



APUSIC
固若长城
睿比世界

API文档

金蝶Apusic应用服务器V10标准版

版权声明

本文档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本文档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本文档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本文档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本文档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本文档如有更新，不另行通知。对本文档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

- 1 AlertConfigController
- 2 AppClassLibController
- 3 ApplIncrementController
- 4 ApplicationController
- 5 AuditLogController
- 6 AuthController
- 7 BaselineCheckController
- 8 CaptchaController
- 9 CertificateStoreController
- 10 ClassAnalysisController
- 11 ClassLoaderController
- 12 CracController
- 13 DashboardController
- 14 DatasourceController
- 15 HttpChannelController
- 16 InspectionController
- 17 InvokeChainTraceController
- 18 JmxConfigController
- 19 JndiResourceController
- 20 JvmConfigController
- 21 LicenseController
- 22 LogController
- 23 LoginLogController
- 24 MetricsController
- 25 MonitorConfigController
- 26 MonitorDataController
- 27 NotificationController
- 28 PasswordController
- 29 PatchController
- 30 ReplayController
- 31 ResourceController
- 32 RoleController

- 33 ScriptRecordingController
- 34 SecurityConfigController
- 35 SecurityRealmController
- 36 SessionManagerController
- 37 SharedLibController
- 38 SnapshotConfigController
- 39 SnapshotController
- 40 SnmpConfigController
- 41 SqlTraceController
- 42 SystemController
- 43 ThreadDiagnosisController
- 44 ThreadPoolController
- 45 ThreadTraceController
- 46 UpgradeHistoryController
- 47 UserController
- 48 UserPermissionController
- 49 VirtualHostController
- 50 附录： HTTP 方法说明

1 AlertConfigController

基础路径: /api/alert-config

方法	路径	函数	参数	返回类型	说明
GET	/api/alert-config	getConfig		Result<AlertConfigDTO>	查询告警配置（开关/通知方式/阈值指标）
PUT	/api/alert-config	saveConfig	@RequestBody AlertConfigDTO dto	Result<Void>	保存告警配置，保存前捕获变更用于审计对比
POST	/api/alert-config/restore-defaults	restoreDefaults		Result<AlertConfigDTO>	恢复告警配置为默认值，记审计

AlertConfigDTO 字段说明

字段名	类型	中文说明
enabled	Integer	告警总开关（0=关 1=开）
intervalMinutes	Integer	告警间隔（分钟，1-527040）
emailEnabled	Integer	邮件通知开关
emailRecipients	String	接收邮件地址，多个用英文分号隔开
emailSmtpHost	String	SMTP 服务器地址
emailSmtpSsl	Integer	SMTP SSL 开关
emailSmtpPort	Integer	SMTP 端口（1-65535）
emailSmtpUser	String	SMTP 发送邮箱
emailSmtpPassword	String	SMTP 密码
snmpEnabled	Integer	SNMP 通知开关
snmpVersion	String	SNMP 版本（v2c/v3）
snmpCommunity	String	SNMP 团体名
snmpAddress	String	SNMP 目标地址

snmpSecurity	String	SNMPv3 安全级别
snmpUsername	String	SNMPv3 用户名
snmpAuthMethod	String	SNMPv3 认证方式
snmpAuthPassword	String	SNMPv3 认证密码
snmpCipher	String	SNMPv3 加密算法
snmpPrivPassword	String	SNMPv3 加密密码
snmpSenderEngineId	String	SNMPv3 发送端 engine ID（只读，仅用于前端展示）
platformEnabled	Integer	告警平台通知开关
platformLevel	String	上报告警平台的告警级别
platformUrl	String	告警平台地址（http/https）
metrics	List	告警指标阈值配置列表（metricCode/threshold/consecutiveCount/unit）

2 AppClassLibController

基础路径: `/api/apps/{appId}/class-libs`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/apps/{appId}/class-libs</code>	list	@PathVariable Long appId, @RequestParam int page, int size	Result<PageResult<Map<String, Object>>>	分页查询应用的类库 (lib 目录 JAR 包) 列表
POST	<code>/api/apps/{appId}/class-libs</code>	upload	@PathVariable String appId, @RequestParam MultipartFile file	Result<Void>	上传 JAR 包到应用类库目录; appId 路径段兼容数字 ID 或应用名 (脚本回放)
DELETE	<code>/api/apps/{appId}/class-libs</code>	delete	@PathVariable String appId, @RequestParam String jarName	Result<Void>	删除应用类库中的指定 JAR 包

3 AppIncrementController

基础路径: `/api/apps/{appId}/increments`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/apps/{appId}/increments</code>	list	@PathVariable Long appId, @RequestParam int page, int size	Result<PageResult<AppIncrement>>	分页查询应用的增量更新包列表
POST	<code>/api/apps/{appId}/increments</code>	upload	@PathVariable String appId, @RequestParam MultipartFile file, String description	Result<Long>	上传增量更新包, 返回增量包 ID; appId 路径段兼容数字 ID 或应用名
DELETE	<code>/api/apps/{appId}/increments/{id}</code>	delete	@PathVariable String appId, Long id	Result<Void>	删除指定增量更新包
POST	<code>/api/apps/{appId}/increments/{id}/apply</code>	apply	@PathVariable String appId, Long id	Result<Void>	应用指定增量更新包到应用

4 ApplicationController

基础路径: `/api/apps`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/apps</code>	list	AppQueryDTO query	Result<PageResult<ApplicationDTO>>	分页查询应用列表 (按名称/上下文 路径/虚拟主机/状 态过滤)
GET	<code>/api/apps/{id}</code>	get	@PathVariable Application app	Result<ApplicationDTO>	按 ID 获取应用详情
GET	<code>/api/apps/{id}/urls</code>	getAppUrls	@PathVariable Application app	Result<List<AppUrlDTO>>	获取应用的访问 URL 列表
GET	<code>/api/apps/check-name</code>	checkAppName	@RequestParam String appName, String virtualHost, Long excludeId	Result<Boolean>	校验应用名是否已存在 (可排除指定 ID)
GET	<code>/api/apps/check-context</code>	checkContextPath	@RequestParam String contextPath, String virtualHost, Long excludeId	Result<Boolean>	校验上下文路径是否已存在 (可排除指定 ID)
GET	<code>/api/apps/check-context-backup</code>	checkContextPathForBackup	@RequestParam String contextPath, String virtualHost, String appName	Result<Boolean>	备份恢复时校验上下文路径冲突
POST	<code>/api/apps</code>	create	@RequestBody ApplicationDTO dto	Result<Long>	创建应用, 返回新应用 ID
PUT	<code>/api/apps/{id}</code>	update	@PathVariable Application app, @RequestBody ApplicationDTO dto	Result<Void>	更新应用配置 (审计记录变更前差异)
DELETE	<code>/api/apps/{id}</code>	delete	@PathVariable Application app, @R	Result<Void>	卸载/删除应用, force=true 时强制删除 (不进回收站)

			equestParam boolean force		
POST	/api/apps/{id}/start	start	@PathVariable Application app	Result<Void>	启动应用
POST	/api/apps/{id}/stop	stop	@PathVariable Application app	Result<Void>	停止应用
POST	/api/apps/{id}/reload	reload	@PathVariable Application app	Result<Void>	重载应用
POST	/api/apps/{id}/redeploy	redeploy	@PathVariable Application app, @RequestParam ApplicationDTO dto, MultipartFile file, String localWarPath, String localDirPath	Result<Void>	重新部署应用（支持上传 war 或指定服务器本地路径）
POST	/api/apps/{id}/deploy	deploy	@PathVariable Application app, @RequestParam MultipartFile file, String localWarPath, String localDirPath	Result<Void>	部署应用包（war 文件/本地 war 路径/本地目录路径三选一）
GET	/api/apps/war-browse	browseWarFiles	@RequestParam String path	Result<ServerFileListResponse>	浏览服务器上可部署的 war 文件目录
GET	/api/apps/backups	getBackups		Result<List<Map<String, Object>>>	获取全部应用备份列表
POST	/api/apps/{id}/backups	createBackup	@PathVariable Application app	Result<Void>	为指定应用创建备份
GET	/api/apps/backups/{backupName}/config	getBackupConfig	@PathVariable String backupName	Result<Map<String, Object>>	获取备份包中的应用配置信息

POST	/api/apps/backup s/{backupName}/re store	restore Backu p	@PathVariable str ing backupName, @RequestBody Ap plicationDTO dto	ResponseEntity<R esult<Void>>	从备份恢复应用, 冲突时返回 409
POST	/api/apps/backup s/{backupName}/re store-force	restore Backu pForce	@PathVariable str ing backupName, @RequestBody Ap plicationDTO dto	Result<Void>	强制从备份恢复应 用（覆盖已有应 用）
GET	/api/apps/backup s/{backupName}/do wnload	downl oadBa ckup	@PathVariable str ing backupName	ResponseEntity<R esource>	下载备份包文件
DELET E	/api/apps/backup s/{backupName}	delete Backu p	@PathVariable str ing backupName	Result<Void>	删除备份包
GET	/api/apps/recycle d	getRec ycledA pps	AppQueryDTO query	Result<List<Recyc ledAppDTO>>	获取回收站中的应 用列表
POST	/api/apps/recycl e/{fileName}/rest ore	restore	@PathVariable str ing fileName, @R equestBody Appli cationDTO dto	Result<Void>	从回收站恢复应用
GET	/api/apps/recycl e/{fileName}/conf ig	getRec ycleAr chiveC onfig	@PathVariable str ing fileName	Result<Map<Strin g, Object>>	获取回收站归档中 的应用配置
DELET E	/api/apps/recycl e/{fileName}/perm anent	perma nently Delete	@PathVariable str ing fileName	Result<Void>	彻底删除回收站中 的应用（不可恢 复）
DELET E	/api/apps/externa l	delete Extern al	@RequestBody Ex ternalAppInfo info	Result<Void>	删除外部部署的应 用（非管控台部 署）
POST	/api/apps/externa l/backups	create Extern alBack up	@RequestBody Ex ternalAppInfo info	Result<Map<Strin g, Object>>	为外部部署的应用 创建备份

POST	/api/apps/external/redeploy	redeployExternal	@RequestPart ExternalAppInfo info, MultipartFile file, String localWarPath	Result<Void>	重新部署外部应用
POST	/api/apps/external/reload	reloadExternal	@RequestBody ExternalAppInfo info	Result<Void>	重载外部应用
POST	/api/apps/external/redeploy-with-props	redeployExternalWithProps	@RequestPart ExternalAppInfo info, ApplicationDTO dto, MultipartFile file, String localWarPath	Result<Void>	携带配置属性重新部署外部应用
GET	/api/apps/recycle/config	getRecycleConfig		Result<Map<String, Object>>	获取回收站配置 (保留路径/保留天数等)
PUT	/api/apps/recycle/config	updateRecycleConfig	@RequestBody Map<String, Object> config	Result<?>	保存回收站配置, 校验失败返回 400 及字段错误
GET	/api/apps/backup/config	getBackupConfig		Result<Map<String, Object>>	获取备份配置
PUT	/api/apps/backup/config	updateBackupConfig	@RequestBody Map<String, Object> config	Result<?>	保存备份配置, 校验失败返回 400 及字段错误

