**软件开发设计文档**

**文档管理信息表**

|  |  |
| --- | --- |
| **主题** |  |
| **版本** |  |
| **内容** |  |
| **关键字** |  |
| **参考文档** |  |
| **创建时间** |  |
| **创建人** |  |
| **最新发布日期** |  |

**文档变更纪录**

|  |  |  |
| --- | --- | --- |
| **更改人** | **日期** | **更改内容** |
| 龙雲 | 2020-6-30 | 创建文件 |
|  |  |  |
|  |  |  |
|  |  |  |

**目 录**

1 开发规划 1

1.1 开发人员 1

1.2 开发计划 2

1.3 开发环境和工具 3

1.4 开发规范 3

2 总体设计 4

2.1 概念术语描述 4

**2.1.1** 术语1 4

**2.1.2** 术语2 4

2.2 基本设计描述 5

**2.2.1** 系统总体逻辑结构图 5

**2.2.2** 系统部署结构图 6

2.3 模块列表 8

3 数据结构 9

4 接口规范 10

4.1 <模块1 API> 10

**4.1.1** Interface1 10

**4.1.2** Interface2 10

4.2 <模块2 API> 11

4.3 <模块3 API> 11

4.4 <数据库API> 11

5 模块设计 12

5.1 Module1设计 12

**5.1.1** 模块设计描述 12

**5.1.2** 模块界面描述 12

5.2 Module2设计 13

6 附录 14

6.1 第三方组件 14

6.2 参考资料 15

6.3 附加文档 15

# 开发规划

## 开发人员

|  |  |  |
| --- | --- | --- |
| **人员** | **主要职责** | **备注** |
| 龙雲 | * 项目设计 * 前端编写与维护 * 后端编写与维护 * 项目进度控制 |  |

## 开发计划

|  |  |  |  |
| --- | --- | --- | --- |
| **阶段** | **主要工作** | **应完成的文档部分** | **起止时间** |
| 需求分析 | 确定系统运行环境 |  | 2020.6.24 |
| 建立系统逻辑模型 | 系统总体逻辑架构图 |
| 确定系统功能及性能要求 | 基本设计描述 |
| 确认项目开发计划 | 开发计划 |
| 概要设计 | 建立系统总体结构，划分功能模块 | 模块列表、模块设计 | 2020.6.25 |
| 数据库设计 | 数据库API |
| 实现 | 编写程序源代码 |  | 2020.6.26  -  2020.6.28 |
| 进行测试和调试 |  |
| 编写操作文档 | 操作文档 |
| 测试 | 测试整个软件系统 |  | 2020.6.29 |
| 上线 | 部署上线 |  | 2020.6.30 |

## 开发基础设施和代码生产工具

**开发工具（包括程序语言与技术框架）**

|  |  |
| --- | --- |
| **工具** | **作用** |
| React | 开发网站前端 |
| Element-React | 前端UI框架 |
| Express | 构建后端服务 |
| MongoDB | 后端数据库 |

## 开发与运行环境（IDE与服务器）

|  |  |
| --- | --- |
| **环境名称** | **对应的开发流程** |
| VS code | 网站前端和后端开发与封装 |
| Node js | 网站运行环境 |
| Docker | 网站的部署与挂载运行 |

## 开发规范

|  |  |
| --- | --- |
| **规范名称** | **对应的开发流程** |
| React编码规范 | 网站前端开发 |
| Express规范 | 网站后端API编写 |
| 系统目录规范 | 后端静态文件的管理规范 |

# 总体设计

## 概念术语描述

**前后端分离**

前后端分离已成为互联网项目开发的业界标准使用方式，通过客户端渲染+API的方式效的对系统各层进行解耦，并且前后端分离会为以后的大型分布式架构、弹性计算架构、微服务架构、多端化服务（多种客户端，例如：浏览器，车载终端，安卓，IOS等等）打下坚实的基础。

