

基于 IIS 的考试系统技术设计与实现

陈翔

广州气象卫星地面站, 广东 广州 510640

DOI: 10.61369/TACS.2025070019

摘要 : 系统在 IIS 环境下完成考试的调试与上线, 文中围绕需求要点、分层架构、关键模块(评分、Excel 导入、鉴权 / CORS)、数据模型与事务一致性、IIS 部署与可观测性、安全与性能优化展开, 给出可复现的技术设计方案, 为同类考试测评系统提供工程化参考。

关键词 : IIS; 考试系统; ASP.NET Core 7; MySQL 5.7

Technical Design and Implementation of an IIS-based Examination System

Chen Xiang

Guangzhou Meteorological Satellite Ground Station, Guangzhou, Guangdong 510640

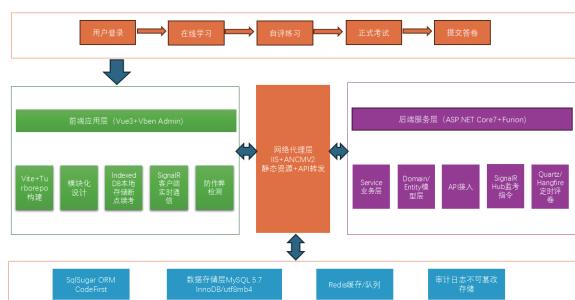
Abstract : The system is IIS Complete the debugging and launch of the exam under certain circumstances, focusing on the requirements, hierarchical architecture, and key modules (grading, etc.) in the article Excel Import and authentication/ CORS)、Data model and transaction consistency IIS Deployment and observability, security and performance optimization are carried out to develop a technical design solution that can recover cash, providing engineering reference for similar examination and evaluation systems.

Keywords : IIS; examination system; ASP.NET Core 7; MySQL 5.7

“风云”系列气象卫星地面系统是我国气象卫星运行管理、数据接收处理和产品分发的核心平台。它由指控中心、接收站、数据处理中心、分发与应用平台四大子系统组成, 承担遥测遥控、观测数据加工、气象产品生成与服务等任务, 为气象预报、灾害监测和科研提供高质量卫星数据与产品。随着风云卫星系统更新迭代, 卫星技术理论与操作流程不断丰富, 对技术人员的知识水平和实操能力提出了更高要求, 传统线下培训耗时长、资源分散且考评反馈滞后, 难以满足高效、可追溯的培训需求。因此, 本项目旨在构建一个集“文档管理 + 在线学习 + 自评考评”于一体的信息化平台, 实现文档规范化存储、培训内容在线化交付和考评流程闭环管理。

一、系统架构设计

系统采用前后端分离架构: 前端为 Vue 3 + Vben Admin + Vite (pnpm/turborepo 构建), 后端为 ASP.NET Core 7 (Furion 框架风格, Service 即 API), 数据库为 MySQL 5.7, ORM 选用 SqlSugar 并通过 CodeFirst 初始化^[1]。



(一) 前端设计

系统前端采用单页应用架构, 基于 Vue 3 和 Vben Admin 快速搭建管理后台风格的考试平台界面。系统以 Turborepo 管理多包单仓, 使用 pnpm workspace 做依赖管理和并行构建, 便于团队协同与模块复用。设计采用模块化设计, 按功能拆分为 packages/app (主应用)、packages/ui (设计系统与公共组件)、packages/shared (接口 types、工具函数、hooks)。ui 包封装 ExamTimer、ProTable、QuestionCard 等通用控件, 保证风格统一与可维护性。设计采用模块化管理考试会话、计时器与答题草稿。关键数据周期性写入 IndexedDB 或 localForage, 以支持断点续考与离线答题恢复。通过 WebSocket/SignalR 实现监考指令、即时评分与心跳检测; 实现自动重连与消息队列。防作弊逻辑在前端检测聚焦切换、多开、截图与摄像头状态, 并将行为事件上报后端审计。

