

智慧幕墙：数据集管理平台系统架构与实现

System Architecture and Implementation of Intelligent Curtain Wall Dataset Management Platform

项目成员：林继申、刘淑仪、中谷天音

指导教师：黄杰

同济大学计算机科学与技术学院

2024 年 12 月 11 日



目录

Contents

1

设计原则

Design Principles

2

技术选型

Technology Selection

3

系统架构与安全性

System Architecture and Security

4

接口设计

Interface Design

5

数据流图

Data Flow Diagram

6

平台展示

Platform Presentation

01

设计原则

Design Principles





1. 高内聚低耦合

高内聚 意味着模块内部的功能紧密相关，职责单一，便于理解和维护。例如，在Spring Boot中，一个Service类应专注于处理特定的业务逻辑，而不是混杂多个职责。

低耦合 则指模块之间的依赖关系尽量减少，通过接口或事件机制进行通信。例如，前端Vue组件之间通过Vuex状态管理进行通信，而不是直接相互引用。

```
@Service
public class OssService {

    private final OSS ossClient;

    @Autowired
    public OssService(OSS ossClient) {
        this.ossClient = ossClient;
    }

    public byte[] getObject(String bucketName, String objectKey) { ...
    }

    public String putObject(String bucketName, String objectKey, MultipartFile file) throws IOException { ...
    }
}

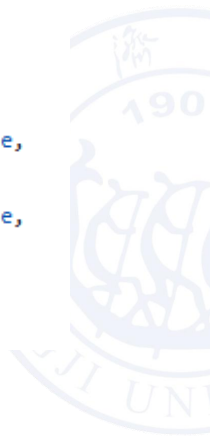
@Service
public class SshService {

    public InputStream[] executeCommand(String host, int port, String username, String password, String command) throws Exception {
        JSch jsch = new JSch();
        Session session = jsch.getSession(username, host, port);
        session.setPassword(password);
        session.setConfig("StrictHostKeyChecking", "no");
        session.connect();

        ChannelExec channel = (ChannelExec) session.openChannel("exec");
        channel.setCommand(command);
        InputStream inputStream = channel.getInputStream();
        InputStream errorStream = channel.getErrStream();
        channel.connect();

        return new InputStream[]{inputStream, errorStream};
    }
}
```

```
state: {
    bucket: '',
    accessKeyId: '',
    headerVisible: true,
    path: '',
    ossList: [],
    loading: false,
    CodeEditor: {
        size: '',
        code: '',
        path: ''
    },
    language: {},
    editorShow: false,
    uploadList: [],
    transmissionShow: false,
    selections: [],
    copys: [],
    copyPath: '',
    copyNum: 0,
    copyTotal: 0,
    copyVisible: false,
    deleteNum: 0,
    deleteConfirm: false,
    deleteTotal: 0,
    deleteVisible: false,
    state: false,
    shear: null
},
```





```
@PostMapping("/upload/**")
public ResponseEntity<String> uploadFile(@RequestParam("file") MultipartFile file,
                                         @RequestParam("userName") String userName,
                                         @RequestParam("password") String password,
                                         HttpServletRequest request) {

    // Authenticate user
    AuthenticationController.AuthenticationRequest authRequest = new AuthenticationController.AuthenticationRequest(userName, password);
    AuthenticationController.AuthenticationResponse authResponse = authenticationController.authenticate(authRequest);

    // Check if authentication failed
    if (StringUtils.isEmpty(authResponse.accessKeyId()) || StringUtils.isEmpty(authResponse.accessKeySecret())) {
        return ResponseEntity.status(401).body("Authentication failed: Invalid username or password");
    }

    // Generate object key for the uploaded file
    String fullPath = request.getRequestURI().replace(request.getContextPath(), "");
    String objectKey = userName + '/' + fullPath.substring("/oss/upload/".length());

    // Validate objectKey
    if (!isValidObjectKey(objectKey)) {
        return ResponseEntity.status(400).body("Invalid object key format: Directories must contain only [A-Za-z0-9-] and filenames must contain only [A-Za-z0-9-.]");
    }

    // Upload file to OSS
    try {
        String uploadedFileKey = ossService.putObject(bucket, objectKey, file);
        String downloadUrl = "http://110.42.214.164:9000/oss/download/" + uploadedFileKey;
        return ResponseEntity.ok(downloadUrl);
    } catch (IOException e) {
        return ResponseEntity.status(500).body("File upload failed: " + e.getMessage());
    }
}
```

