

临沂市“城市大脑”物联网平台项目  
信息资源服务系统  
详细设计说明书

编制单位：临沂大学

编制时间：2023年9月

## 目录

|                        |   |
|------------------------|---|
| 第 1 章 前言 .....         | 1 |
| 1.1. 编写目的 .....        | 1 |
| 1.2. 参考资料 .....        | 1 |
| 第 2 章 项目综述 .....       | 1 |
| 2.1. 建设内容 .....        | 1 |
| 2.2. 设计原则 .....        | 3 |
| 2.3. 技术实现及技术路线要求 ..... | 4 |
| 2.4. 遵循依据及参考规范 .....   | 6 |
| 第 3 章 系统实现设计 .....     | 7 |
| 3.1. 系统架构图 .....       | 7 |
| 第 4 章 软件系统设计 .....     | 8 |
| 4.1. 信息资源服务 .....      | 8 |

# 第 1 章 前言

## 1.1. 编写目的

本文档是根据项目合同的要求及需求调研的结果，整理出一份关于临沂大学智慧物联信息资源服务平台的软件设计文档。该说明书经客户和开发项目经理确认后，作为软件开发的依据。本文档的预期读者包括项目技术人员、项目架构师、程序员和测试工程师等。

## 1.2. 参考资料

- 《智慧物联信息资源服务平台需求说明书》
- 《智慧物联信息资源服务平台功能清单》

# 第 2 章 项目综述

## 2.1. 建设内容

- （一）统一认证登录。本系统采用统一认证登录方式，极大的提高了安全性。
- （二）服务管理功能。其中又包含数据目录、领域管理、接口发布三个子模块。服务管理模块是一个核心模块，它负责管理项目中的所有构建内容、数据、领域和接口发布。该模块将支持项目团队协作，确保数据和接口发布的一致性和可追踪性，数据在模块内部和与其他模块之间的流动方式是通过 API 接口和数据库连接实现的。
- （三）资源应用审批功能。其中又包含应用列表审批、资源申请审批、

用户评分三个子模块。资源应用审批模块是项目中的关键环节，旨在管理和优化资源分配、应用列表的审批流程、资源的请求和用户评分的评估。在这个模块内，批准和监督资源的使用、应用的列入和用户的评价都得到了仔细的处理。在项目中起着关键作用，确保资源的合理使用、应用的质量和用户满意度。通过资源应用审批模块，系统能够实现资源的高效管理、应用的精选和用户的积极参与，从而支持项目的成功执行。

- （四）互动交流功能。其中又包含数据需求、问题反馈、纠错信息三个子模块。互动交流功能模块提供了一个平台，用户可以提交数据需求、问题反馈以及纠错信息，以实现更加紧密的交流和反馈。促进了用户与系统的积极互动，提供了一种渠道，用户可以表达需求、报告问题、纠正数据，从而有助于改进和优化系统，以满足项目需求。这个模块强调用户参与和反馈，以提高系统的质量和用户满意度。
- （五）平台管理功能。平台管理模块是项目中的关键模块，包括门户咨询维护、说明文档、常见问题、统一认证、隐私政策、API 接口预警以及接口预警记录六个子模块，旨在提供一站式的平台管理和维护功能，以确保系统的可用性和合规性。提供了一系列工具和功能，以确保系统的可用性、用户支持和合规性。这有助于满足项目需求，提供用户友好的平台和维护功能，以确保平台的可靠性和性能。
- （六）统一消息功能。统一消息模块是一个关键的通信工具，允许用户访问和查看消息列表。这个模块用于管理和呈现系统生成的消息，以使用户能够及时获取信息、通知和通信。消息列表通常包括各种类型的消息，如通知、提醒、事件更新等，以确保用户了解系统中的重要信息。

有助于用户在系统内保持信息流畅，获得关键信息，了解系统事件和活  
动，以提高用户的体验和互动。这对于项目管理和用户参与至关重要。

➤ （七）系统管理。系统管理模块主要负责管理系统的用户角色、用户管  
理、菜单配置和部门管理。它提供了对系统访问权限、用户设置以及界  
面配置的全面控制和管理。这个模块有助于维护系统的安全性、可维护  
性和用户管理。对于项目的管理和维护至关重要。