同时，这种开发方式也很好的体现了设计模式中的关注点分离原则，明确了程序员的分工，便于后续的迭代和维护，让项目更加的工程化、开发更加的流程化。

**跨域问题**

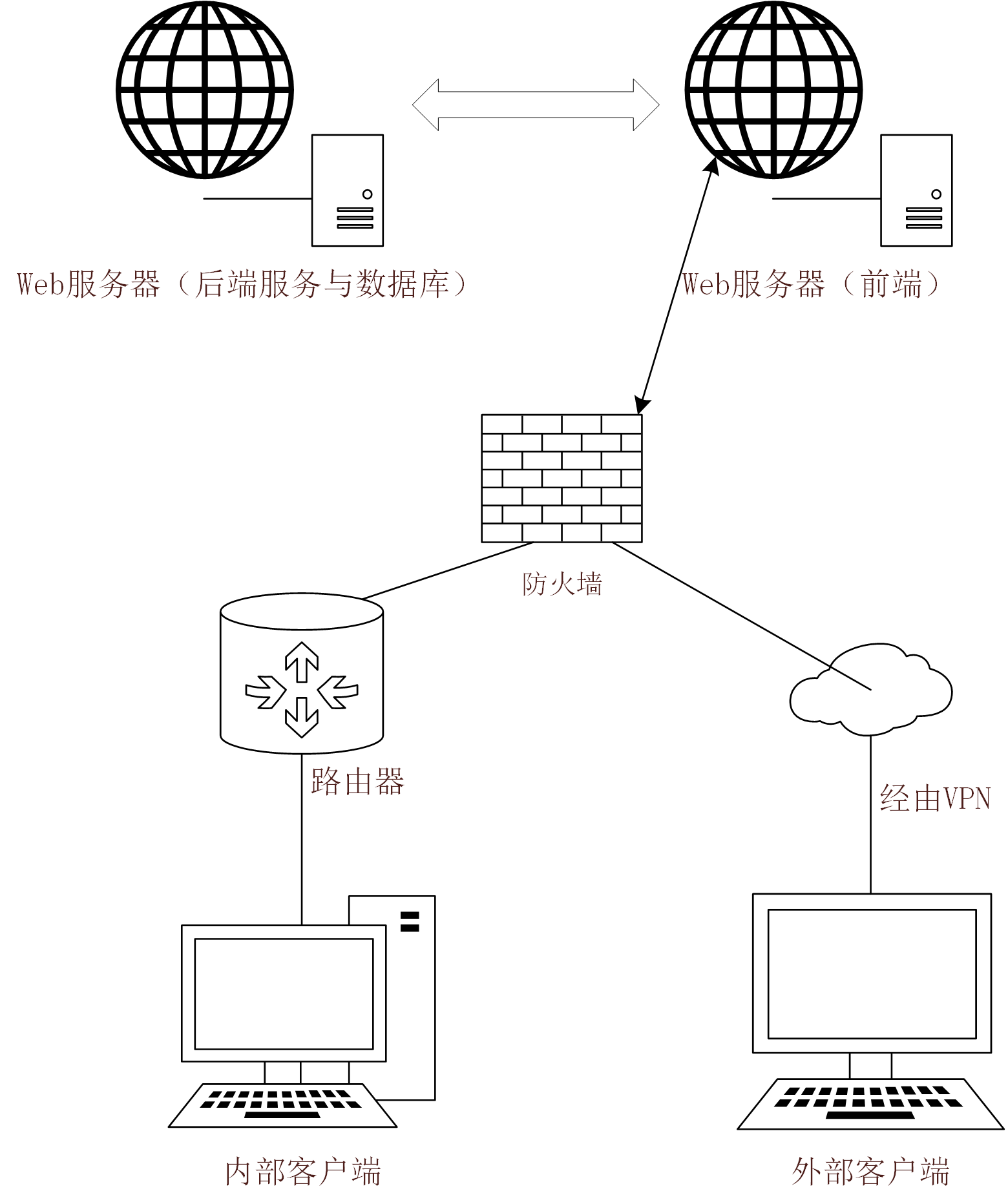
当一个请求url的协议、域名、端口三者之间任意一个与当前页面url不同即为跨域，正常情况下这种请求会被禁止，这是由于浏览器的同源策略限制。同源策略（Same origin policy）是一种约定，它是浏览器最核心也最基本的安全功能，如果缺少了同源策略，则浏览器的正常功能可能都会受到影响。可以说Web是构建在同源策略基础之上的，浏览器只是针对同源策略的一种实现。同源策略会阻止一个域的javascript脚本和另外一个域的内容进行交互。所谓同源（即指在同一个域）就是两个页面具有相同的协议（protocol），主机（host）和端口号（port）。

在前后端分离架构的项目中，由于前端页面和后台服务分离开发，各自成项，因此二者可以各自打包部署在不同的服务器上，这种做法将项目高度解耦，不仅便于维护和迭代，同时也缓解了服务器端压力，提升了容灾性。但是由于不同服务器在url、端口号上都存在差异，因此就会出现跨域问题。本项目解决了不同域名和端口之间的交互问题，从而实现了将前后端分离部署，与业界主流的项目开发与部署模式保持一致。