ApplicationDTO 字段说明

字段名	类型	中文说明
id	Long	应用 ID
appName	String	应用名称
contextPath	String	应用上下文路径

status	Integer	应用运行状态
classLoaderOrder	String	类加载顺序
jspPrecompile	Boolean	是否预编译 JSP
virtualHost	String	所属虚拟主机
sessionManager	String	会话管理器配置
deployOrder	Integer	部署顺序
sharedLib	String	共享库名称
sharedLibId	Long	共享库 ID
sharedLibPriority	String	共享库加载优先级
deployTimeout	Integer	部署超时时间（秒）
classLevel	Integer	密级标识（秘密/机密/绝密）
description	String	应用描述
deployTime	String	部署时间
recycledTime	String	回收时间
securityDomain	String	安全域
deployProps	Map<String, String>	部署属性
managedByConsole	Boolean	是否由管控台部署（false=外部部署）
docBase	String	应用文档根目录
deploySource	String	部署来源：CONSOLE（管控台部署）/MANUAL_CONFIG（手动配置）/AUTO_DEPLOY（自动扫描部署）
originalVirtualHost	String	修改虚拟主机前的原虚拟主机名（迁移用）

AppQueryDTO 字段说明（继承 PageQueryDTO：page 页码默认1，size 每页条数默认20）

字段名	类型	中文说明
appName	String	应用名称（模糊查询）
contextPath	String	上下文路径

virtualHost	String	虚拟主机
status	Integer	应用状态

5 AuditLogController

基础路径: /api/audit/operation-logs

方法	路径	函数	参数	返回类型	说明
GET	/api/audit/operation-logs	list	AuditLogQueryDTO query	Result<PageResult<AuditLogEntity>>	分页查询操作审计日志（按操作人/事件类型/操作对象/结果/时间范围过滤）
GET	/api/audit/operation-logs/{id}	getById	@PathVariable Long id	Result<AuditLogEntity>	按 ID 获取单条审计日志（description 中的 i18n key 按当前 Locale 翻译），不存在返回 404
GET	/api/audit/operation-logs/export	export	AuditLogQueryDTO query, HttpServletResponse response	void (application/zip)	导出审计日志：按查询条件导出全部记录，打包为 operation_audit.zip（内含 UTF-8 BOM 的 operation_audit.csv）

AuditLogQueryDTO 字段说明（继承 PageQueryDTO：page 页码默认1，size 每页条数默认20）

字段名	类型	中文说明
username	String	操作人用户名
operation	String	事件类型 (CREATE/EDIT/DELETE/START/STOP/AUTH/IMPORT/EXPORT/LICENSE/BACKUP)
target	String	操作对象
result	String	操作结果 (SUCCESS/FAIL)
startTime	String	开始时间
endTime	String	结束时间

6 AuthController

基础路径: /api/auth

方法	路径	函数	参数	返回类型	说明
POST	/api/auth/login	login	@RequestBody LoginRequest request, HttpServletRequest servletRequest	Result<LoginResponse>	用户登录: AES 解密用户名密码, 依次校验验证码、账号状态 (锁定/禁用/过期)、访问时间与 IP 白名单、OTP、邮箱验证码, 成功后签发 JWT 并记录登录日志
POST	/api/auth/cli-login	cliLogin	@RequestBody LoginRequest request, HttpServletRequest servletRequest	Result<LoginResponse>	CLI 登录: 同登录流程但不校验图形验证码
POST	/api/auth/logout	logout	HttpServletRequest request	Result<Void>	注销: 将当前 token 加入黑名单并记录注销日志
GET	/api/auth/captcha-status	getCaptchaStatus		Result<Map<String, Object>>	获取图形验证码开关状态 (enabled)
GET	/api/auth/otp-status	getOtpStatus		Result<Map<String, Object>>	获取 OTP 双因子认证全局开关状态
GET	/api/auth/email-code/status	getEmailCodeStatus		Result<Map<String, Object>>	获取邮箱验证码登录校验开关状态
POST	/api/auth/email-code/send	sendEmailCode	@RequestBody Map<String, String> body (username)	Result<Map<String, Object>>	发送登录邮箱验证码, 含 60 秒重发限制

POST	/api/auth/otp/setup	setupOtp	@RequestBody Map<String, String> body	Result<Map<String, Object>>	为当前登录用户生成 OTP 密钥及二维码 (secret、otp AuthUri、qrCode DataUri)
POST	/api/auth/otp/verify-setup	verifyOtpSetup	@RequestBody Map<String, String> body (secret、code)	Result<Void>	校验动态码并完成 OTP 绑定
POST	/api/auth/otp/disable	disableOtp	@RequestBody Map<String, String> body (code)	Result<Void>	关闭当前用户的 OTP (需输入当前动态码确认)
PUT	/api/auth/profile	updateProfile	@RequestBody Map<String, String> body (phone/email/description)	Result<Void>	修改当前登录用户自己的个人信息
GET	/api/auth/me	me		Result<Map<String, Object>>	获取当前登录用户信息 (角色、密级、联系方式、会话超时、资源编码列表等)

LoginRequest 字段说明

字段名	类型	中文说明
username	String	用户名 (AES 加密传输, 服务端解密)
password	String	密码 (AES 加密传输, 服务端解密)
captchaKey	String	图形验证码 key
captchaCode	String	图形验证码答案
otpCode	String	OTP 动态码
otpSecret	String	OTP 密钥 (首次绑定时由后端下发、前端回传)
emailCode	String	邮箱验证码

LoginResponse 字段说明

字段名	类型	中文说明
token	String	JWT 访问令牌
tokenType	String	令牌类型, 固定 Bearer
username	String	登录用户名
roles	List	用户角色编码列表
firstLogin	boolean	是否首次登录 (需强制改密)
passwordExpired	boolean	密码已过期, 需要修改密码
requireOtp	boolean	全局是否要求 OTP 校验
otpEnabled	boolean	当前用户是否已绑定 OTP
requireEmailCode	boolean	本次登录是否需要邮箱验证码
emailCodeEnabled	boolean	邮箱验证码功能是否启用
otpQrDataUri	String	OTP 首次绑定的二维码 base64 dataUri
otpSecret	String	OTP 首次绑定的密钥 (二次提交时回传)
passwordExpireWarning	boolean	密码即将过期警告 (剩余 7 天内)
resources	List	用户拥有的资源编码列表 (前端菜单权限用)

7 BaselineCheckController

基础路径: `/api/baseline`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/baseline/rules</code>	<code>listRules</code>		<code>Result<String></code>	列出全部基线检查规则 (XML/JSON 字符串)
GET	<code>/api/baseline/rules/{id}</code>	<code>getRule</code>	<code>@PathVariable String id</code>	<code>Result<String></code>	按规则 ID 获取单条规则内容 (id 仅允许字母数字下划线连字符, 否则 400)
POST	<code>/api/baseline/rules</code>	<code>saveRule</code>	<code>@RequestBody String ruleXml</code>	<code>Result<String></code>	新增基线检查规则 (规则 XML 原文入参)
PUT	<code>/api/baseline/rules/{id}</code>	<code>updateRule</code>	<code>@PathVariable String id, @RequestBody String ruleXml</code>	<code>Result<String></code>	更新指定 ID 的规则 (整体保存规则 XML)
DELETE	<code>/api/baseline/rules/{id}</code>	<code>deleteRule</code>	<code>@PathVariable String id</code>	<code>Result<String></code>	删除指定 ID 的基线规则
PUT	<code>/api/baseline/rules/{id}/enabled</code>	<code>setRuleEnabled</code>	<code>@PathVariable String id, @RequestBody EnabledRequest request</code>	<code>Result<String></code>	启用/停用指定基线规则
GET	<code>/api/baseline/script</code>	<code>readScript</code>	<code>@RequestParam String fileName</code>	<code>Result<String></code>	读取检查脚本文件内容
POST	<code>/api/baseline/script</code>	<code>saveScript</code>	<code>@RequestParam String fileName, @RequestBody String content</code>	<code>Result<String></code>	保存检查脚本文件
POST	<code>/api/baseline/checks</code>	<code>startCheck</code>	<code>@RequestBody StartCheckRequest request</code>	<code>Result<String></code>	手动触发一次基线检查, 返回任务 ID

					(ruleIds 为空表示 全量规则)
GET	/api/baseline/che cks/{taskId}	pollTas k	@PathVariable str ing taskId	Result<String>	轮询检查任务执行 进度/结果
GET	/api/baseline/rep orts	listRep orts	ReportQuery query (绑定查询参数: name/overall/start Time/endTime/pa ge/size)	Result<PageResult <BaselineReport> >	分页查询基线检查 报告历史
GET	/api/baseline/rep orts/{id}	getRe port	@PathVariable Lon g id	Result<BaselineRe port>	获取单份基线报告 详情
DELET E	/api/baseline/rep orts/{id}	delete Report	@PathVariable Lon g id	Result<Void>	删除单份基线报告
DELET E	/api/baseline/rep orts	delete Report s	@RequestBody Li st<Long> ids	Result<Void>	批量删除基线报告
GET	/api/baseline/rep orts/{id}/downloa d	downl oadRe port	@PathVariable Lon g id, @RequestPar am(required=false) String format	ResponseEntity<B yteArrayResource >	下载单份报告文件 (format 可选, 如 pdf/html 等, 附件 流返回)
POST	/api/baseline/rep orts/export	export Report s	@RequestBody Li st<Long> ids	ResponseEntity<B yteArrayResource >	将多份报告打包为 ZIP 导出下载
GET	/api/baseline/set tings	getSet tings		Result<String>	读取基线检查全部 配置项
PUT	/api/baseline/set tings	saveSe tting	@RequestBody co nfigEntry entry	Result<String>	修改单个基线配置 项; cron/总开关 变更发布事件实现 定时任务热重载
GET	/api/baseline/rep ort-config	getRe portC onfig		Result<String>	读取报告相关配置 (同 getSettings)
PUT	/api/baseline/rep ort-config	saveRe portC onfig	@RequestBody co nfigEntry entry	Result<String>	修改报告相关配置 项

StartCheckRequest 字段说明

字段名	类型	中文说明
ruleIds	List	要执行的规则 ID 列表，为空表示执行全部规则

EnabledRequest 字段说明

字段名	类型	中文说明
enabled	boolean	规则启用状态 (true 启用 / false 停用)

ReportQuery 字段说明

字段名	类型	中文说明
name	String	报告名称模糊查询
overall	String	报告总体结论 (通过/不通过等) 筛选
startTime	LocalDateTime	检查时间起 (ISO 日期时间格式)
endTime	LocalDateTime	检查时间止
page	int	页码，默认 1
size	int	每页条数，默认 10

ConfigEntry 字段说明

字段名	类型	中文说明
key	String	配置项键名 (如 baseline.schedule.cron、baseline.enabled)
value	String	配置项值

BaselineReport 字段说明 (返回实体，继承 BaseEntity 公共字段 id/createdTime/updatedTime/deleted)

字段名	类型	中文说明
name	String	报告名称
instance	String	检查目标实例标识
checkTime	LocalDateTime	检查执行时间
overall	String	总体结论 (如 PASS/FAIL)

passCount	Integer	通过项数量
minorCount	Integer	轻微问题数量
generalCount	Integer	一般问题数量
severeCount	Integer	严重问题数量
ruleCount	Integer	参与检查的规则总数
reportFile	String	报告文件存储路径
format	String	报告文件格式
reportJson	String	报告明细 JSON 内容
triggerType	String	触发类型 (MANUAL 手动 / SCHEDULED 定时)
size	Long	报告文件大小 (字节), 查询时按实际文件填充, 非数据库字段

8 CaptchaController

基础路径: /api/auth

方法	路径	函数	参数	返回类型	说明
GET	/api/auth/captcha	getCa ptcha		Result<Map<Strin g, String>>	生成图形验证码， 返回 key 与 base6 4 图片
GET	/api/auth/captcha -test	getCa ptchaT est	@RequestParam s tring key	Result<Map<Strin g, String>>	测试辅助接口：按 key 获取验证码明 文，仅自动化测试 环境开启（aas.sec urity.captcha.test- enabled），禁止 在生产环境开启， 否则验证码机制失 效

