换。

### 4. C++ 计算模块与 Python 后端之间的接口

C++ 计算模块通过 pybind11 将函数暴露给 Python，实现语言间的无缝调用。

- C++ 模块接口设计：

1. C++ 函数实现 (`src/calculator.cpp`)

在 `calculator` 命名空间中实现了各种计算函数，包括基本运算、统计计算、矩阵运算和线性回归，使用 C++ 标准库实现高性能计算。

2. 头文件声明 (`src/calculator.h`)

声明所有计算函数的接口，定义数据类型。

3. Python 绑定 (`src/binding.cpp`)

使用 pybind11 创建 Python 模块 `cpp_calculator`，将 C++ 函数绑定到 Python 函数，添加了函数文档和参数名称。

### 5. 三层架构中的接口设计图解

为了更好地展示前端、后端和计算模块之间的接口关系，下面的详细的图表可更清晰地理解三层架构中的接口设计和数据流程。

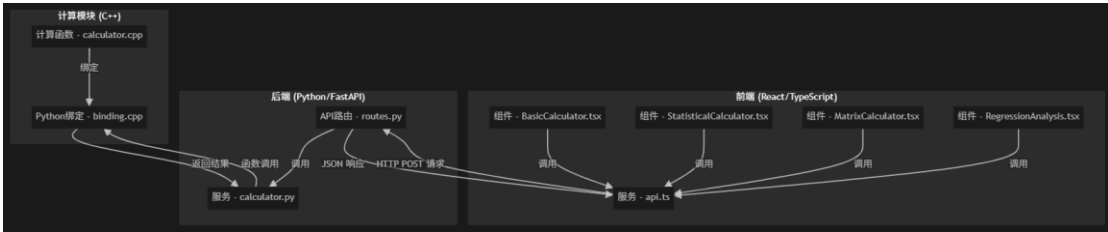


图 1 创建三层架构接口关系图

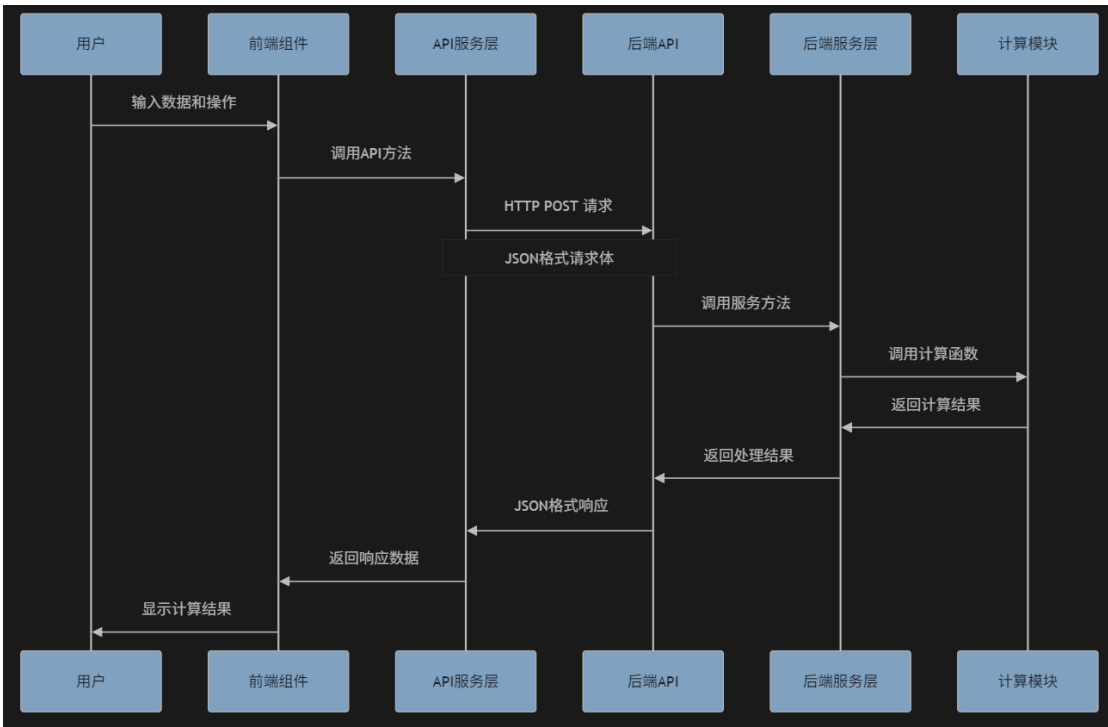


图 2 前端与后端之间的接口

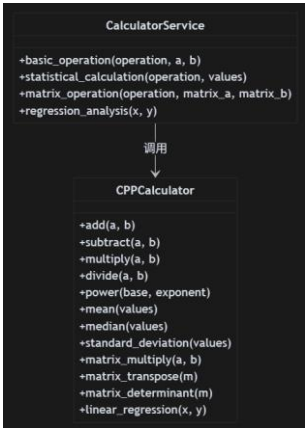


图 3 后端与 C++ 计算模块之间的接口

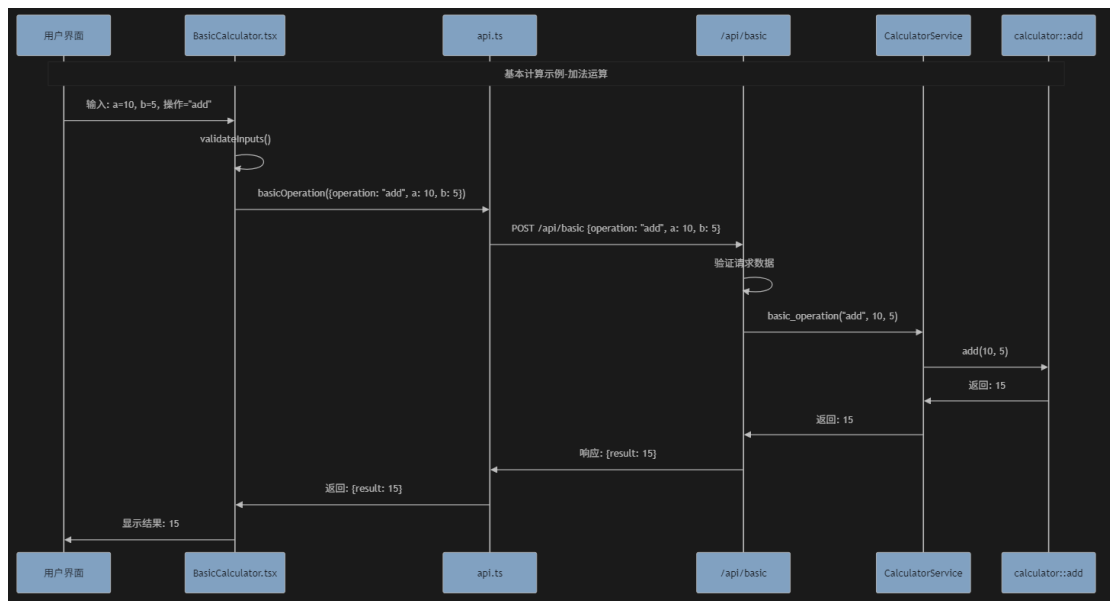


图 4 完整的数据流程示例图