前段渲染框架

概要设计文档

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： |  |
| 当前版本： | 1.0 |
| 作 者： | 杨武 |
| 完成日期： | 2022/11 |

目录

[1. 引言 3](#_Toc152024054)

[1.1 目的 3](#_Toc152024055)

[1.2 技术 3](#_Toc152024056)

[1.3 参考文献 3](#_Toc152024057)

[1.5 概述 3](#_Toc152024058)

[2. 系统架构 3](#_Toc152024059)

[2.1 总体架构 3](#_Toc152024060)

[2.2 模块划分 4](#_Toc152024061)

[1.解析器 4](#_Toc152024062)

[2,渲染器 4](#_Toc152024063)

[3,数据代理 4](#_Toc152024064)

[4.路由 4](#_Toc152024065)

[5.音乐平台组件 4](#_Toc152024066)

[3 数据流程 4](#_Toc152024067)

[3.1.渲染流程 4](#_Toc152024068)

[3.2.更新流程 5](#_Toc152024069)

[4 接口设计 5](#_Toc152024070)

# 1. 引言

## 1.1 目的

本项目旨在搭建一个前段渲染框架并在此基础上搭建一个在线音乐平台测试前段框架的性能。采用node.js模仿vue3的渲染逻辑，并对其部分逻辑进行一定的优化。在此基础上，搭建一个在线音乐平台，测试框架的性能。

## 1.2 技术

1.node.js

2.Spingboot

## 1.3 参考文献

1.《Vue.js设计与实现》

2.《基于Vue+SpringCloud博客的设计与实现》

## 1.5 概述

# 2. 系统架构

## 2.1 总体架构

渲染框架采用MVVM的渲染架构，将视图（view）和模型（model）完全分离，通过一个视图模型（viewModel）来实现两者之间的交互。

在MVVM架构中，视图只负责显示用户界面，模型只负责存储数据，视图模型则负责处理业务逻辑和数据操作，并通过数据绑定和命令模式来与视图进行通信。

后端采用MVC的架构, [MVC模式中，Model（模型）表示应用程序核心（比如数据库记录列表）；View（视图）显示数据（数据库记录）；Controller（控制器）处理输入（写入数据库记录）。MVC模式定义Web应用程序带有三个逻辑层：业务层（模型逻辑）、显示层（视图逻辑）和输入控制（控制器逻辑）1。Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据1。View（视图）是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的1。Controller（控制器）是应用程序中处理用户交互的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据](https://www.runoob.com/aspnet/mvc-intro.html)。

## 2.2 模块划分

### 1.解析器

1解析器负责读取组件，并将组件编译成相应的虚拟DOM

### 2,渲染器

1. 渲染器负责将虚拟DOM和数据绑定
2. 渲染器负责生成对应的DOM，并绑定到父节点上
3. 渲染器负责更新组件
4. 渲染器负责卸载组件

### 3,数据代理

1. 数据代理负责在绑定数据触发时出发渲染器
2. 数据代理负责监听数据，监听数据变化时触发绑定函数

### 4.路由

1. 路由负责提供当前页面组件

2. 路由负责传值

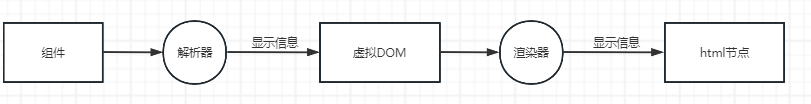
3. 路由负责控制是否有进入的权限

### 5.音乐平台组件

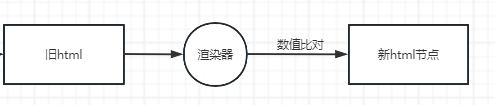
1.音乐平台细分为登录，注册，歌曲平台

# 3 数据流程

## 3.1.渲染流程



## 3.2.更新流程



# 4 接口设计

|  |  |  |  |
| --- | --- | --- | --- |
| 接口名称 | 接口参数 | 执行时间 | 备注 |
| BeforeEnterRouter | 无 | 进入路由之前执行 |  |
| AfterEnterRouter | 无 | 进入路由之后执行 |  |
| BeforeCreate | 组件虚拟DOM | 组件数据绑定之前执行 |  |
| Mouted | 组件挂载父节点 | 组件挂载之后执行 |  |
| UnMouted | 组件挂载父节点 | 组件卸载之后执行 |  |