9 CertificateStoreController

基础路径: `/api/config/web/certs`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/config/web/certs</code>	list	@RequestParam int page, int size, String name, String storeType	Result<PageResult<CertificateStore>>	分页查询证书库列表（密码字段以 15 个星号掩码返回），storeType 仅支持 keystore/truststore
GET	<code>/api/config/web/certs/{id}</code>	get	@PathVariable Long id	Result<CertificateStore>	按 ID 获取证书库详情（密码掩码返回），不存在返回 404
POST	<code>/api/config/web/certs</code>	create	@RequestBody CertificateStoreDTO dto	Result<Long>	创建证书库，返回新记录 ID
PUT	<code>/api/config/web/certs/{idOrName}</code>	update	@PathVariable String idOrName, @RequestBody CertificateStoreDTO dto	Result<Void>	按 ID 或名称更新证书库
DELETE	<code>/api/config/web/certs/{idOrName}</code>	delete	@PathVariable String idOrName	Result<Void>	按 ID 或名称删除证书库
GET	<code>/api/config/web/certs/{id}/entries</code>	listEntries	@PathVariable Long id, @RequestParam String password	Result<List<CertEntryDTO>>	列出证书库中的证书条目（可选密码解锁）
DELETE	<code>/api/config/web/certs/{idOrName}/entries/{alias}</code>	deleteEntry	@PathVariable String idOrName, String alias, @RequestParam String password	Result<Void>	删除证书库中指定别名的证书条目
POST	<code>/api/config/web/certs/{idOrName}/import</code>	importEntries	@PathVariable String idOrName, @	Result<Void>	从上传的证书库文件导入条目（可选

	import-entries		RequestParam MultiPartFile file, String sourcePassword, String aliases, String targetPassword, String targetAlias, String entryPassword		别名筛选与目标别名/密码)
POST	/api/config/web/certs/{idOrName}/import-pem	import Pem	@PathVariable String idOrName, @RequestParam String content, String alias, String targetPassword, String entryPassword, boolean isPrivateKey	Result<Void>	粘贴 PEM 内容导入证书或私钥 (isPrivateKey 标识私钥)
POST	/api/config/web/certs/{idOrName}/import	import Store	@PathVariable String idOrName, @RequestParam MultiPartFile file	Result<Void>	整体导入证书库文件 (覆盖替换)
GET	/api/config/web/certs/{id}/download	download	@PathVariable Long id	ResponseEntity<Resource>	下载证书库文件, 文件不存在返回 404

CertificateStoreDTO 字段说明

字段名	类型	中文说明
id	Long	证书库 ID
name	String	证书库名称 (创建必填, 最长 64 字符, 仅字母/数字/下划线/短横线/点号)
storeType	String	证书库用途 (创建必填, keystore 或 truststore)
type	String	证书库类型 (创建必填, JKS 或 PKCS12)
password	String	证书库密码 (支持 ENCRYPT* 密文, 最长 128/256 字符)
description	String	描述 (最长 255 字符)
enabled	Integer	是否启用

10 ClassAnalysisController

基础路径: /api/class-analysis

方法	路径	函数	参数	返回类型	说明
POST	/api/class-analysis/upload	upload	@RequestParam ("file") MultipartFile file	Result<Map<String, Object>>	上传待分析的文件 (jar/war 等) , 返回 filePath/fileName/fileName/fileName
POST	/api/class-analysis/start	start	@RequestBody classAnalysisStartRequest req	Result<Map<String, Object>>	启动类分析任务, 返回 recordId (分析记录 ID)
GET	/api/class-analysis/records/{id}	getRecord	@PathVariable Long id	Result<ClassAnalysisRecordDTO>	查询单条类分析记录详情 (含问题列表)
GET	/api/class-analysis/records	listRecords	ClassAnalysisRecordQuery q (绑定查询参数: startTime/endTime/target/appOrFileName/page/pageSize)	Result<PageResult<ClassAnalysisRecordDTO>>	分页查询类分析历史记录
DELETE	/api/class-analysis/records/{id}	deleteRecord	@PathVariable Long id	Result<Void>	删除指定类分析记录
GET	/api/class-analysis/records/{id}/export	exportReport	@PathVariable Long id	ResponseEntity<byte[]>	将分析记录渲染为 PDF 报告下载
GET	/api/class-analysis/deployed-apps	listDeployedApps	@RequestParam(required=false) String keyword, @RequestParam(defaultValue="1") int page, @RequestParam(defaultValue="20") int pageSize	Result<PageResult<AppOptionDTO>>	分页查询已部署应用列表 (供选择分析目标)

ClassAnalysisStartRequest 字段说明

字段名	类型	中文说明
name	String	分析任务名称
target	String	分析目标类型（如上传文件/已部署应用）
appld	Long	目标应用 ID（分析已部署应用时填写）
uploadFilePath	String	已上传文件的存储路径（分析上传文件时填写）
scope	String	分析范围
checkConflict	Boolean	是否执行类冲突检查
checkRedundant	Boolean	是否执行冗余类检查

11 ClassLoaderController

基础路径: /api/classloader

方法	路径	函数	参数	返回类型	说明
GET	/api/classloader/tree	getClassLoaderTree		Result<Map<String, Object>>	获取类加载器树 (data.classLoaders 为 ClassLoader Node 树形列表)
GET	/api/classloader/detail/{id}	getClassLoaderDetail	@PathVariable String id	Result<ClassLoaderDetail>	按类加载器 ID 获取详情 (类名、实例、hash、类路径等)
GET	/api/classloader/search	searchClass	@RequestParam String className	Result<List<ClassSearchResult>>	按类名搜索类在各类加载器中的加载情况 (名称/状态/所属应用/加载器/位置)

ClassLoaderNode 字段说明

字段名	类型	中文说明
id	String	节点唯一标识
name	String	类加载器显示名称
className	String	类加载器实现类全名
instance	String	实例标识
hash	String	实例 hash
app	String	所属应用
isAppLoader	boolean	是否为应用类加载器
paths	List	类搜索路径列表
children	List	子类加载器节点

12 CracController

基础路径: /api/config/jvm/crac

方法	路径	函数	参数	返回类型	说明
GET	/api/config/jvm/crac/env-check	envCheck		Result<CracEnvCheckReport>	同步执行 7 项 CRaC 环境兼容性检测（开启 CRaC 开关时触发）
GET	/api/config/jvm/crac/status	status		Result<CracStatusDTO>	获取「制作镜像」按钮置灰联动状态：已开启保存 + 无待重启 + 当前 JVM 支持 CRaC + 非恢复态进程 + 运行时目录与固定值一致时才允许制作；reason 取值 notEnabled/pendingRestart/notCracJdk/restored/dirMismatch；同时返回 crac.log 路径
POST	/api/config/jvm/crac/images	makeImage	@RequestBody cracMakeImageRequest request	Result<Void>	制作 CRaC 镜像：校验通过后派生 detached helper 异步完成镜像生成与服务器拉起（结果见 logs/crac.log）；检查点目录已有镜像数据时返回业务码 409，前端确认后以 force=true 重试

CracMakeImageRequest 字段说明

字段名	类型	中文说明
name	String	镜像名称（必填，限字母/数字/下划线/中划线/点，最长 64）

desc	String	镜像说明（可选，最长 200）
force	boolean	检查点目录已有镜像数据时是否强制覆盖

13 DashboardController

基础路径: /api/dashboard

方法	路径	函数	参数	返回类型	说明
GET	/api/dashboard/overview	getOverview		Result<DashboardOverviewDTO>	获取仪表盘首页概览数据（系统统计、应用状态、快捷信息等）

14 DatasourceController

基础路径: `/api/resources/datasources`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/resources/datasources</code>	list	DatasourceQueryDTO query	Result<PageResult<Datasource>>	分页查询数据源列表
GET	<code>/api/resources/datasources/{id}</code>	get	@PathVariable Long id	Result<Datasource>	按 ID 获取数据源详情
POST	<code>/api/resources/datasources (JSON)</code>	create ByJson	@Valid @RequestBody Datasource ds	Result<Datasource>	以 JSON 方式创建数据源（使用已上传的驱动路径）
POST	<code>/api/resources/datasources (multipart)</code>	create	@Valid @RequestParam DataSource, MultipartFile driverFile, boolean Overwrite	Result<Datasource>	以 multipart 方式创建数据源，可同时上传驱动包（overwrite 控制覆盖同名驱动）
PUT	<code>/api/resources/datasources/{idOrName}</code>	update	@PathVariable String idOrName, @Valid @RequestBody Datasource ds	Result<Void>	按 ID 或名称更新数据源
DELETE	<code>/api/resources/datasources/{idOrName}</code>	delete	@PathVariable String idOrName	Result<Void>	按 ID 或名称删除数据源
POST	<code>/api/resources/datasources/{idOrName}/status</code>	toggle Status	@PathVariable String idOrName, @RequestParam Integer status	Result<Void>	启用/停用数据源（status 仅允许 0 或 1）
POST	<code>/api/resources/datasources/ping</code>	pingBy Params	@RequestBody Datasource ds	Result<Map<String, Object>>	按传入参数测试数据源连通性（url 必填）
POST	<code>/api/resources/datasources/{idOrName}/ping</code>	ping	@PathVariable String idOrName	Result<Map<String, Object>>	测试已存在数据源的连通性

POST	/api/resources/datasources/{idOrName}/driver	uploadDriver	@PathVariable String idOrName, @RequestParam MultipartFile driverFile, boolean overwrite	Result<Void>	为数据源上传 JDBC 驱动包
GET	/api/resources/datasources/driver-exists	checkDriverExists	@RequestParam String filename	Result<Map<String, Object>>	检查驱动目录是否存在同名驱动包（上传前预检）
POST	/api/resources/datasources/check-driver-path	checkDriverPath	@RequestBody Map<String, String> body (driverPath)	Result<Map<String, Object>>	检查指定的驱动路径是否存在

Datasource 字段说明（创建/更新请求体，仅列主要业务字段）

字段名	类型	中文说明
name	String	数据源名称（最长 128 字符）
jndiName	String	JNDI 名称
dbType	String	数据库类型
driverClass	String	JDBC 驱动类名（最长 128 字符）
url	String	连接 URL（最长 500 字符）
username	String	连接用户名
password	String	连接密码
maxConnections	Integer	最大连接数（0-10000）
initialConnections	Integer	初始连接数（0-10000）
maxWaitMs	Integer	获取连接最大等待时间（毫秒，-1 表示无限等待）
validationQuery	String	连接验证 SQL
classLevel	Integer	密级标识
status	Integer	状态
driverJarPath	String	驱动 JAR 路径

driverPath	String	驱动路径
slowSqlMs	Integer	慢 SQL 阈值（毫秒）
leakDetectionThresholdMs	Integer	连接泄漏检测阈值（毫秒）
suspectTimeoutMs	Integer	疑似泄露判定超时（毫秒）
testOnBorrow	Integer	获取连接时是否校验（0/1）
testOnReturn	Integer	归还连接时是否校验（0/1）
testWhileIdle	Integer	空闲时是否校验（0/1）
testOnConnect	Integer	建立连接时是否校验（0/1）
resourceType	String	资源类型
host	String	主机名
port	Integer	端口号（1-65535）
databaseName	String	数据库名
customProperties	String	自定义属性
idleTimeoutMs	Integer	空闲超时时间（毫秒）
idleCheckPeriodMs	Integer	空闲检测周期（毫秒）
maxCheckConnections	Integer	最大检测连接数
minIdle	Integer	最小空闲连接数
maxIdle	Integer	最大空闲连接数
validationIntervalMs	Integer	连接校验间隔（毫秒）
validationTimeoutSeconds	Integer	校验超时时间（秒，0-3600）
maxLifetimeMs	Integer	连接最大存活时间（毫秒）
leakRecover	Integer	是否回收泄露连接（0/1）
connectionAutoReclaim	Integer	连接自动回收（0/1）
removeAbandonedOnMaintenance	Integer	维护时移除废弃连接（0/1）
logAbandoned	Integer	记录废弃连接日志（0/1）
abandonWhenPercentageFull	Integer	废弃百分比阈值（0-100）

autoCommit	Integer	自动提交 (0/1)
sqlTraceEnabled	Integer	SQL 追踪开关 (0/1)
failedSqlAlertEnabled	Integer	失败 SQL 告警开关 (0/1)
haProbeEnabled	Integer	HA 探测开关 (0/1)