## 基本设计描述

本项目包括登录、注册及任务清单管理功能，其中，任务清单管理包括新建任务、删除任务、修改任务状态、批量删除已完成任务等功能。本项目基于jwt实现了登录注册功能，动态的token来进行身份认证，具有很好地安全性。如果用户不注销，刷新后也可以直接进入任务清单界面，即可以保持登录状态，满足了不同用户对各自Todo数据的操作需求。

### 系统部署结构图



# 数据结构

本项目采用MongoDB数据库，这是一个基于分布式文件存储的数据库。数据结构如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | collection | Document | 注释/说明 |
| 1 | users | {  "\_id": "5ef9f3d8d283ab1dbccb92a9",  "createAt": "2020-06-29T13:40:42.942Z",  "username": "Longyun",  "password": "52c69e3a57331081823331c4e69d3f2e",  "\_\_v": 0  } | 存储用户信息，其中，用户密码是MD5加密后的。 |
| 2 | todos | {  "\_id": "5efab4a4066e6448ecddee9d",  "createAt": "2020-06-30T03:41:02.296Z",  "username": " Longyun ",  "todoname": "任务1",  "status": true,  "\_\_v": 0  } | 存储用户的任务名及其状态。 |

# 后端接口规范

在前后端分离的架构设计越来越流行的趋势下，API开发已经成为了后端的必备技能。在本项目中，我们采用nodejs 下的Express构建API，为前端提供服务。

## <TodoMVCService API>

在本项目中，前端通过HTTP POST 操作发出请求，服务器端接收请求后对数据库操作并做出响应，由此实现项目的功能。

### 注册

* **请求地址**

|  |  |
| --- | --- |
| **操作** | **注册** |
| **协议** | **HTTP** |
| **方式** | **Post** |
| **地址** | [http://127.0.0.1:3001/users/register](http://127.0.0.1:3001/users/register%20) |

* **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **参数名** | **中文名称** | **必选** | **备注** |
| **1** | **username** | **用户名** | **Y** | **用户名是唯一的，不允许重名** |
| **2** | **password** | **密码** | **Y** | **MD5加密** |

* **返回数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **返回值** | **中文名称** | **备注** |
| **1** | **Status** | **状态** | **请求返回HTTP 200 HTTP 400 等返回值，** |
| **2** | **Message** | **返回信息** | **返回成功信息或错误信息** |
| **3** | **token** | **口令** | **包括用户名等信息** |

### 登录

* **请求地址**

|  |  |
| --- | --- |
| **操作** | **用户登录** |
| **协议** | **HTTP** |
| **方式** | **Post** |
| **地址** | <http://127.0.0.1:3001/users/login> |

* **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **参数名** | **中文名称** | **必选** | **备注** |
| **1** | **username** | **用户名** | **Y** | **用户名是唯一的，不允许重名** |
| **2** | **password** | **密码** | **Y** | **MD5加密** |

* **返回数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **返回值** | **中文名称** | **备注** |
| **1** | **Status** | **状态** | **请求返回HTTP 200 HTTP 400 等返回值，** |
| **2** | **Message** | **返回信息** | **返回成功信息或错误信息** |
| **3** | **token** | **口令** | **包括用户名等信息** |

### 新建任务

* **请求地址**

|  |  |
| --- | --- |
| **操作** | **用户登录** |
| **协议** | **HTTP** |
| **方式** | **Post** |
| **请求头** | **'Content-Type': 'application/x-www-form-urlencoded',**  **'Authorization': "Bearer " + token** |
| **地址** | [http://127.0.0.1:3001/todos/create](http://127.0.0.1:3001/todos/create%20) |

* **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **参数名** | **中文名称** | **必选** | **备注** |
| **1** | **todoname** | **任务名** | **Y** |  |
| **2** | **status** | **状态** | **N** | **默认TRUE（未完成）** |

* **返回数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **返回值** | **中文名称** | **备注** |
| **1** | **Status** | **状态** | **请求返回HTTP 200 HTTP 400 等返回值，** |
| **2** | **Message** | **返回信息** | **返回成功信息或错误信息** |

### 删除任务

* **请求地址**

|  |  |
| --- | --- |
| **操作** | **删除任务** |
| **协议** | **HTTP** |
| **方式** | **Post** |
| **请求头** | **'Content-Type': 'application/x-www-form-urlencoded',**  **'Authorization': "Bearer " + token** |
| **地址** | <http://127.0.0.1:3001/todos/del> |

* **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **参数名** | **中文名称** | **必选** | **备注** |
| **1** | **\_id** | **任务id** | **Y** |  |