2. 错误处理机制

有效的错误处理机制能够提升用户体验和系统稳定性。

后端Spring Boot: 定义明确的错误码和错误信息，便于前端识别和处理不同类型的错误。

前端Vue: 处理后端返回的错误码，进行相应的提示或处理。



3. 模块化原则

前端Vue：利用Vue的组件化特性，将界面拆分为独立、可复用的组件，提升开发效率和代码可维护性。同时使用Vue Router进行页面路由管理，确保各模块之间的独立性。

后端Spring Boot：按照功能或业务领域划分模块，如用户管理、数据集管理、权限控制等，每个模块独立开发和维护。

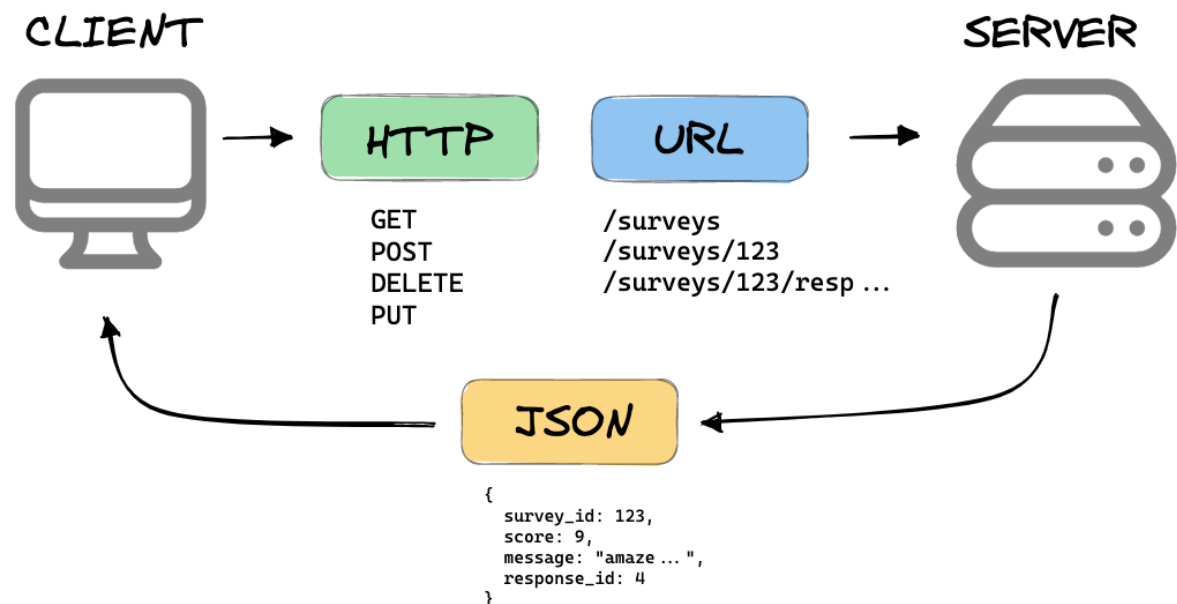
> public	
▼ src	
> api	API
▼ components	组件
Editor.vue	文本浏览
Header.vue	顶部导航
List.vue	文件列表
Operation.vue	文件操作
Transmission.vue	文件传输
▼ pages	页面
HomePage.vue	
LoginPage.vue	
> router	路由
> store	状态管理
> tool	
App.vue	
element-ui.js	
main.js	

> .mvn	
▼ src	
> main	
▼ java/intelligentcurtainwall/ossmanagement	
▼ config	配置
OssConfig.java	OSS配置
WebConfig.java	网络配置
▼ controller	控制器
AuthenticationController.java	登录鉴权控制器
DeploymentLogsController.java	部署日志控制器
DockerLogsController.java	输出日志控制器
OssController.java	OSS控制器
▼ service	服务
OssService.java	OSS服务
SshService.java	SSH服务
OssManagementApplication.java	
> resources	
> test/java/intelligentcurtainwall/ossmanagement	



4. RESTful API设计

遵循**RESTful设计原则**，使前后端交互更加清晰和高效。使用标准的HTTP方法（GET、POST、PUT、DELETE）对应相关操作。设计清晰的URL结构，反映资源之间的关系。利用状态码传达操作结果，增强接口的可理解性。