15 HttpChannelController

基础路径: /api/config/web/channels

方法	路径	函数	参数	返回类型	说明
GET	/api/config/web/channels	list	@RequestParam int page, int size, String name, String protocol, Integer enabled	Result<PageResult<HttpChannel>>	分页查询 HTTP 通道列表
GET	/api/config/web/channels/{id}	get	@PathVariable Long id	Result<HttpChannel>	按 ID 获取 HTTP 通道详情, 不存在返回 404
POST	/api/config/web/channels	create	@RequestBody HttpChannel channel	Result<Void>	新增 HTTP 通道 (需重启生效)
PUT	/api/config/web/channels/{idOrName}	update	@PathVariable String idOrName, @RequestBody HttpChannel channel	Result<Void>	按 ID 或名称更新 HTTP 通道 (需重启生效)
DELETE	/api/config/web/channels/{idOrName}	delete	@PathVariable String idOrName	Result<Void>	按 ID 或名称删除 HTTP 通道 (需重启生效)
GET	/api/config/web/channels/cipher-suites	getSupportedCipherSuites		Result<List<String>>	获取 JVM 支持的强加密套件列表 (已过滤弱 CBC/RSA 套件)

HttpChannel 字段说明 (@RequestBody, 字段较多, 按分组概述)

字段名	类型	中文说明
name	String	通道名称
protocol	String	协议 (HTTP/HTTPS)
host	String	监听主机

port	Integer	监听端口
executor	String	关联线程池 executor 名称
enabled	Integer	是否启用
sslEnabled	Integer	是否启用 SSL
redirectPort	Integer	重定向端口
acceptCount	Integer	等待队列长度
connectionTimeout	Integer	连接超时（毫秒）
proxyName / proxyPort	String / Integer	反向代理主机名/端口
receiveBufferSize / sendBufferSize	Integer	接收/发送缓冲区大小
processorCache	Integer	处理器缓存大小
maxConnections	Integer	最大连接数
tcpNoDelay	Integer	是否启用 TCP_NODELAY
maxKeepAliveRequests / keepAliveTimeout	Integer	长连接最大请求数/超时
useIPVHosts	Integer	是否按 IP 匹配虚拟主机
longThreadEnabled 等 longThread*	-	长线程检测开关及阈值/周期/日志目录
sdosEnabled 等 sdos*	-	慢拒绝服务（SDoS）防护开关、模式、阈值、黑名单 TTL 等
generalProperties	String	一般自定义属性
gracefulStopAwaitMillis	Integer	优雅停止等待时间（毫秒）
uriEncoding / responseCharacterEncoding	String	URI/响应字符编码
enableLookups	Integer	是否允许 DNS 反查
compression 等 compression*	-	响应压缩开关、可压缩 MIME 类型、最小压缩大小
httpTrace	Integer	HTTP 访问跟踪开关
maxPostSize / maxParameterCount / maxHeaderCount / maxHttpRequestSize	Integer	POST 大小/参数个数/请求头个数与大小上限

poweredBy	Integer	是否显示 Server 响应头
disableUploadTimeout / connectionUploadTimeout	Integer	上传超时配置
useBodyEncodingForURI / asyncTimeout	-	URI 编码来源/异步请求超时
responseHeaders	String	自定义响应头
gmEnabled	Integer	国密 SSL 开关
clientAuth	String	SSL 客户端认证模式
keyPassword / certificateFormat / tlsProtocols / sslProtocol	String	SSL 密钥密码/证书格式/TLS 协议版本
certificateAlias	String	证书别名
keystorePath / keystorePassword	String	密钥库路径/密码
truststorePath / truststorePassword / truststoreType	String	信任库路径/密码/类型
sslCipherSuites	String	SSL 加密套件列表
certificateType	String	证书类型 (JSON 属性名 type)
certificateCustom / keystoreName / keystoreEntry / truststoreName / truststoreEntry	-	证书库引用方式及条目
http2Enabled 等 http2*	-	HTTP/2 开关及压缩、并发流、头/帧大小、读写超时等配置
http2Properties	String	HTTP/2 自定义属性

16 InspectionController

基础路径: /api/inspection/reports

方法	路径	函数	参数	返回类型	说明
GET	/api/inspection/reports	list	@RequestParam(required=false) String name, String status, LocalDateTime startTime, LocalDateTime endTime, @RequestParam(defaultValue="1") int page, @RequestParam(defaultValue="10") int size	Result<PageResult<InspectionReportDTO>>	分页查询巡检报告列表（时间格式 yy-MM-dd HH:mm:ss）
GET	/api/inspection/reports/has-running	hasRunning		Result<Boolean>	查询当前是否存在执行中的巡检任务
GET	/api/inspection/reports/{id}	get	@PathVariable Long id	Result<InspectionReportDTO>	获取单份巡检报告详情（含汇总、实例列表、基线项、监控数据），不存在时报 404
GET	/api/inspection/reports/{id}/status	status	@PathVariable Long id	Result<String>	查询巡检报告执行状态（RUNNING/SUCCESS/FAILED等）
GET	/api/inspection/reports/{id}/summary	summary	@PathVariable Long id	Result<InspectionReportDTO.InspectionSummaryDTO>	获取报告摘要（CPU/内存/堆/磁盘超80% 计数）
POST	/api/inspection/reports	trigger	@RequestBody @Valid InspectionTriggerDTO dto	Result<InspectionReportDTO>	触发一次新巡检，返回报告（dto.timeoutSeconds 为采集超时秒数）

POST	/api/inspection/reports/{id}/rerun	rerun	@PathVariable Long id	Result<InspectionReportDTO>	基于历史报告重新执行巡检
GET	/api/inspection/reports/{id}/download	download	@PathVariable Long id	ResponseEntity<Resource>	下载巡检报告 ZIP 包（含全部快照文件）
DELETE	/api/inspection/reports/{id}	delete	@PathVariable Long id	Result<Void>	删除指定巡检报告
GET	/api/inspection/reports/{id}/snapshots/{type}	snapshotText	@PathVariable Long id, @PathVariable String type	Result<String>	在线查看报告内指定类型系统快照（jstack/lsof/netstat/top）文本内容
GET	/api/inspection/reports/{id}/snapshots/{type}/download	downloadSnapshot	@PathVariable Long id, @PathVariable String type	ResponseEntity<Resource>	下载指定类型快照 XML 文件

InspectionTriggerDTO 字段说明

字段名	类型	中文说明
timeoutSeconds	Integer	巡检采集超时时间（秒），必填，1~172800

InspectionReportDTO 字段说明（响应 DTO）

字段名	类型	中文说明
id	Long	报告 ID
name	String	报告名称
timeoutSeconds	Integer	采集超时秒数
status	String	执行状态
startedAt	LocalDateTime	开始时间
finishedAt	LocalDateTime	结束时间
createdTime	LocalDateTime	创建时间
summary	InspectionSummaryDTO	报告摘要（仅详情接口返回）
instances	List	巡检实例列表（仅详情接口返回）

baselineItems	List	安全基线检查项列表
monitorData	Map<String, Object>	监控数据（应用程序/HTTP通道/数据源/线程池/OS/JVM）
snapshotStatuses	Map<String, SnapshotStatus>	系统快照采集状态（key: jstack/lsof/netstat/top），含 status（OK/NOT_INSTALLED/FALLBACK/SKIPPED/FAILED/TIMEOUT）与 message

17 InvokeChainTraceController

基础路径: `/api/monitor/invoke-chain`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/monitor/invoke-chain/records</code>	listRecords	@RequestParam(required=false) LocalDateTime startTime, LocalDateTime endTime, String status, Long minSize, @RequestParam(defaultValue="1") int page, @RequestParam(defaultValue="5") int size	Result<PageResult<InvokeChainTrace>>	分页查询调用链追踪历史记录（时间格式 yyyy-MM-dd HH:mm:ss）
DELETE	<code>/api/monitor/invoke-chain/records/{id}</code>	deleteRecord	@PathVariable Long id	Result<Void>	删除指定追踪记录
GET	<code>/api/monitor/invoke-chain/settings</code>	getSettings		Result<Map<String, Object>>	获取调用链分析配置（含 apmHome 等）
PUT	<code>/api/monitor/invoke-chain/settings/apm-home</code>	saveApmHome	@RequestBody Map<String, String> body（取 apmHome 键）	Result<Map<String, Object>>	保存 APM 探针安装目录（apmHome），需重启生效
POST	<code>/api/monitor/invoke-chain/trace/start</code>	startTrace	@RequestBody @Valid InvokeChainTraceStartRequest request	Result<Map<String, Object>>	启动调用链追踪（按耗时阈值/包过滤等参数）
POST	<code>/api/monitor/invoke-chain/trace/stop</code>	stopTrace		Result<Map<String, Object>>	停止当前调用链追踪
GET	<code>/api/monitor/invoke-chain/trace/status</code>	status		Result<Map<String, Object>>	查询追踪运行状态

	atus				
GET	/api/monitor/invoke-chain/records/{id}/xml	downloadXml	@PathVariable Long id	ResponseEntity<Resource>	下载追踪结果 XML 文件
GET	/api/monitor/invoke-chain/records/{id}/flame-graph	flameGraph	@PathVariable Long id	ResponseEntity<Resource>	生成并下载该记录的火焰图 HTML 文件

InvokeChainTraceStartRequest 字段说明

字段名	类型	中文说明
cost	Long	请求最小耗时（毫秒），必填，正整数
methodMinCost	Long	方法最小耗时（毫秒），必填，正整数且必须小于 cost
excPkgs	String	排除的包名，多个用逗号分隔（不能含空格）
incPkgs	String	包含的包名，多个用逗号分隔（不能含空格）
appContext	String	目标应用上下文路径（如 /myapp），可选；为空不按应用过滤
deadline	LocalDateTime	追踪截止时间（yyyy-MM-dd HH:mm:ss），可选；为空不自动停止

InvokeChainTrace 字段说明（返回实体，继承 BaseEntity 公共字段 id/createdTime/updatedTime/deleted）

字段名	类型	中文说明
startTime	LocalDateTime	追踪开始时间
endTime	LocalDateTime	追踪结束时间
status	String	状态：SUCCESS / FAILED
params	String	追踪参数快照 JSON (cost/methodMinCost/incPkgs/excPkgs/appContext/deadline)
fileName	String	追踪结果 XML 文件名（用于下载）
filePath	String	追踪结果 XML 文件绝对路径（探针所在机器路径）
dataSize	Long	追踪结果 XML 数据大小（字节）
errorMsg	String	失败/异常信息