➤ （八）banner 管理。Banner 图的管理 展示 添加

## 2.2. 设计原则

➤ 清晰性：确保系统的设计和代码是清晰易懂的。使用有意义的命名、注  
释和文档，以便其他开发人员能够理解和维护系统；

➤ 模块化：将系统拆分模块，每个模块处理特定功能或任务。这提高了代  
码的可维护性，同时也方便了单元测试和集成；

➤ 耦合性：减少模块之间的依赖性，以降低系统的复杂性。使用接口、抽  
象和依赖注入等方法来实现松耦合；

➤ 高内聚：确保每个模块或类的职责是一致的，避免一个类或方法负责过  
多的事情。这有助于代码的可维护性和可读性；

➤ 可拓展性：考虑未来的需求，使系统能够容易地扩展新功能或修改现有  
功能。使用设计模式和适当的架构来实现可扩展性；

➤ 安全性：确保系统设计和实现满足安全性需求，包括数据加密、身份验  
证和访问控制；

➤ 可测试性：编写易于测试的代码，以支持单元测试、集成测试和验收测  
试。这有助于检测和修复问题，确保系统的质量；

➤ 异常处理：设计健壮异常处理机制，以处理意外情况和错误。提供明确的错误消息和日志记录；

➤ 可重用性：鼓励重用现有的代码和组件，减少重复工作，提高效率；

➤ 用户体验：关注用户界面的可用性和用户体验。确保界面直观、友好和易于导航；

➤ 文档化：为设计、代码和系统提供适当的文档。这有助于团队成员理解系统，同时也为未来的维护和升级提供指导；

➤ 性能优化：关注系统的性能，使用合适的数据结构和算法，避免不必要的性能瓶颈。

## 2.3. 技术实现及技术路线要求

信息资源服务平台在充分考虑了安全性、稳定性、高性能、独立性、实用性、兼容性、可维护性情况之后，决定采用 springboot 微服务框架服务完成所有功能建设。

➤ 架构：

初步选定采用基于 springboot 微服务服务架构，实现：

- 服务间通讯采用 http 通信及消息队列，数据格式采用 json 格式
- 通过 Docker 实现服务的自动化部署
- 能够快速创建基于 spring 的程序
- 能够直接使用 Java main 方法启动内嵌的 Tomcat 服务器运行 springboot 程序，不需要部署 war 包
- 提供约定的 starter POM 来简化 Maven 配置，让 Maven 的配置变得简单
- 自动化配置，根据项目的 Maven 依赖配置，springboot 自动配置 spring、

springmvc 等

- 提供了程序的健康检查功能
- 基本可以完全不使用 xml 配合文件，采用注解配置

#### ➤ 数据库

初步选定采用 MySQL。

- 快速弹性，应对不确定的业务增长；
- 性能卓越、服务稳定，很少出现异常宕机。
- 开放源代码且无版权制约，自主性及使用成本低。
- 体积小，安装方便，易于维护。
- 支持多种操作软件，提供多种 API 接口，支持多种开发语言。

#### ➤ 接口

初步选定采用 RESTful API 的服务接口，方便不同的前端设备与后端进行通信，可以通过一套统一的接口为 Web，iOS 和 Android 提供服务。

#### ➤ 界面

初步选定管理端采用 Element UI+vue 的开发方式

- 组件化模块，提高代码复用率，且让代码更好维护
- 声明式编码，让编码人员不用直接操作 DOM，提高开发效率
- 使用 DOM+优秀的 Diff 算法，尽量复用 DOM 节点