5. 可扩展性和可维护性

设计系统时考虑未来的扩展需求和维护便利性。

- **代码规范**：遵循统一的编码规范和最佳实践，确保代码一致性。
- **文档化**：编写详细的代码文档和API文档，方便团队协作。



02

技术选型

Technology Selection



技术选型 | Technology Selection



前端



Vue.js



Element

后端



Java

AXIOS



Spring Boot

持久化



阿里云
对象存储服务 (OSS)

部署



webpack



docker



03

系统架构与安全性

System Architecture and Security





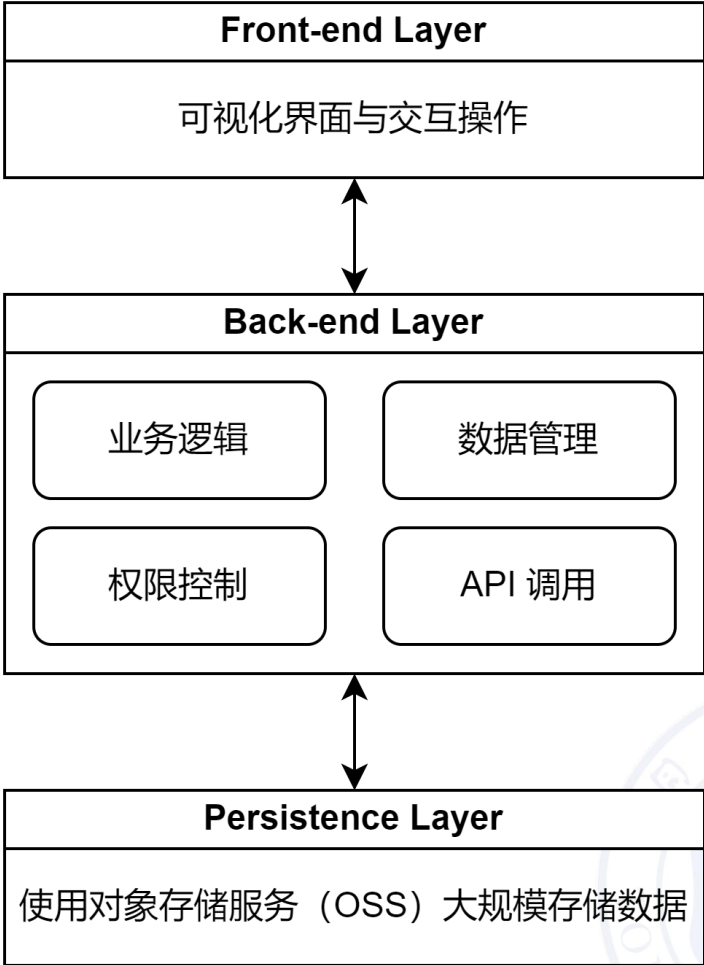
1. 系统架构

Front-end Layer (前端层)： 前端负责将用户的操作通过可视化界面进行呈现，并将请求发送至后端进行处理。

Back-end Layer (后端层)：

- **业务逻辑：** 根据平台功能需求，对用户的操作进行处理。
- **数据管理：** 负责数据集的管理、维护与相关操作。
- **权限控制：** 确保只有符合权限要求的用户才能访问和操作特定数据。
- **API 调用：** 为前端层提供统一的接口方便调用。

Persistence Layer (持久化层)： 使用阿里云对象存储（OSS）服务大规模存储数据。





2. 安全性

阿里云RAM访问控制

- 利用阿里云RAM（Resource Access Management）对用户、角色及访问策略进行严格管控。
- 基于**最小权限原则**，仅授予符合条件的用户或角色访问相应数据集和API的权限。

定时备份

- 对数据集及相关元数据进行定期自动化备份，防止数据丢失或损坏。
- 在出现故障或数据被误删时，可快速通过备份进行数据恢复，保证数据安全性与业务连续性。
- **备份策略：每周一00:00进行备份，每个备份保留30天**