18 JmxConfigController

基础路径: /api/config/jmx

方法	路径	函数	参数	返回类型	说明
GET	/api/config/jmx	getConfig		Result<JmxConfig>	获取 JMX 配置 (keyStorePassword/keyPassword 以星号掩码返回)
GET	/api/config/jmx/cipher-suites	getSupportedCipherSuites		Result<List<String>>	获取 JVM 支持的强加密套件列表 (已过滤弱套件)
PUT	/api/config/jmx	saveConfig	@RequestBody JmxConfig config	Result<Void>	保存 JMX 配置 (字符串字段去空格, 需重启生效)

JmxConfig 字段说明

字段名	类型	中文说明
enabled	Integer	是否启用 JMX
host	String	JMX 监听主机
port	Integer	JMX 监听端口
authenticate	Integer	是否启用认证
passwordFile	String	密码文件路径
accessFile	String	访问控制文件路径
sslEnabled	Integer	是否启用 SSL
clientAuth	Integer	是否要求客户端证书认证
keyStoreFile	String	密钥库文件路径
keyPassword	String	密钥密码
keyStorePassword	String	密钥库密码

keyStoreType	String	密钥库类型
trustStoreFile	String	信任库文件路径
trustStorePassword	String	信任库密码
trustStoreType	String	信任库类型
enabledCipherSuites	String	启用的加密套件列表
certificateCustom	Integer	是否自定义证书库引用
keyStoreName	String	密钥库名称（证书库引用）
trustStoreName	String	信任库名称（证书库引用）
keyStoreEntry	String	密钥库证书条目别名

19 JndiResourceController

基础路径: /api/resources/jndi

方法	路径	函数	参数	返回类型	说明
GET	/api/resources/jndi	list	PageQueryDTO query	Result<PageResult<JndiResource>>	分页查询 JNDI 资源列表
GET	/api/resources/jndi/{id}	get	@PathVariable Long id	Result<JndiResource>	按 ID 获取 JNDI 资源详情, 不存在返回 404
POST	/api/resources/jndi	create	@RequestBody JndiResourceDTO dto	Result<Void>	创建 JNDI 资源
PUT	/api/resources/jndi/{id}	update	@PathVariable Long id, @RequestBody JndiResourceDTO dto	Result<Void>	更新 JNDI 资源
DELETE	/api/resources/jndi/{id}	delete	@PathVariable Long id	Result<Void>	删除 JNDI 资源
POST	/api/resources/jndi/{id}/status	toggle Status	@PathVariable Long id, @RequestParam Integer status	Result<Void>	启用/停用 JNDI 资源 (status 只能为 0 或 1)

JndiResourceDTO 字段说明

字段名	类型	中文说明
jndiName	String	JNDI 名称 (创建必填, 最长 256 字符)
resourceType	String	资源类型 (创建必填)
factoryClass	String	工厂类全限定类名 (创建必填, 最长 512 字符)
properties	String	自定义属性 (JSON 对象字符串, 最长 65536 字符)
description	String	资源描述 (最长 256 字符)

20 JvmConfigController

基础路径: /api/config/jvm

方法	路径	函数	参数	返回类型	说明
GET	/api/config/jvm	getConfig		Result<JvmConfig>	获取 JVM 配置
PUT	/api/config/jvm	saveConfig	@RequestBody JvmConfig config	Result<?>	保存 JVM 配置： 校验堆大小（Xms ≤Xmx、正整数）、元空间（-1 或正数、初始≤最大）、JVM 选项格式（-D/-X/-XX:/-javaagent: 开头且不重复）、本地库路径存在性、调试选项格式；开启 C RaC 时校验 fdPolicy 必填且为存在的文件。400 时 data 捎带 cracFdPolicy 等字段名供前端展示。需重启生效

JvmConfig 字段说明

字段名	类型	中文说明
jdkPath	String	JDK 路径
javaVersion	String	Java 版本
xms	String	堆最小值（MB，正整数）
xmx	String	堆最大值（MB，不小于 xms）
xmn	String	新生代大小
permSize / maxPermSize	String	永久代初始/最大大小（兼容旧配置）
metaspaceSize	String	元空间初始值（-1 或正整数）

maxMetaspaceSize	String	元空间最大值（-1 或正整数，不小于初始值）
gcType	String	垃圾收集器类型
gcLogPath	String	GC 日志路径
debugEnabled	Integer	是否启用远程调试
debugOptions	String	调试选项（仅允许 -D/-X/-XX:/-agentlib: 开头的 token）
nativeLibraryPath	String	本地库路径（多个路径以路径分隔符分隔，均需存在）
jvmOptions	String	附加 JVM 选项（每行一条，-D/-X/-XX:/-javaagent: 开头且不重复）
oomAction	String	OOM 处理策略
cracEnabled	Integer	是否启用 CRaC 特性
cracCheckpointDir	String	CRaC 检查点目录（服务端固定下发，请求值忽略）
cracFdPolicy	String	CRaC 文件描述符策略文件路径（开启 CRaC 时必填且必须存在）
cracTraceStartupTime	Integer	CRaC 启动耗时跟踪
cracPrintResources	Integer	CRaC 打印资源信息
cracCollectFdStacktraces	Integer	CRaC 收集 fd 堆栈
cracFdPolicyDefault	String	fdPolicy 默认路径（运行时解析，不入库，供前端恢复默认值）

21 LicenseController

基础路径: /api/license

方法	路径	函数	参数	返回类型	说明
GET	/api/license/info	info		Result<Map<String, Object>>	获取当前生效授权信息（含机器标识id、是否有授权、特性码、授权对象、起止日期、授权类型、是否永久、剩余天数、产品名/版本、是否云环境）
GET	/api/license/days-remaining	daysRemaining		Result<Map<String, Object>>	获取授权剩余天数
POST	/api/license/update	update	@RequestParam multipartFile file, String content, String fileName	Result<License>	更新授权（上传 .lic.xml 文件或提交授权文本内容，带文件大小/文件名/内容长度校验）

22 LogController

基础路径: /api/logs

方法	路径	函数	参数	返回类型	说明
GET	/api/logs/files	getLogFiles		Result<List<Map<String, Object>>>	获取日志文件列表（扫描系统日志目录与访问日志目录，含文件名、路径、大小、最后修改时间、类型）
GET	/api/logs/content	getLogContent	@RequestParam String fileName, long offset, String direction, int pageSize, String keyword, String logLevel, String startTime, String endTime	Result<LogContentResponse>	流式双向游标分页读取日志内容（支持 forward/backward/last/tail/raw-* 方向及关键字、级别、时间范围过滤）
GET	/api/logs/raw	getLogRawContent	@RequestParam String fileName	Result<LogRawContentResponse>	读取日志文件完整原始内容（UTF-8 失败时回退 GBK，返回内容与文件大小）
GET	/api/logs/config	getLogConfig		Result<Map<String, Object>>	获取系统日志与访问日志配置（systemLog/accessLog 两部分）
PUT	/api/logs/config	updateLogConfig	@RequestBody Map<String, Object> config	Result<?>	更新日志配置（校验后保存系统/访问日志配置并下发到 AAS 实例应用，需重启生效）
POST	/api/logs/rotate	rotateLog	@RequestParam String fileName	Result<Void>	滚动（轮转）指定日志文件
DELETE	/api/logs/clear	clearLogs	@RequestParam	Result<Void>	清空指定日志文件内容

			tring fileName		
POST	/api/logs/export	export Log	@RequestBody Map<String, String> body	Result<Map<String, Object>>	生成日志导出下载链接（返回 downloadUrl 与 zip 文件名）
GET	/api/logs/download	downloadLog	@RequestParam String fileName, String logType, String keyword, String logLevel, String startTime, String endTime, HttpServletResponse response	void	下载日志 zip 文件（有过滤条件时仅导出匹配的日志条目）
POST	/api/logs/config/sync	syncLogConfig		Result<Map<String, Object>>	从 AAS 实例同步日志配置回控制台并返回最新配置

23 LoginLogController

基础路径: /api/audit/login-logs

方法	路径	函数	参数	返回类型	说明
GET	/api/audit/login-logs	list	LoginLogQueryDTO query	Result<PageResult<LoginLog>>	分页查询登录审计日志
GET	/api/audit/login-logs/export	export	LoginLogQueryDTO query, HttpServletResponse response	void	导出登录审计日志为 zip 包裹的 CSV（含事件时间、账号、事件类型、IP、描述、结果）

24 MetricsController

基础路径: /metrics

方法	路径	函数	参数	返回类型	说明
GET	/metrics	metrics	HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	Prometheus 指标端点 (text/plain) , 开关关闭时 404; 支持 Basic Auth / Access Token + IP 白名单
GET	/metrics/collectors	listCollectors	HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	列出所有 collector 名称与类目映射 (JSON, 需开启 Restful 开关并通过认证)
GET	/metrics/data	collectAll	HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	获取全部监控数据 (合并所有 collector) , 返回 Restful MetricsResponse
GET	/metrics/data/{category}	collectCategory	@PathVariable String category, HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	获取单类目 (category) 监控数据, 未知类目返回 404
GET	/metrics/data/{category}/{name}	collectOne	@PathVariable String category, @PathVariable String name, HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	获取单 collector (category/name) 监控数据, 未知 collector 返回 404

25 MonitorConfigController

基础路径: `/api/monitor/config`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/monitor/config</code>	<code>getConfig</code>		<code>Result<MonitorConfig></code>	获取监控服务配置 (SNMP v3 密码以星号掩码返回, Prometheus 凭据解密返回且密码掩码)
PUT	<code>/api/monitor/config</code>	<code>saveConfig</code>	<code>@RequestBody MonitorConfigDTO</code>	<code>Result<Void></code>	保存监控服务/告警/SNMP 配置 (不含接口服务字段; Full GC 监控关闭时停止采集器)
POST	<code>/api/monitor/config/restore-defaults</code>	<code>restoreDefaults</code>		<code>Result<MonitorConfig></code>	恢复监控服务默认配置
POST	<code>/api/monitor/config/prometheus-auth-token/refresh</code>	<code>refreshPrometheusAuthToken</code>		<code>Result<Map<String, String>></code>	重新生成 Prometheus Authorization Bearer token (明文仅本次响应返回, 存库加密)
PUT	<code>/api/monitor/config/interface-service</code>	<code>updateInterfaceService</code>	<code>@RequestBody InterfaceServiceConfigDTO</code>	<code>Result<MonitorConfig></code>	仅更新 Prometheus/Restful 接口服务配置 (开关、认证方式、Basic Auth 凭据、Token、白名单; Basic Auth 密码非空时校验复杂度)
POST	<code>/api/monitor/config/interface-service/restore-defaults</code>	<code>restoreInterfaceService</code>		<code>Result<MonitorConfig></code>	仅恢复接口服务相关字段为默认值

		iceDef aults			
--	--	-----------------	--	--	--

MonitorConfigDTO 字段说明（仅列业务字段，省略均为 0/1 开关的重复校验说明）

字段名	类型	中文说明
monitorEnabled	Integer	监控服务总开关 (0/1)
mbeanMonitorEnabled	Integer	MBean 监控开关 (0/1)
collectIntervalSeconds	Integer	采集周期 (秒, 10~3600)
jvmMonitorEnabled	Integer	JVM 监控开关 (0/1)
threadPoolMonitorEnabled	Integer	线程池监控开关 (0/1)
dsMonitorEnabled	Integer	JDBC 连接池监控开关 (0/1)
httpMonitorEnabled	Integer	HTTP 通道监控开关 (0/1)
sessionMonitorEnabled	Integer	会话监控开关 (0/1)
systemResourceMonitorEnabled	Integer	系统资源监控开关 (0/1)
fullGcMonitorEnabled	Integer	Full GC 监控开关 (0/1)
cpuMonitorEnabled	Integer	CPU 监控开关 (0/1)
memoryMonitorEnabled	Integer	内存监控开关 (0/1)
maxDataPoints	Integer	最大数据点数
replaySampleInterval	Integer	回放采样间隔 (秒, 1~86400)
alertEnabled	Integer	告警开关 (0/1)
alertIntervalMinutes	Integer	告警间隔 (分钟, 1~1440)
alertCpuThreshold	Integer	CPU 告警阈值 (0~100)
alertMemoryThreshold	Integer	内存告警阈值 (0~100)
alertDiskThreshold	Integer	磁盘告警阈值 (0~100)
httpThreadAlertEnabled	Integer	HTTP 线程告警开关 (0/1)
httpThreadThreshold	Integer	HTTP 线程告警阈值
cpuAlertEnabled	Integer	CPU 告警通知开关 (0/1)