* **返回数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **返回值** | **中文名称** | **备注** |
| **1** | **Status** | **状态** | **请求返回HTTP 200 HTTP 400 等返回值，** |
| **2** | **Message** | **返回信息** | **返回成功信息或错误信息** |

### 删除全部已完成任务

* **请求地址**

|  |  |
| --- | --- |
| **操作** | **删除全部已完成任务** |
| **协议** | **HTTP** |
| **方式** | **Post** |
| **请求头** | **'Content-Type': 'application/x-www-form-urlencoded',**  **'Authorization': "Bearer " + token** |
| **地址** | [http://127.0.0.1:3001/todos/delteCompleted](http://127.0.0.1:3001/todos/delteCompleted%20) |

* **返回数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **返回值** | **中文名称** | **备注** |
| **1** | **Status** | **状态** | **请求返回HTTP 200 HTTP 400 等返回值，** |
| **2** | **Message** | **返回信息** | **返回成功信息或错误信息** |

### 修改任务状态

* **请求地址**

|  |  |
| --- | --- |
| **操作** | **修改任务状态** |
| **协议** | **HTTP** |
| **方式** | **Post** |
| **请求头** | **'Content-Type': 'application/x-www-form-urlencoded',**  **'Authorization': "Bearer " + token** |
| **地址** | [http://127.0.0.1:3001/todos/changeStatus](http://127.0.0.1:3001/todos/del) |

* **请求参数**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **参数名** | **中文名称** | **必选** | **备注** |
| **1** | **\_id** | **任务id** | **Y** |  |

* **返回数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **返回值** | **中文名称** | **备注** |
| **1** | **Status** | **状态** | **请求返回HTTP 200 HTTP 400 等返回值，** |
| **2** | **Message** | **返回信息** | **返回成功信息或错误信息** |

### 获取任务列表

* **请求地址**

|  |  |
| --- | --- |
| **操作** | **获取任务列表** |
| **协议** | **HTTP** |
| **方式** | **Post** |
| **请求头** | **'Content-Type': 'application/x-www-form-urlencoded',**  **'Authorization': "Bearer " + token** |
| **地址** | <http://127.0.0.1:3001/todos/todoList> |

* **返回数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **返回值** | **中文名称** | **备注** |
| **1** | **Status** | **状态** | **请求返回HTTP 200 HTTP 400 等返回值，** |
| **2** | **Message** | **返回信息** | **返回成功信息或错误信息** |