RAM访问控制策略

组管理			
组成员管理	权限管理		
添加组成员			
输入用户登录名称或显示名称			
<div>Q</div>			
<div></div>			
<input type="checkbox"/>	用户登录名称/显示名称	加入时间	操作
<input type="checkbox"/>	icw-vibration-detection@minmuslin.onaliyun.com 智慧幕墙：幕墙震动数据检测与展示系统	2024年11月21日 16:32:06	移出用户组
<input type="checkbox"/>	icw-stain-detection@minmuslin.onaliyun.com 智慧幕墙：石材幕墙污渍检测系统	2024年11月21日 16:32:06	移出用户组
<input type="checkbox"/>	icw-spalling-detection@minmuslin.onaliyun.com 智慧幕墙：玻璃幕墙爆裂检测系统	2024年11月21日 16:32:05	移出用户组
<input type="checkbox"/>	icw-resilience-assessment@minmuslin.onaliyun.com 智慧幕墙：幕墙韧性评估软件系统	2024年11月21日 16:32:05	移出用户组
<input type="checkbox"/>	icw-oss-management@minmuslin.onaliyun.com 智慧幕墙：对象存储OS管理控制系统	2024年11月21日 16:32:06	移出用户组
<input type="checkbox"/>	icw-modeling-communication@minmuslin.onaliyun.com 智慧幕墙：无人机采集数据的3D建模与通讯系统	2024年11月21日 16:32:06	移出用户组
<input type="checkbox"/>	icw-mobile-data@minmuslin.onaliyun.com 智慧幕墙：移动端幕墙数据采集与展示系统	2024年11月21日 16:32:05	移出用户组
<input type="checkbox"/>	icw-flatness-detection@minmuslin.onaliyun.com 智慧幕墙：玻璃幕墙平整度检测系统	2024年11月21日 16:32:05	移出用户组
<input type="checkbox"/>	icw-crack-detection@minmuslin.onaliyun.com 智慧幕墙：石材幕墙裂缝检测系统	2024年11月21日 16:32:06	移出用户组
<input type="checkbox"/>	icw-corrosion-detection@minmuslin.onaliyun.com 智慧幕墙：金属幕墙锈蚀污染检测系统	2024年11月21日 16:32:06	移出用户组

定时备份策略

Bucket: tongji-icw			
每个OSS Bucket首次备份享受30天的免费试用期。参见 价格详情 。			
OSS定时备份功能由 云备份 Cloud Backup 提供。相关费用也将计入到云备份的账单中。点击前往 云备份控制台			
当前地域标准型和低频型Bucket: 1		已经产生备份的Bucket: 0	配置备份计划的Bucket: 0
备份计划			
创建备份计划			
OSS Bucket	备份库名/ID	备份策略名称/ID	备份策略详情
tongji-icw	tongji-icw-backup-vault	tongji-icw-backup-policy	执行时间: 在00:00, 仅星期一 保留时间: 1月 特殊保留时间: 不使用 至少保留一个备份: 已开启 自动归档: 不使用 备份异地复制: 不使用 备份锁定: 不使用 备份点检测病毒: 不使用 关联资源: 不使用

04

接口设计

Interface Design



1. 上传文件至OSS

接口描述：通过提供文件及相关用户信息，将文件上传至 OSS 存储。

URL: POST /oss/upload/{文件路径}

传入参数:

参数名	类型	必填	描述
file	MultipartFile	是	要上传的文件
userName	String	是	账号（用于身份验证）
password	String	是	密码（用于身份验证）

响应：

- **成功：**返回文件的下载链接。
 - **HTTP 状态码：**200 OK
 - **响应体：**"downloadUrl": "http://<服务域名>/oss/download/<文件路径>"
- **失败：**返回 HTTP 错误状态码。
 - **401 Unauthorized:** 身份验证失败。
 - **400 Bad Request:** 对象键格式无效（如路径或文件名非法）。
 - **500 Internal Server Error:** 文件上传失败。



2. 从OSS下载文件

接口描述：通过提供文件的路径，从 OSS 存储中下载指定文件。

URL：GET /oss/download/{文件路径}

请求参数：无直接请求参数。

响应：

- **成功：**返回文件的二进制数据，同时响应头包含文件下载的必要信息。
 - **HTTP 状态码：**200 OK
 - **响应头：**Content-Disposition: attachment; filename=<文件名>
 - **响应体：**文件的二进制内容
- **失败：**返回 HTTP 错误状态码。
 - **404 Not Found:** 文件不存在。