#### ➤ 其他

- 云存储：采用阿里云 OSS 的对象存储服务

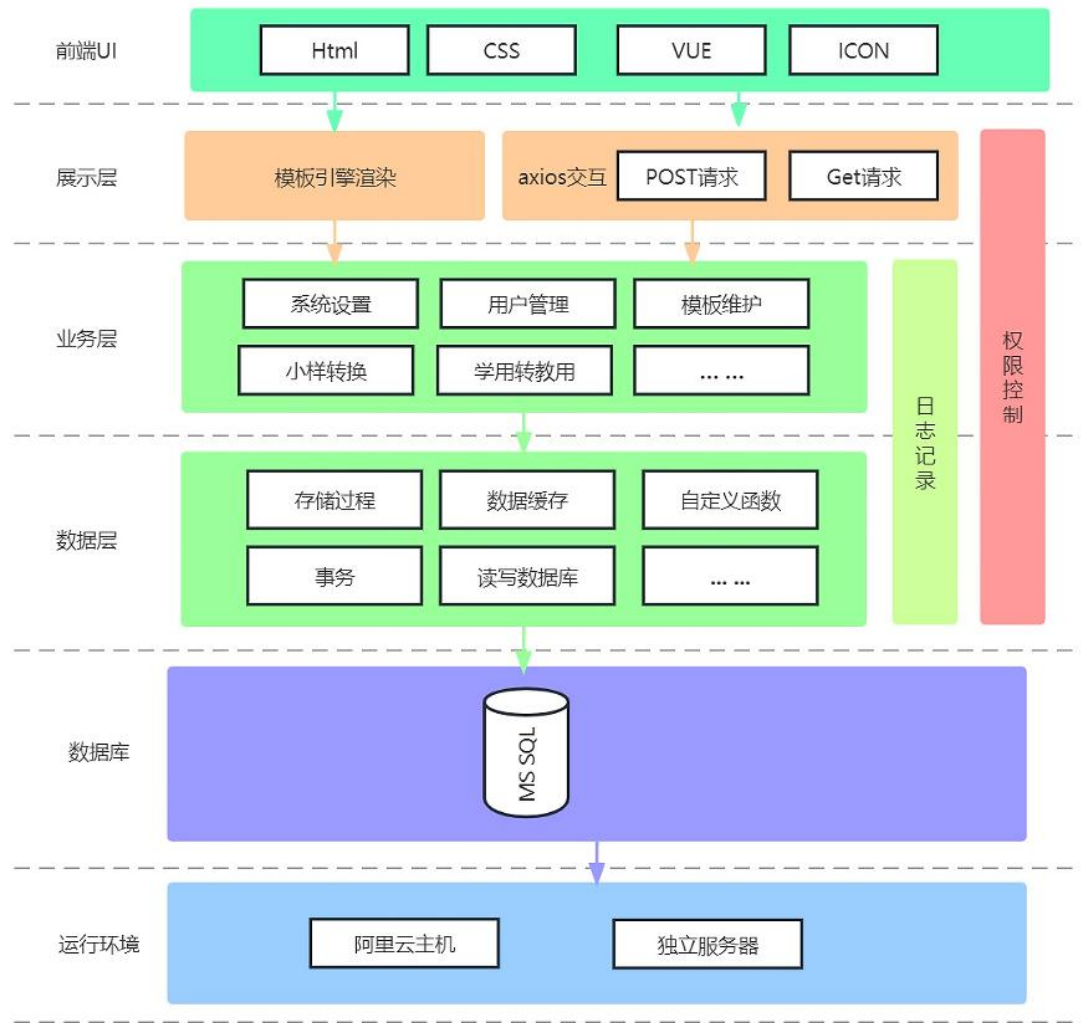
## 2.4. 遵循依据及参考规范

- 《物联网基础安全标准体系建设指南》
- 《城市物联网平台技术标准指南》
- 《城市物联网平台安全技术规范》
- 《GB/T 21063-2007 政务信息资源目录体系》
- 《GB/T 21062-2007 政务信息资源交换体系》
- 《阿里巴巴 Java 开发手册》



# 第3章 系统实现设计

## 3.1. 系统架构图



## 第4章 软件系统设计

### 4.1. 信息资源服务

#### 4.1.1. 登录

##### 1.1.1.1 界面设计



#### 4.1.2. 系统管理

##### 1.1.1.2 功能介绍

- 1) **菜单管理**：实现了对各级功能菜单的添加、删除等功能。
- 2) **用户管理**：用户管理可展示所有在企业微信同步过来的用户，也可以在此新建用户，并给所有用户设置角色。
- 3) **角色管理**：可创建多个用户角色，可新增角色、添加角色描述、配置当前角色对应的功能权限。可为创建的角色设置权限。权限是针对当前角色与所有菜单的查看关系。