图1 系统架构设计流程图

(二) 后端设计

系统后端基于 ASP.NET Core 7，采模块化 API 设计，强调业务服务化、依赖注入与约定优于配置。后端设计按层划分：API 层，负责控制器 / 路由请求出口；Service 层，负责封装业务逻辑，Domain/Entity 层，负责定义模型及数据校验；Infrastructure 层，提供持久化、缓存、消息与外部接口实现。

安全策略采用短期 JWT + refresh token，结合刷新、黑名单与强制登出机制；同时在 API 网关 / 中间件开启速率限制、IP 白名单与全局异常过滤。实时功能通过 SignalR 实现监考指令与即时评分，支持心跳、重连与消息队列化。缓存采用 Redis，配合分布式锁与缓存穿透、失效策略，提升并发场景下的一致性与性能。

为防作弊，服务端记录行为日志、聚焦切换事件与答题轨迹，提供审计查询与回放接口；关键操作加密传输并写入不可篡改的审计链。后台任务使用 Quartz/Hangfire 处理定时评卷、数据回写与清理任务。开发体验集成 Swagger/OpenAPI、API 版本管理与自动化文档生成，测试覆盖单元与集成测试。

(三) 数据库设计

后端数据库采用 MySQL 5.7，ORM 选用 SqlSugar 并通过 CodeFirst 初始化，以便快速建表、维护实体同步和提高开发效率。系统在 Infrastructure 层封装 SqlSugarClient 配置：统一在 Program.cs 注册连接字符串、AOP 日志（OnLogExecuting）、全局序列化与缓存策略、连接池与重试策略。使用 CodeFirst 的 CreateDatabase/InitTables 在部署测试环境下自动建库建表，结合可控的迁移脚本审查变更信息。针对 MySQL 5.7 的特点：默认使用 InnoDB 引擎、字符集设为 utf8mb4 与合适校对集；读写分离通过 SqlSugar 配置，结合 Redis 缓存和分布式锁提高并发性能与一致性。

二、关键模块实现

关键模块围绕鉴权、题库、考试管理、答题会话、阅卷评分、防作弊审计、缓存一致性、后台任务与基础设施展开。每个模块既要考虑业务正确性，也要兼顾并发、可恢复性与可观测性。

(一) 鉴权模块

使用短期 JWT + Refresh Token 机制，认证逻辑位于认证中间件，统一处理 Token 校验与续签。登录、登出、强制下线与刷新均写入审计表。对敏感接口加二次验证（如短信或动态口令），并支持黑名单或 Token 版本号策略以实现强制失效。

(二) 题库模块

模块采用 SqlSugar CodeFirst 定义，在 Repository 层设置增删改查与批量导入接口。题库支持按标签 / 难度 / 知识点分页查询、批量操作与导出，并预留相似题检索或嵌入向量检索的扩展点。所有写操作通过事务封装并记录变更历史。

(三) 考试管理

考试管理由 ExamService 模块负责，进行试卷的发布、回

收与状态流转。采用 SignalR 实时通信，服务端 Hub 与 Redis 作会话缓存，定期将会话快照与答题进度落库以支持断点续考。前端也会周期性本地持久化（IndexedDB），以应对网络抖动^[2]。重连时按快照合并变更并防止重复提交。

(四) 阅卷与评分

阅卷与评分采用异步任务队列 + 工作池的方式，结合自动评分引擎与人工阅卷流程，实现高并发、可伸缩、可观测与可回滚的评分体系。评分流程记录操作日志和时间线，支持回滚与重新评分。

三、部署实施

在部署前请先在服务器上启用 IIS 管理控制台、应用程序开发及常见 HTTP 功能，并安装 ASP.NET Core Hosting Bundle（针对 .NET 7）。安装完成并重启后，在模块列表确认 AspNetCoreModuleV2 可用，以便 ANCM 将请求转发到 Kestrel^[3]。