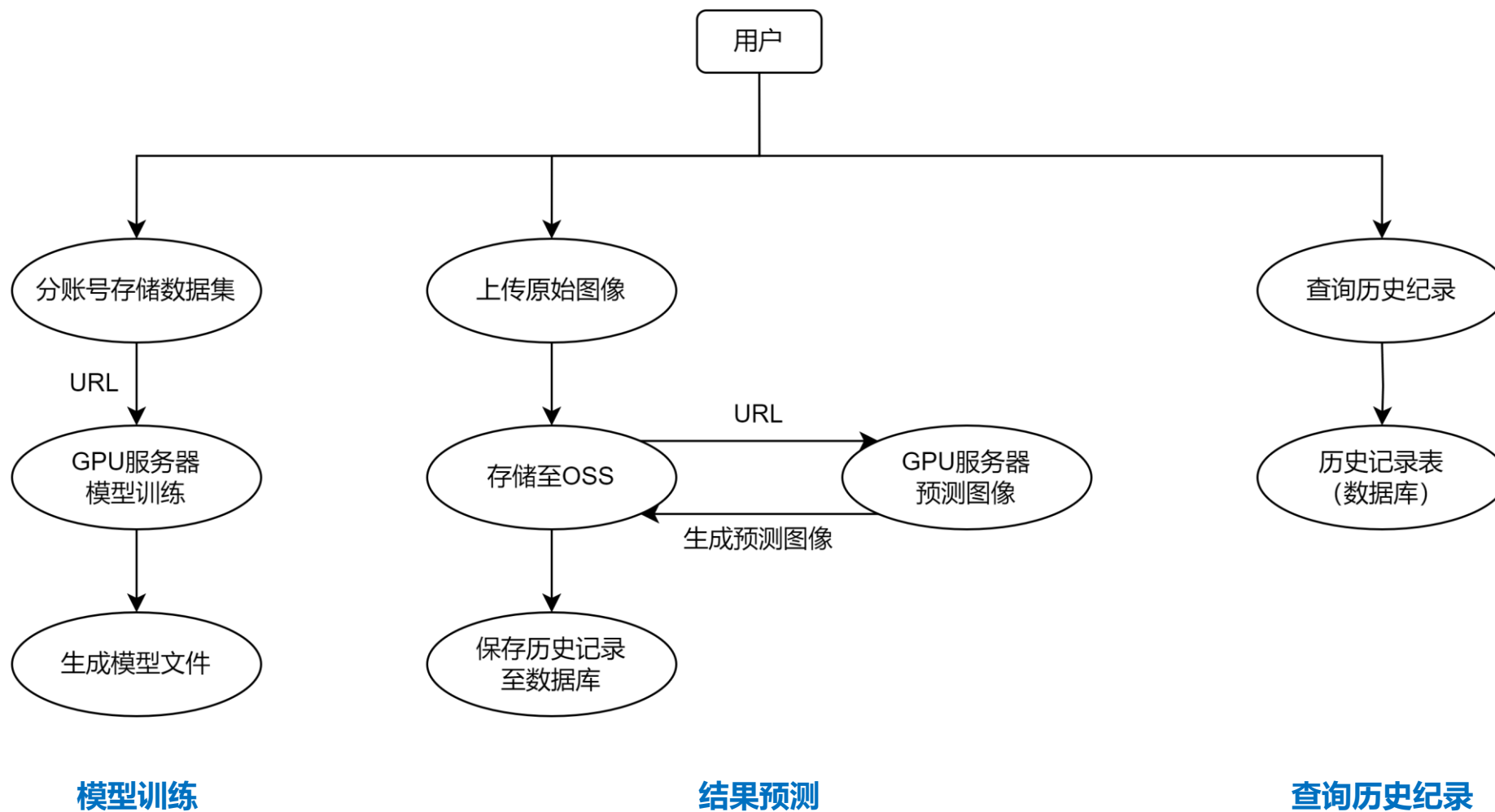
05

数据流图

Data Flow Diagram



数据流图 | Data Flow Diagram




06

平台展示

Platform Presentation

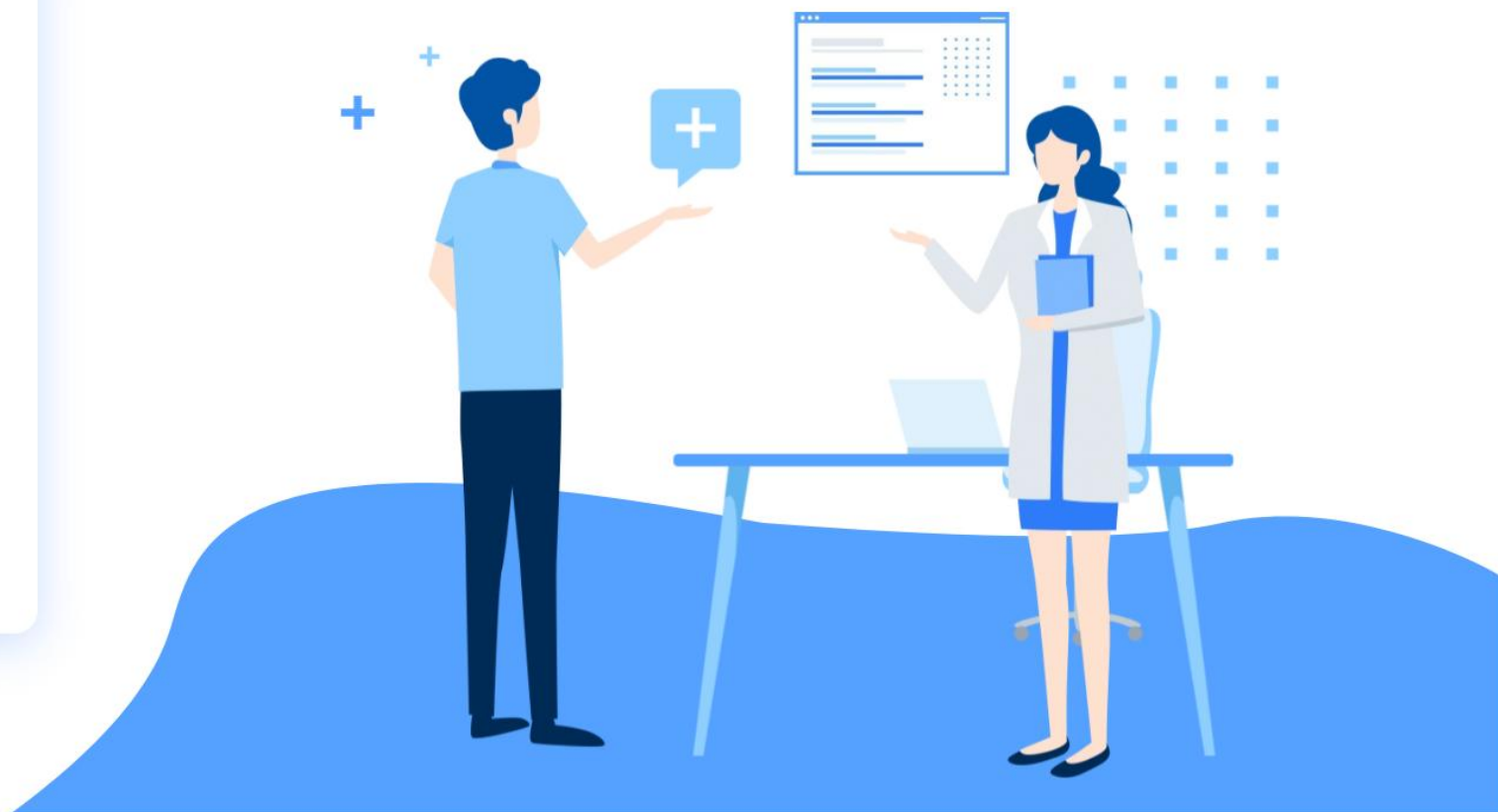




欢迎登录系统

智慧幕墙数据集管理平台

登录



平台展示 | Platform Presentation












智慧幕墙数据集管理平台

智慧幕墙数据集管理平台管理员 注销

← ↺ 🏠 oss://tongji-icw/

欢迎使用智慧幕墙数据集管理平台!

<input type="checkbox"/>	数据集	子系统
<input type="checkbox"/>	 corrosion-detection	金属幕墙锈蚀污渍检测系统
<input type="checkbox"/>	 crack-detection	石材幕墙裂缝检测系统
<input type="checkbox"/>	 flatness-detection	玻璃幕墙平整度检测系统
<input type="checkbox"/>	 mobile-data	移动端幕墙数据采集与展示系统
<input type="checkbox"/>	 modeling-communication	无人机采集数据的 3D 建模与通讯系统
<input type="checkbox"/>	 resilience-assessment	幕墙韧性评估软件系统
<input type="checkbox"/>	 spalling-detection	玻璃幕墙爆裂检测系统
<input type="checkbox"/>	 stain-detection	石材幕墙污渍检测系统
<input type="checkbox"/>	 vibration-detection	幕墙震动数据检测与展示系统