- 4) **参数管理**:可设置系统所用参数，方便后期维护添加、查看参数，
- 5) **字典管理**:对全平台的字段项进行维护，可新增字典内容、查看字段词语、修改词语、添加词语描述等功能

## 4.1.2.1. 功能详细设计

### 4.1.2.1.1. 菜单管理

| 菜单名称   | 图标 | 排序  | 权限标识 | 组件路径 | 状态 | 创建时间                | 操作   |
|--------|----|-----|------|------|----|---------------------|--|
| > 系统管理 | ⚙️ | 1   |      |      | 正常 | 2023-03-08 16:40:21 | <a href="#">修改</a> <a href="#">新增</a> <a href="#">删除</a> |
| > 系统监控 | 👁️ | 2   |      |      | 正常 | 2023-03-08 16:40:21 | <a href="#">修改</a> <a href="#">新增</a> <a href="#">删除</a> |
| > 系统工具 | 📁  | 3   |      |      | 正常 | 2023-03-08 16:40:21 | <a href="#">修改</a> <a href="#">新增</a> <a href="#">删除</a> |
| > 照排设置 | 📄  | 100 |      |      | 正常 | 2023-03-20 11:26:24 | <a href="#">修改</a> <a href="#">新增</a> <a href="#">删除</a> |

#### (1) 功能描述

**查询**: 点击左侧导航树，将点击的节点详细信息展示在右侧表格中。

**编辑**: 选中某个菜单点击编辑按钮，右侧显示出该菜单的详细内容，可修改菜单名称、地址等及上级所属菜单。

**新增**: 选中某个菜单点击添加按钮，默认为当前菜单添加下级菜单，也可手动修改父级节点来完成新增操作。

#### (2) 主要处理流程

#### (3) 功能实现

**查询**:

1. 在左侧导航树中点击任意节点名称，该条信息的详细信息会在右侧显示。

**编辑**:

1. 在左侧导航树中点击任意节点名称，点击修改该条信息的详细信息会在右侧显示且可修改，修改字段，标题，权限标识，图标，类型，排序，前端组件，

前端地址，路由缓冲，当修改完某个字段进入下一个操作后，js 会对修改的内容进行必填和格式校验，到 js 全部校验通过后点击更新按钮调用后台接口（菜单管理修改）。

2. update\_time 取系统当前时间，del\_flag 默认赋值为 0，然后与前端传输过来的参数一起修改表 sys\_menu 中对应的数据。

3. 调用接口日志，将本次操作记录到日志中，操作人员，操作时间，操作类型，操作模块等信息。

4. 返回前端界面信息，code: 2000, msg: 修改成功。

#### **新增：**

1. 在左侧导航树中点击任意节点名称，点击新增则该名称的 id 会带到右侧的父级节点，节点 id，标题，权限标识，图标，类型，排序，前端组件，前端地址，路由缓冲信息空白，当填写完某个字段进入下一个操作后，js 会对填写的内容进行必填和格式校验，到 js 全部校验通过后点击保存按钮调用后台接口（菜单管理新增）。

2. create\_time 取系统当前时间，del\_flag 默认赋值为 0，然后与前端传输过来的参数一起存到 sys\_menu 表中。

3. 调用接口日志，将本次操作记录到日志中，操作人员，操作时间，操作类型，操作模块等信息。

4. 返回前端界面信息，code: 2000, msg: 修改成功。

#### **删除：**

1. 选中某个节点名称，点击删除，弹出确认删除的弹窗，当点击确认按钮后调用后台删除接口（菜单管理删除）。

2. 后台接收到需删除信息的 id 进行数据库的逻辑删除，update\_time 取当前系统时间，del\_flag 赋值为 1. 然后进行对应数据的修改。