virtualMemoryAlertEnabled	Integer	虚拟内存告警开关 (0/1)
virtualMemoryThreshold	Integer	虚拟内存告警阈值 (0~100)
physicalMemoryAlertEnabled	Integer	物理内存告警开关 (0/1)
physicalMemoryThreshold	Integer	物理内存告警阈值 (0~100)
fullGcAlertEnabled	Integer	Full GC 告警开关 (0/1)
fullGcDurationSeconds	Integer	Full GC 持续时间阈值 (秒)
fullGcFrequencyLimit	Integer	Full GC 频率限制 (次)
jdbcAlertEnabled	Integer	JDBC 告警开关 (0/1)
jdbcErrorThreshold	Integer	JDBC 错误阈值
alertSendMessageEnabled	Integer	告警发送消息开关 (0/1)
alertGenerateSnapshotEnabled	Integer	告警生成快照开关 (0/1)
snapshotThreadEnabled	Integer	线程快照开关 (0/1)
snapshotHeapEnabled	Integer	堆快照开关 (0/1)
snapshotProcessEnabled	Integer	进程快照开关 (0/1)
sendEmailEnabled	Integer	邮件通知开关 (0/1)
sendSnmpEnabled	Integer	SNMP 通知开关 (0/1)
sendPlatformEnabled	Integer	平台通知开关 (0/1)
snmpEnabled	Integer	SNMP 开关 (0/1)
snmpHost	String	SNMP Agent 绑定地址 (IPv4/IPv6 字面量, 最长 128)
snmpPort	Integer	SNMP 端口 (1~65535)
snmpCommunity	String	SNMP 团体名
snmpProtocol	String	SNMP 协议
snmpVersion	String	SNMP 版本
snmpAddress	String	NMS 端地址 (IP/port 格式)
snmpSecurity	String	SNMP v3 安全级别 (无/认证不加密/认证加密)
snmpUsername	String	SNMP v3 用户名

snmpAuthMethod	String	SNMP v3 认证方式 (MD5/SHA)
snmpAuthPassword	String	SNMP v3 认证密码 (明文入参, 服务端加密入库)
snmpCipher	String	SNMP v3 加密算法 (DES/3DES/AES128/AES192/AES256)
snmpPrivPassword	String	SNMP v3 加密密码
prometheusEnabled	Integer	Prometheus 接口开关 (0/1)
restfulEnabled	Integer	Restful 监控数据接口开关 (0/1)
prometheusAuthMethod	String	认证方式 (none/basic/authorization)
prometheusAuthUsername	String	Basic Auth 用户名 (明文入参, 服务端加密)
prometheusAuthPassword	String	Basic Auth 密码 (明文入参, 服务端加密)
prometheusAuthToken	String	Authorization Bearer 值 (明文入参, 服务端加密)
prometheusWhitelist	String	白名单原文 (多行/逗号分隔的 IP 或 CIDR, 含 IPv6)

InterfaceServiceConfigDTO 字段说明

字段名	类型	中文说明
prometheusEnabled	Integer	Prometheus 开关 (0=关闭 1=开启)
restfulEnabled	Integer	Restful 监控数据接口开关 (0=关闭 1=开启)
prometheusAuthMethod	String	认证方式 (none/basic/authorization)
prometheusAuthUsername	String	Basic Auth 用户名 (明文入参, 服务端加密; 仅字母数字_@-)
prometheusAuthPassword	String	Basic Auth 密码 (明文入参, 不允许含 <>; 非空时校验密码策略复杂度)
prometheusAuthToken	String	Authorization Bearer 值 (明文入参, 不允许含 <>)
prometheusWhitelist	String	白名单原文 (多行/逗号分隔的 IP 或 CIDR, 含 IPv6, 最长 512)

26 MonitorDataController

基础路径: /api/monitor/data

方法	路径	函数	参数	返回类型	说明
GET	/api/monitor/data/system	getSystemResource		Result<Map<String, Object>>	获取系统资源实时监控数据（CPU/内存/磁盘等，模块开关关闭时返回null)
GET	/api/monitor/data/system-resource/history	getSystemResourceHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime	Result<SystemResourceTrendDTO>	查询系统资源历史趋势数据（默认最近 1 小时，最长 30 天)
GET	/api/monitor/data/jvm	getJvmMonitor		Result<Map<String, Object>>	获取 JVM 内存/垃圾回收等实时监控数据
GET	/api/monitor/data/jvm/history	getJvmHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime	Result<JvmHistoryDTO>	查询 JVM 监控历史数据（内存、GC 等趋势)
GET	/api/monitor/data/threads	getThreadInfo		Result<Map<String, Object>>	获取 JVM 线程实时监控信息
GET	/api/monitor/data/fullgc/records	fullGcRecords	@RequestParam LocalDateTime startTime, LocalDateTime endTime, Integer current, Integer size	Result<Page<FullGcRecordDTO>>	分页查询 Full GC 记录（默认最近 1 天，最长 30 天)
GET	/api/monitor/data/fullgc/trend	fullGcTrend	@RequestParam LocalDateTime startTime, LocalDateTime endTime	Result<FullGcTrendDTO>	查询 Full GC 趋势统计

GET	/api/monitor/data/fullgc/status	fullGcStatus		Result<Map<String, Object>>	获取 Full GC 采集器状态 (supported 是否支持、listening 是否监听中)
POST	/api/monitor/data/fullgc/switch/on	fullGcSwitchOn		Result<Void>	启动 Full GC 采集监听
POST	/api/monitor/data/fullgc/switch/off	fullGcSwitchOff		Result<Void>	停止 Full GC 采集监听
GET	/api/monitor/data/fullgc/export	exportFullGcRecords	@RequestParam LocalDateTime startTime, LocalDateTime endTime, HttpServletResponse response	void	导出 Full GC 记录为 CSV 文件
GET	/api/monitor/data/jndi	getJndiTree		Result<Map<String, Object>>	获取 JNDI 树结构
GET	/api/monitor/data/sql	getSqlTrace		Result<Map<String, Object>>	获取 SQL 追踪监控数据
GET	/api/monitor/data/perf	getPerfData		Result<Map<String, Object>>	获取性能采集数据
GET	/api/monitor/data/datasource	getDataSourceMonitor		Result<Map<String, Object>>	获取数据源连接池实时监控数据
GET	/api/monitor/data/datasource-replay	getDataSourceReplay		Result<Map<String, Object>>	获取数据源监控数据 (回放场景, 不受开关限制)
POST	/api/monitor/data/datasource/test	testDataSource	@RequestParam String jndiName	Result<Boolean>	测试指定 JNDI 数据源连通性

GET	/api/monitor/data/datasource/history	getDataSourceHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime, String jndiName	Result<DataSourceTrendDTO>	查询数据源历史趋势（可按 JNDI 名过滤）
GET	/api/monitor/data/http	getHttpChannelMonitor		Result<Map<String, Object>>	获取 HTTP 通道实时监控数据
GET	/api/monitor/data/http-replay	getHttpChannelReplay		Result<Map<String, Object>>	获取 HTTP 通道监控数据（回放场景）
GET	/api/monitor/data/threadpool	getThreadPoolMonitor		Result<Map<String, Object>>	获取线程池实时监控数据
GET	/api/monitor/data/threadpool-replay	getThreadPoolReplay		Result<Map<String, Object>>	获取线程池监控数据（回放场景）
GET	/api/monitor/data/threadpool/history	getThreadPoolHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime, String poolName	Result<ThreadPoolHistoryDTO>	查询线程池历史数据（可按线程池名过滤）
GET	/api/monitor/data/threadpool/{name}/detail	getThreadPoolDetail	@PathVariable String name	Result<Map<String, Object>>	获取指定线程池详细信息
GET	/api/monitor/data/app-sessions	getAppSessionMonitor		Result<Map<String, Object>>	获取应用会话实时监控数据
GET	/api/monitor/data/app-sessions-replay	getAppSessionReplay		Result<Map<String, Object>>	获取应用会话监控数据（回放场景）

		onReply			
GET	/api/monitor/data/app-sessions/history	getAppSessionHistory	@RequestParam <code>LocalDateTime startTime</code> , <code>LocalDateTime endTime</code> , <code>String contextPath</code>	Result<AppSessionTrendDTO>	查询应用会话历史趋势（可按应用上下文路径过滤）
GET	/api/monitor/data/http/trend	getHttpChannelTrend	@RequestParam <code>String port</code>	Result<Map<String, Object>>	获取 HTTP 通道请求趋势数据（可按端口过滤）
GET	/api/monitor/data/http/history	getHttpChannelHistory	@RequestParam <code>LocalDateTime startTime</code> , <code>LocalDateTime endTime</code> , <code>String port</code>	Result<HttpChannelHistoryDTO>	查询 HTTP 通道历史数据（可按端口过滤）
GET	/api/monitor/data/http-connections	getHttpConnections		Result<Map<String, Object>>	获取 HTTP 连接数数据（仪表盘/监控共用）

27 NotificationController

基础路径: /api/notifications

方法	路径	函数	参数	返回类型	说明
GET	/api/notifications	list	@RequestParam Integer page, Integer size, Integer readStatus	Result<PageResult<Notification>>	分页查询当前用户的通知列表（可按已读状态过滤）
GET	/api/notifications/unread-count	unreadCount		Result<Map<String, Object>>	获取当前用户未读通知数量
PUT	/api/notifications/{id}/read	markRead	@PathVariable Long id	Result<Void>	将指定通知标记为已读
PUT	/api/notifications/read-all	markAllRead		Result<Void>	将当前用户全部通知标记为已读
POST	/api/notifications	create	@RequestBody Map<String, String> body (title/type/content/severity/targetUser)	Result<Notification>	创建通知（可指定目标用户，审计记录）
DELETE	/api/notifications	deleteBatch	@RequestBody List<Long> ids	Result<Integer>	批量删除通知，返回删除条数

28 PasswordController

基础路径: /api/auth

方法	路径	函数	参数	返回类型	说明
GET	/api/auth/password-policy	getPasswordPolicy		Result<PasswordPolicyDTO>	获取当前系统密码策略
POST	/api/auth/change-password	changePassword	@RequestBody changePasswordRequest request	Result<Void>	修改当前登录用户密码（AES 加密传输，校验旧密码、密码策略、历史密码重复；成功后强制该用户所有会话失效）

ChangePasswordRequest 字段说明

字段名	类型	中文说明
oldPassword	String	旧密码（AES 加密后的密文）
newPassword	String	新密码（AES 加密后的密文）
confirmPassword	String	确认新密码（AES 加密后的密文，须与 newPassword 一致）