# 前端界面设计

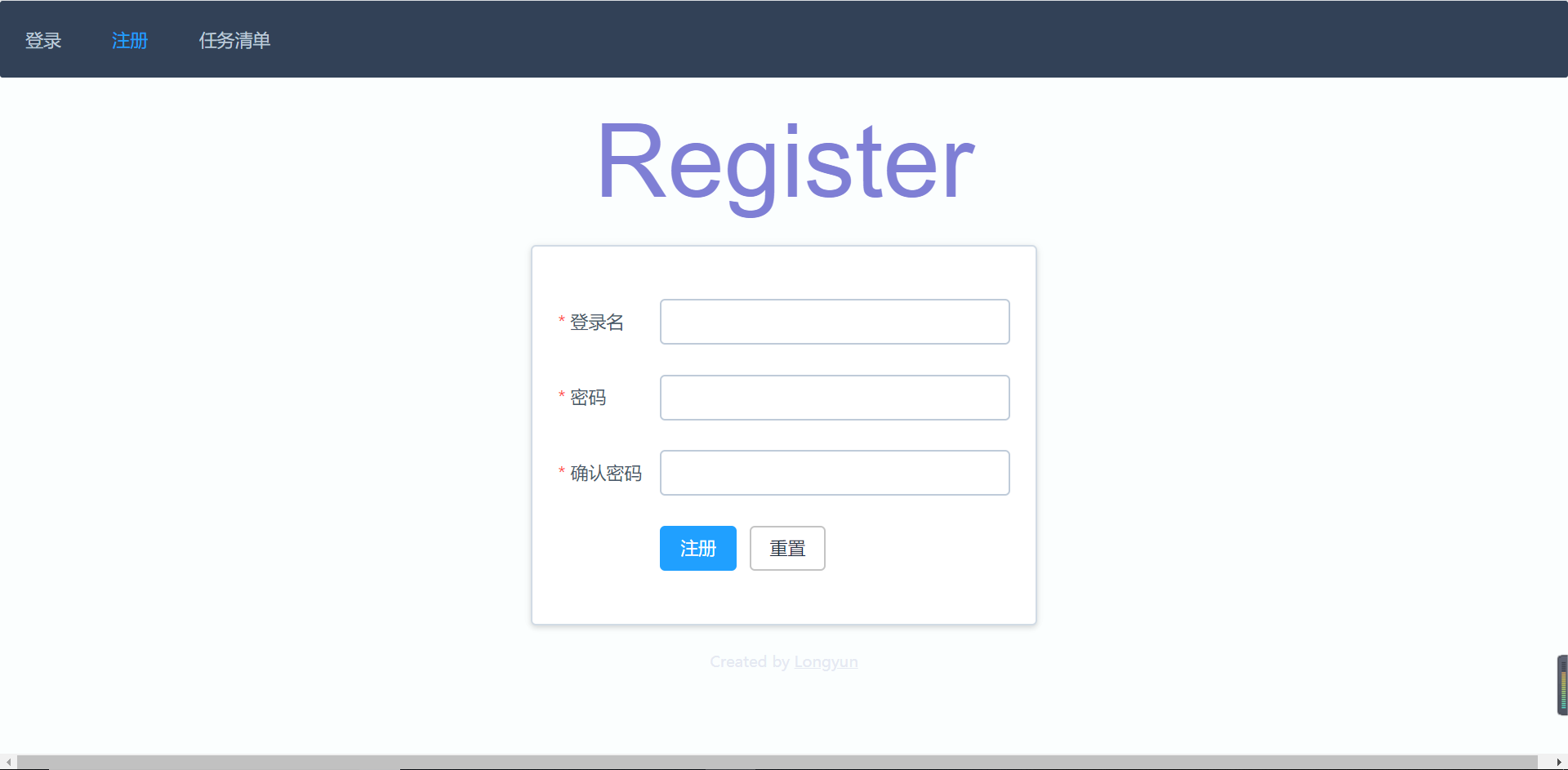
本项目共有四个界面，即注册界面、登录界面、任务清单界面和使用情况统计界面。

## 注册界面设计

**注 册 界 面 设 计 说 明 表**

编号：001

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 注册 | 模块代号 | 001 |
| 适用范围 | 局部 | 类型 | 窗口 |
| 模块功能 | 为首次登录的用户实现注册账号的功能。 | | |
| 限制条件 | 用户必须能够访问互联网。 | | |
| 输 入 | 用户名、密码、确认密码 | | |
| 输 出 | 注册成功发放token | | |
| 相关对象及接口 | ToDoMVCService API  地址：<http://127.0.0.1:3001/users/register/>  涉及操作：POST  协议：HTTP | | |
| 备注 |  | | |



## 登录界面设计

**登 录 界 面 设 计 说 明 表**

编号：002

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 登录 | 模块代号 | 002 |
| 适用范围 | 局部 | 类型 | 窗口 |
| 模块功能 | 用户登录 | | |
| 限制条件 | 用户必须能够访问互联网。 | | |
| 输 入 | 用户名、密码 | | |
| 输 出 | 登录成功发放token | | |
| 相关对象及接口 | ToDoMVCService API  地址：<http://127.0.0.1:3001/users/login/>  涉及操作：POST  协议：HTTP | | |



## 任务清单界面设计

**任 务 清 单 设 计 说 明 表**

编号：003

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 任务清单 | 模块代号 | 003 |
| 适用范围 | 局部 | 类型 | 窗口 |
| 模块功能 | 查询指定用户Todo数据  新建Todo数据  改变TodoTodo状态  删除Todo数据  删除全部已完成Todo数据  全部完成/全部未完成  不同状态的任务数量统计 | | |
| 相关对象及接口 | ToDoMVCService API  涉及地址：  [http://127.0.0.1:3001/todos/todoList/](http://127.0.0.1:3001/users/login/)  <http://127.0.0.1:3001/todos/create/>  [http://127.0.0.1:3001/todos/changeStatus/](http://127.0.0.1:3001/users/login/)  <http://127.0.0.1:3001/todos/del/>  [http://127.0.0.1:3001/todos/delteCompleted/](http://127.0.0.1:3001/users/login/)  涉及操作：POST  协议：HTTP | | |



## 使用情况统计界面设计

**使 用 情 况 统 计 界 面 设 计 说 明 表**

编号：004

|  |  |  |  |
| --- | --- | --- | --- |
| 模块名称 | 使用情况统计 | 模块代号 | 004 |
| 适用范围 | 局部 | 类型 | 窗口 |
| 模块功能 | 查询不同日期的系统使用情况 | | |
| 相关对象及接口 | ToDoMVCService API  涉及地址：  http://127.0.0.1:3001/  涉及操作：POST  协议：HTTP | | |