(一) 前端部署（示例端口：3000）

在构建前于 .env.production 中将 VITE_API_BASE 指向后端地址（例如 http://127.0.0.1:8080），关闭 PWA。使用 pnpm 安装并通过 Turborepo 构建：pnpm install → pnpm turbo run build，使其位于 dist 目录。IIS 新建站点指向 dist 根目录，添加包含 SPA 回退规则的 web.config（需先安装 URL Rewrite 模块），确保前端路由在刷新时能回退到 index.html^[4]。

(二) 后端部署（示例端口：8080）

通过 dotnet publish 生成发布目录，在 IIS 中新建站点并绑定该目录，应用程序池设置为“无托管代码”。web.config 配置使用 ANCM v2 将请求代理到 Kestrel^[5]。程序在 Program/Startup 中应配置好 CORS 白名单与 JWT 鉴权策略；将 Uploads 等文件目录挂载为静态目录，并赋予应用池最小写入权限；若目录缺失，应用应能在启动时自动创建并记录日志。

(三) 数据库与初始化

数据库创建时统一使用 utf8mb4 字符集和 utf8mb4_unicode_ci 校对规则，建立只具最小权限且限制主机访问的业务账号（仅授权必要的 DML/DDL/INDEX 权限）。首次部署可通过受控接口（例如 /api/dbinit/init）触发 SqlSugar 的 CodeFirst 建表与数据写入，初始化完成关闭该接口或添加鉴权以防止误用^[6]。

四、应用成效

系统在 IIS 环境下完成设计与实现。首先在可用性与稳定性方面，在 IIS 上借助 ANCM v2 运行 Kestrel，配合独立应用程序探测实现自动重启与故障隔离。进程具备自动恢复与重启功能，大幅降低考试期间的服务中断，必要时能够迅速回滚，缩短了修复周期。其次在性能上，前端通过 Vite 打包成静态资源并交由 IIS 提供服务，配合压缩和 CDN 提升传输效率，用户体验显著改善；后端则凭借 Redis、SqlSugar 与异步评分工作池拆分负载，增强

并发处理能力并在流量峰值期间保持评分延迟可控。安全与合规性方面，系统采用 HTTPS、JWT + refresh token、最小权限数据库账户与审计链设计，前后端联动记录不可篡改的答题与操作轨迹，为异议处理和合规审计提供可靠证据^[7-8]。

本系统通过功能测试、安全测试、性能测试等，能够实现技

术人员的快速上岗、动态掌握学习进展、自动化考评培训评估，达到了预期的目标^[9-10]。

参考文献

- [1] 乔俊玲. 基于 Struts+Hibernate+Ajax 框架开发网上考试系统的设计与实现 [J]. 电脑知识与技术, 2019, 15(8):61–62.
- [2] 褚娟. 基于微服务的在线考试系统分析与设计 [J]. 现代信息科技, 2024, 8(12):76–80.
- [3] 李金平, 张春青. 基于 ASP+SQL 的远程 MIS 系统安全特性分析与研究 [J]. 北方交通大学学报, 2002, 26(6):76–79.
- [4] 李晓丽, 赵丽辉. 基于 ASP 计算机应用基础课程网上考试系统设计 [J]. 电子技术与软件工程, 2017(18):186–186.
- [5] 虞飞舟. ASP 技术在信息发布中的应用研究 [D]. 北京: 北京交通大学, 2000, 02:23–24.
- [6] 赵红云, 马蚕祥, 赵藕祥, 何根宏. 一种在线考试系统设计与实现 [J]. 电子科技期刊. 2001, 21: 24–27.
- [7] 漆礼根. 基于 WEB 的在线考试系统设计与实现 [D]. 南昌大学, 2010.
- [8] 田军, 孟祥娟. 基于 ASP.NET 的网上考试系统设计与实现 [J]. 电脑与信息技术, 2012, 20(1):3.
- [9] 胡磊. 基于 ASP.NET 在线考试系统设计与实现 [D]. 复旦大学, 2010.
- [10] 史杨. 基于 PHP 技术网络考试系统的开发 [J]. 电脑知识与技术: 学术交流, 2011.