29 PatchController

基础路径: /api/system/patches

方法	路径	函数	参数	返回类型	说明
GET	/api/system/patches	list	PageQueryDTO query	Result<PageResult<Patch>>	分页查询补丁列表
POST	/api/system/patches/upload	upload	@RequestParam String name, String version, String platform, String description, MultipartFile file	Result<Void>	上传补丁包 (multipart 表单)
POST	/api/system/patches/{id}/apply	apply	@PathVariable Long id, String upgradeNote, Integer forceUpgrade	Result<Void>	应用 (升级) 指定补丁, 可附升级说明与强制升级标志
POST	/api/system/patches/{id}/rollback	rollback	@PathVariable Long id, String rollbackNote	Result<Void>	回滚指定补丁
DELETE	/api/system/patches/{id}	delete	@PathVariable Long id	Result<Void>	删除指定补丁
GET	/api/system/patches/{id}/progress	getProgress	@PathVariable Long id	Result<Map<String, Object>>	获取补丁应用/回滚进度

Patch 字段说明

字段名	类型	中文说明
id	Long	补丁ID
patchName	String	补丁名称
version	String	补丁版本
platform	String	适用平台版本
description	String	补丁描述

filePath	String	补丁文件路径
status	Integer	补丁状态（0=未应用，1=已应用，2=失败，3=已回退）
progress	Integer	安装进度百分比
installTime	LocalDateTime	安装时间
rolledBackTime	LocalDateTime	回退时间
upgradeNote	String	升级备注
rollbackNote	String	回退备注
forceUpgrade	Integer	强制升级标志
progressDetail	String	进度详情（JSON格式）

30 ReplayController

基础路径: /api/monitor/replay

方法	路径	函数	参数	返回类型	说明
GET	/api/monitor/replay/system	system	@RequestParam s tring start, String end	Result<Map<Strin g, Object>>	查询系统维度（主 机/系统资源）历 史采样点回放数据
GET	/api/monitor/replay/jvm	jvm	@RequestParam s tring start, String end	Result<Map<Strin g, Object>>	查询 JVM 维度历 史采样点回放数据
GET	/api/monitor/replay/datasource	dataso urce	@RequestParam s tring start, String end, String target	Result<Map<Strin g, Object>>	查询指定数据源（t arget）的历史采样 点回放数据
GET	/api/monitor/replay/app-session	appSe ssion	@RequestParam s tring start, String end, String target	Result<Map<Strin g, Object>>	查询指定应用（tar get）会话数历史 采样点回放数据
GET	/api/monitor/replay/http-channel	httpCh annel	@RequestParam s tring start, String end, String target	Result<Map<Strin g, Object>>	查询指定 HTTP 通 道（target）的历 史采样点回放数据
GET	/api/monitor/replay/thread-pool	thread Pool	@RequestParam s tring start, String end, String target	Result<Map<Strin g, Object>>	查询指定线程池（t arget）的历史采样 点回放数据

31 ResourceController

基础路径: /api/resources

方法	路径	函数	参数	返回类型	说明
GET	/api/resources/permission-tree	permissionTree		Result<List<PermissionTreeNode>>	获取权限树（模块 → 页面 → 操作三级结构），用于角色权限配置

PermissionTreeNode 字段说明

字段名	类型	中文说明
code	String	节点编码（菜单编码或 API 权限编码，如 app_list:create）
name	String	节点显示名称
nameKey	String	i18n 名称 key（menu:xxx 或 permission:xxx）
resourceId	Long	关联资源 ID（菜单节点才有）
children	List	子节点列表（页面/操作）

32 RoleController

基础路径: /api/roles

方法	路径	函数	参数	返回类型	说明
GET	/api/roles	list	PageQueryDTO query, @RequestParam String roleCode, Integer builtin	Result<PageResult<RoleDTO>>	分页查询角色列表（可按角色编码、是否内置过滤）
GET	/api/roles/all	listAll		Result<List<RoleDTO>>	查询全部角色（不分页）
GET	/api/roles/{id}	get	@PathVariable Long id	Result<RoleDTO>	按 ID 查询角色详情
POST	/api/roles	create	@RequestBody RoleDTO dto	Result<Void>	创建角色（Create 校验分组，空字符串字段规范化为 null）
PUT	/api/roles/{id}	update	@PathVariable Long id, @RequestBody RoleDTO dto	Result<Void>	更新指定角色
DELETE	/api/roles/{id}	delete	@PathVariable Long id	Result<Void>	删除指定角色
DELETE	/api/roles/batch	deleteBatch	@RequestBody List<Long> ids	Result<Integer>	批量删除角色，返回删除条数

RoleDTO 字段说明

字段名	类型	中文说明
id	Long	角色 ID
roleCode	String	角色编码（创建必填，仅字母/数字/下划线/短横线，最长 64）
roleName	String	角色显示名（最长 64）
description	String	角色说明（最长 255）
builtin	Integer	是否内置角色（1=内置，不可修改删除）

classLevel	Integer	角色密级标识
resourceIds	List	关联的资源 ID 列表
resourceCodes	List	关联的资源编码列表（用于显示）
userNames	List	关联的用户名列表
userCount	Integer	关联用户数（列表展示用）
defaultRole	Integer	是否默认角色（仅一个角色可为默认）
permissions	List	权限编码列表（含菜单和操作权限）

33 ScriptRecordingController

基础路径: /api/script-recording

方法	路径	函数	参数	返回类型	说明
GET	/api/script-recording	list	@RequestParam s tring name, Strin g status, LocalDate Time startTime, Lo calDateTime endTi me, long page, lo ng size	Result<PageResult <ScriptRecording VO>>	分页查询脚本录制 任务列表，支持名 称/状态/时间范围 过滤
GET	/api/script-recor ding/active	active	Authentication aut hentication	Result<Boolean>	当前用户是否有录 制中任务（注销确 认用，仅返回自己 的状态）
POST	/api/script-recor ding	create	@RequestBody sc riptRecordingCrea tedTo dto, Authen tication authentica tion	Result<ScriptReco rdingVO>	创建脚本录制任务 （以当前用户为创 建者），记审计
POST	/api/script-recor ding/{id}/stop	stop	@PathVariable Lon g id, Authenticatio n authentication	Result<ScriptReco rdingStopResult>	停止录制并生成脚 本（aascli/postma n/curl）
POST	/api/script-recor ding/{id}/stop-em pty	stopE mpty	@PathVariable Lon g id, Authenticatio n authentication	Result<Void>	停止录制但不生成 脚本（空任务丢 弃），记审计
DELET E	/api/script-recor ding/{id}	delete	@PathVariable Lon g id, Authenticatio n authentication	Result<Void>	删除录制任务（仅 创建者本人）
GET	/api/script-recor ding/{id}/downloa d	downl oad	@PathVariable Lon g id, @RequestPar am List<String> f ormats	ResponseEntity<b yte[]>	下载录制生成的脚 本 ZIP 包（默认打 包 aascli/postma n/curl 三种格式）

ScriptRecordingCreateDTO 字段说明

字段名	类型	中文说明
name	String	脚本名称（字母/数字/下划线/短横线/点，1-255 字符）
timeoutMinutes	Integer	超时时间（分钟），0=无限制
description	String	说明，可空，不超过 1000 字符

ScriptRecordingVO 字段说明

字段名	类型	中文说明
id	Long	录制任务 ID
name	String	脚本名称
status	String	任务状态（录制中/已停止等）
description	String	说明
timeoutMinutes	Integer	超时时间（分钟）
creator	String	创建者用户名
startTime	LocalDateTime	开始录制时间
endTime	LocalDateTime	结束录制时间
opCount	Integer	已录制操作数
elapsedSeconds	long	已录制时长（秒），录制中实时计算

34 SecurityConfigController

基础路径: /api/security/config

方法	路径	函数	参数	返回类型	说明
GET	/api/security/config	getConfigs		Result<Map<String, String>>	获取全部安全配置项 (smtp_password 统一以加密串返回)
PUT	/api/security/config	updateConfigs	@RequestBody Map<String, String> configs, @RequestHeader String X-Current-User-Email	Result<Map<String, Object>>	批量更新安全配置 (backup_enabled 由关转开时立即执行一次备份; smtp_verify_enabled 开启时自动发送测试邮件并返回结果)
POST	/api/security/config/test-mail	testMail	@RequestBody Map<String, String> body (to)	Result<Map<String, Object>>	发送 SMTP 测试邮件到指定收件人
GET	/api/security/config/smtp	getSmtpEnabled		Result<Map<String, Object>>	获取邮件验证开关状态
PUT	/api/security/config/smtp	updateSmtpEnabled	@RequestBody Map<String, Object> body (smtpEnabled)	Result<Void>	更新邮件验证开关
GET	/api/security/config/auth-policy	getAuthPolicy		Result<Map<String, Object>>	获取认证策略配置
PUT	/api/security/config/auth-policy	updateAuthPolicy	@RequestBody Map<String, Object> policy	Result<Void>	保存认证策略配置
POST	/api/security/config/backup-now	backupNow		Result<Map<String, Object>>	立即执行一次业务配置手动备份

GET	/api/security/config/backup-list	listBackups		Result<List<Map<String, Object>>>	查询业务配置备份文件列表
POST	/api/security/config/backup-restore	restoreBackup	@RequestBody Map<String, String> body (fileName)	Result<Map<String, Object>>	按文件名恢复业务配置备份
DELETE	/api/security/config/backup/{fileName}	deleteBackup	@PathVariable String fileName	Result<Void>	删除指定业务配置备份文件
POST	/api/security/config/restart	restart		Result<Map<String, Object>>	重启 AAS 服务器 (返回 RESTARTING 状态)

SecurityConfig 字段说明

字段名	类型	中文说明
id	Long	配置ID
configKey	String	配置项键名
configValue	String	配置项值
description	String	配置说明

35 SecurityRealmController

基础路径: /api/security/realms

方法	路径	函数	参数	返回类型	说明
GET	/api/security/realms	list	SecurityRealmQueryDTO query	Result<PageResult<SecurityRealmDTO>>	分页查询安全域列表
GET	/api/security/realms/all	listAll		Result<List<SecurityRealmDTO>>	查询全部安全域 (不分页)
GET	/api/security/realms/{id}	get	@PathVariable Long id	Result<SecurityRealmDTO>	按 ID 查询安全域详情 (不存在返回 404)
POST	/api/security/realms	create	@RequestBody SecurityRealmDTO dto	Result<Void>	创建安全域 (需重启生效)
PUT	/api/security/realms/{id}	update	@PathVariable Long id, @RequestBody SecurityRealmDTO dto	Result<Void>	更新安全域 (需重启生效)
DELETE	/api/security/realms/{id}	delete	@PathVariable Long id	Result<Void>	删除安全域 (需重启生效)
DELETE	/api/security/realms/batch	deleteBatch	@RequestBody List<Long> ids	Result<Integer>	批量删除安全域, 返回删除条数
GET	/api/security/realms/user-database-resources	getUserDatabaseResources		Result<List<String>>	获取 AAS 实例中可用的用户数据库资源名列表