← ↺ 🏠 oss://tongji-icw/stain-detection/output/

📁 文件

+ 目录

📌 地址

📄 复制

✂ 剪切

✖ 删除

📋 粘贴 (60)

✖



平台展示 | Platform Presentation



智慧幕墙数据集管理平台

智慧幕墙数据集管理平台管理员 注销

oss://tongji-icw/stain-detection/upload/

文件 目录 地址 复制 剪切 删除

名称

1733834645057-ts.jpg

1733834689530-ts.jpg

1733834839188-ts.jpg

1733886466150-ts.jpg

1733886477219-ts.jpg

test.jpg

test2.jpg

图像预览

关闭

类型 / 大小	修改时间	操作
344.71 KB	2024-12-10 20:44	获取地址 下载 重命名 删除
344.71 KB	2024-12-10 20:44	获取地址 下载 重命名 删除
344.71 KB	2024-12-10 20:47	获取地址 下载 重命名 删除
344.71 KB	2024-12-11 11:08	获取地址 下载 重命名 删除
344.71 KB	2024-12-11 11:08	获取地址 下载 重命名 删除
909.10 KB	2024-12-05 19:47	获取地址 下载 重命名 删除
996.10 KB	2024-12-05 19:54	获取地址 下载 重命名 删除

版权所有 © 2024 同济大学 - 智慧幕墙数据集管理平台 (前端仓库 后端仓库)

0 / 0

平台展示 | Platform Presentation



智慧幕墙数据集管理平台

智慧幕墙数据集管理平台管理员 注销

oss://tongji-icw/stain-detection/output/

文件 目录 地址 复制 剪切 删除

名称

annotated-image20241210205647.jpg

annotated-image20241210205921.jpg

annotated-image20241210210130.jpg

annotated-image20241210210508.jpg

annotated-image20241210210932.jpg

annotated-image20241210211145.jpg

result-image20241208201806-1.jpg

result-image20241208201806-2.jpg

result-image20241209192027-1.jpg

result-image20241209192027-2.jpg

result-image20241210204158-1.jpg

result-image20241210204158-2.jpg

result-image20241210204806-1.jpg

result-image20241210204806-2.jpg

result-image20241210205050-1.jpg

图像预览

关闭

类型 / 大小	修改时间	操作
1.01 MB	2024-12-10 20:56	获取地址 下载 重命名 删除
1.01 MB	2024-12-10 20:59	获取地址 下载 重命名 删除
1.01 MB	2024-12-10 21:01	获取地址 下载 重命名 删除
1.01 MB	2024-12-10 21:05	获取地址 下载 重命名 删除
1.01 MB	2024-12-10 21:09	获取地址 下载 重命名 删除
1.01 MB	2024-12-10 21:11	获取地址 下载 重命名 删除
118.90 KB	2024-12-08 20:18	获取地址 下载 重命名 删除
217.77 KB	2024-12-08 20:18	获取地址 下载 重命名 删除
118.90 KB	2024-12-09 19:20	获取地址 下载 重命名 删除
217.77 KB	2024-12-09 19:20	获取地址 下载 重命名 删除
118.90 KB	2024-12-10 20:41	获取地址 下载 重命名 删除
217.77 KB	2024-12-10 20:42	获取地址 下载 重命名 删除
118.90 KB	2024-12-10 20:48	获取地址 下载 重命名 删除
217.77 KB	2024-12-10 20:48	获取地址 下载 重命名 删除
118.90 KB	2024-12-10 20:50	获取地址 下载 重命名 删除

版权所有 © 2024 同济大学 - 智慧幕墙数据集管理平台 (前端仓库 后端仓库)

0 / 0

The background of the slide features a large, light blue circular seal of Tongji University. The seal contains the university's name in Chinese characters '同济大学' and English 'TONGJI UNIVERSITY' around the perimeter, with the year '1907' in the center. A blue rectangular frame is positioned on the right side of the slide, partially overlapping the seal.

感谢聆听！

智慧幕墙：数据集管理平台系统架构与实现

System Architecture and Implementation of Intelligent Curtain Wall Dataset Management Platform

项目成员：林继申、刘淑仪、中谷天音

指导教师：黄杰

同济大学计算机科学与技术学院

2024 年 12 月 11 日