SecurityRealmDTO 字段说明

字段名	类型	中文说明
id	Long	安全域 ID
name	String	安全域名称

type	String	类型 (DataSourceRealm/UserDatabaseRealm/JAASRealm/JNDIRealm/Cu
config	String	配置内容 (JSON 或自定义格式)
xmlConfig	String	XML 配置输出 (用于 server.xml)
description	String	描述
passwordHashAlgorithm	String	密码哈希算法 (无/MD5/SHA-1/SHA-256/SHA-512)
boundApps	List	绑定应用名称列表
lockEnabled	Boolean	是否启用账号锁定机制
maxFailedAttempts	Integer	允许密码错误次数
lockTimeout	Integer	锁定超时时间 (秒)
cacheSize	Integer	缓存大小
cacheRemovalWarningTime	Integer	缓存移除警告时间 (秒)
userTable	String	用户表名 (DataSource Realm)
usernameField	String	用户名字段
passwordField	String	密码字段
roleTable	String	角色表名
roleField	String	角色字段
dataSourceJndi	String	数据源 JNDI 名称 (DataSource Realm)
resourceName	String	资源名 (UserDatabase Realm)
jaasLoginModules	List	JAAS LoginModule 配置列表 (含 loginModuleClass/controlFlag/optic
userPrincipalClass	String	用户 Principal 实现类 (JAAS Realm)
rolePrincipalClass	String	角色 Principal 实现类
contextFactory	String	上下文工厂类名 (JNDI Realm)
jndiConnectionUrl	String	连接 URL
alternateUrl	String	备用连接 URL
connectionUsername	String	连接用户名

connectionPassword	String	连接密码
userSearchBase	String	用户搜索基准
userSearchFilter	String	用户搜索匹配
userSubtree	Boolean	用户子树搜索
userPattern	String	用户 DN 模式
userSearchAsUser	Boolean	以用户身份搜索用户
userPasswordAttr	String	用户密码属性名
userRoleName	String	用户条目中的角色属性名
userRoleAttribute	String	用户角色属性（用于 roleSearch 替换）
roleSearchBase	String	角色搜索基准
roleSearchFilter	String	角色搜索匹配
roleNameAttribute	String	角色名称属性
roleSubtree	Boolean	角色子树搜索
roleNested	Boolean	嵌套角色搜索
roleSearchAsUser	Boolean	以用户身份搜索角色
commonRole	String	通用角色
connectionTimeout	String	连接超时（毫秒）
readTimeout	String	读取超时（毫秒）
connectionPoolSize	Integer	连接池大小（须大于 0）

36 SessionManagerController

基础路径: /api/config/web/sessions

方法	路径	函数	参数	返回类型	说明
GET	/api/config/web/sessions/all	listAll		Result<List<SessionManager>>	查询全部会话管理器（不分页）
GET	/api/config/web/sessions/redis-plugin-status	redisPluginStatus		Result<Map<String, Boolean>>	获取 Redis 会话插件状态 (loaded/jarsInstalled/commonLoaderConfigured)
GET	/api/config/web/sessions	list	@RequestParam int page, int size, String name, String type	Result<PageResult<SessionManagerListItem>>	分页查询会话管理器列表（可按名称、类型过滤）
GET	/api/config/web/sessions/{id}	get	@PathVariable Long id	Result<SessionManagerListItem>	按 ID 查询会话管理器详情（不存在返回 404）
POST	/api/config/web/sessions	create	@RequestBody SessionManager sm	Result<Void>	创建会话管理器（含名称/类型/Redis/路径等字段级校验，名称重复报错）
PUT	/api/config/web/sessions/{id}	update	@PathVariable Long id, @RequestBody SessionManager sm	Result<Void>	更新会话管理器（同 create 校验，排除自身查重）
DELETE	/api/config/web/sessions/{id}	delete	@PathVariable Long id	Result<Void>	删除会话管理器（逻辑删除，预存名称供审计）

SessionManager 字段说明（不含 BaseEntity 公共字段）

字段名	类型	中文说明
name	String	会话管理器名称（必填，最长 128）

type	String	类型 (memory/file/redis/database)
maxSessions	Integer	最大活跃会话数 (-1 表示不限制, 否则 ≥ 1)
persistPath	String	会话持久化文件路径 (memory 类型, 仅允许相对路径)
saveOnRestart	Integer	重启时是否保存会话
maxIdleBackup	Integer	空闲备份阈值 (秒, -1 或 ≥ 0)
minIdleSwap	Integer	最小空闲交换阈值 (秒)
maxIdleSwap	Integer	最大空闲交换阈值 (秒)
persistenceDir	String	持久化目录 (file 类型, 仅允许相对路径)
redisMode	String	Redis 模式 (standalone/sentinel)
redisNodes	String	Redis 节点 (host:port, 多节点逗号分隔)
redisMasterName	String	哨兵模式主节点名称
redisPassword	String	Redis 密码 (最长 256)
redisDatabase	Integer	Redis 数据库索引 (0~15)
redisConnectTimeout	Integer	Redis 连接超时 (毫秒)
redisMaxConn	Integer	Redis 最大连接数 (≥ 1)
redisMinIdle	Integer	Redis 最小空闲连接数 (≥ 0)
redisMaxIdle	Integer	Redis 最大空闲连接数
redisMaxWait	Integer	Redis 最大等待时间 (毫秒)
sessionPersistPolicies	String	会话持久化策略 (SAVE_ON_CHANGE/ALWAYS_SAVE_AFTER_REQUEST, 逗号组合)
sessionAttributeNameFilter	String	会话属性名过滤器 (正则表达式)
sessionAttributeValueClassNameFilter	String	会话属性值类名过滤器 (正则表达式)
warnOnFilterFailure	Integer	过滤器失败时是否告警
notifyAttributeOnUnchanged	Integer	属性未变更时是否通知
notifyBindingOnUnchanged	Integer	绑定未变更时是否通知

dataSourceName	String	数据源名称 (database 类型必填)
localDataSource	Integer	是否使用本地数据源

37 SharedLibController

基础路径: `/api/resources/sharedlibs`

方法	路径	函数	参数	返回类型	说明
GET	<code>/api/resources/sharedlibs</code>	list	SharedLibQueryDTO query	Result<PageResult<SharedLib>>	分页查询共享库列表
GET	<code>/api/resources/sharedlibs/{id}</code>	get	@PathVariable Long id	Result<SharedLib>	按 ID 查询共享库详情
POST	<code>/api/resources/sharedlibs</code>	create	@RequestBody SharedLib lib	Result<Long>	创建共享库, 返回新记录 ID
PUT	<code>/api/resources/sharedlibs/{id}</code>	update	@PathVariable Long id, @RequestBody SharedLib lib	Result<Void>	更新共享库
DELETE	<code>/api/resources/sharedlibs/{id}</code>	delete	@PathVariable Long id	Result<Void>	删除共享库
POST	<code>/api/resources/sharedlibs/{id}/upload-jar</code>	uploadJar	@PathVariable Long id, @RequestParam MultipartFile file	Result<Void>	向共享库上传 JAR 文件
GET	<code>/api/resources/sharedlibs/{id}/jars</code>	listJarFiles	@PathVariable Long id	Result<List<Map<String, Object>>>	列出共享库下已上传的 JAR 文件
GET	<code>/api/resources/sharedlibs/{id}/jars/{fileName}/download</code>	downloadJar	@PathVariable Long id, @PathVariable String fileName, HttpServletResponse response	void	下载共享库中指定 JAR 文件
DELETE	<code>/api/resources/sharedlibs/{id}/jars/{fileName}</code>	deleteJarFile	@PathVariable Long id, @PathVariable String fileName	Result<Void>	删除共享库中指定 JAR 文件
POST	<code>/api/resources/sharedlibs/{id}/jars/delete-batch</code>	deleteJarFiles	@PathVariable Long id, @RequestBo	Result<Void>	批量删除共享库中的 JAR 文件

			dy List<String> fileNames		
POST	/api/resources/sharedlibs/{id}/apps	bindApp	@PathVariable Long id, @RequestParam Long appld	Result<Void>	将共享库绑定到指定应用
DELETE	/api/resources/sharedlibs/{id}/apps	unbindApp	@PathVariable Long id, @RequestParam Long appld	Result<Void>	解除共享库与应用的绑定
GET	/api/resources/sharedlibs/{id}/apps	listBoundApps	@PathVariable Long id	Result<List<Map<String, Object>>>	查询共享库已绑定的应用列表
PUT	/api/resources/sharedlibs/{id}/priority	updatePriority	@PathVariable Long id, @RequestParam Integer priority	Result<Void>	更新共享库加载优先级 (0~9999, 数值越小优先级越高)

SharedLib 字段说明 (不含 BaseEntity 公共字段)

字段名	类型	中文说明
name	String	共享库名称 (必填, 最长 128)
path	String	共享库路径 (最长 512)
version	String	版本号 (最长 64)
description	String	描述
classLevel	Integer	类级别 (≥ 0)
status	Integer	状态 (0/1)
loadStrategy	String	类加载策略 (application-first/classes-first/shared-first)
sortOrder	Integer	排序值 (≥ 0)
priority	Integer	加载优先级 (0~9999, 数值越小优先级越高, 默认 0)
fileCount	Integer	文件数 (非持久化字段, 列表展示用)
refCount	Integer	引用次数 (非持久化字段, 列表展示用)

38 SnapshotConfigController

基础路径: /api/monitor/snapshot-config

方法	路径	函数	参数	返回类型	说明
GET	/api/monitor/snapshot-config	getConfig		Result<SnapshotConfig>	查询自动快照配置 (当前生效配置)
PUT	/api/monitor/snapshot-config	saveConfig	@RequestBody SnapshotConfigDTO dto	Result<Void>	保存自动快照配置 (含内容与触发规则), 记审计日志
POST	/api/monitor/snapshot-config/restore-defaults	restoreDefaults		Result<SnapshotConfig>	恢复自动快照配置 为默认值

SnapshotConfigDTO 字段说明

字段名	类型	中文说明
id	Long	配置记录 ID
autoEnabled	Integer	自动快照开关 (0=关 1=开)
contentAccessLog	Integer	快照内容: 访问日志开关
contentGcLog	Integer	快照内容: GC日志开关
contentSystemConfig	Integer	快照内容: 系统配置开关
contentSystemLog	Integer	快照内容: 系统日志开关
contentHostMonitor	Integer	快照内容: 主机监控数据开关
contentInstanceMetrics	Integer	快照内容: 实例监视量开关
contentCoreDump	Integer	快照内容: 系统Coredump开关
contentThreadDump	Integer	快照内容: 线程栈开关
threadStackCount	Integer	线程栈抓取次数 (1-10)
contentHeapDump	Integer	快照内容: Heapdump开关
heapdumpMode	String	Heapdump模式 (all/live)

maxSnapshots	Integer	最多保存快照数 (1-1000)
intervalMinutes	Integer	生成间隔时间 (分钟, 1-1440)
ruleCpuThreshold	Double	CPU使用率阈值 (0-100%)
ruleCpuOccurrences	Integer	CPU连续发生次数
ruleMemoryThreshold	Double	内存使用率阈值 (0-100%)
ruleMemoryOccurrences	Integer	内存连续发生次数
ruleFullGcCountThreshold	Integer	FullGC次数阈值
ruleFullGcCountOccurrences	Integer	FullGC次数连续发生次数
ruleFullGcTimeThreshold	Integer	FullGC时间阈值
ruleFullGcTimeOccurrences	Integer	FullGC时间连续发生次数
ruleHttpTimeoutThreshold	Integer	HTTP慢请求数阈值
ruleHttpTimeoutOccurrences	Integer	HTTP慢请求数连续发生次数
ruleHttpWaitThreshold	Integer	HTTP等待任务数阈值
ruleHttpWaitOccurrences	Integer	HTTP等待任务数连续发生次数
ruleHttpErrorThreshold	Integer	HTTP错误数阈值
ruleHttpErrorOccurrences	Integer	HTTP错误数连续发生次数
ruleJdbcWaitThreshold	Integer	JDBC等待请求数阈值
ruleJdbcWaitOccurrences	Integer	JDBC等待请求数连续发生次数
ruleJdbcLeakThreshold	Integer	JDBC疑似泄露连接数阈值
ruleJdbcLeakOccurrences	Integer	JDBC疑似泄露连接数连续发生次数

39 SnapshotController

基础路径: /api/monitor/snapshots

方法	路径	函数	参数	返回类型	说明
GET	/api/monitor/snapshots	getSnapshotList	SnapshotQueryDTO query	Result<PageResult<Snapshot>>	分页查询快照列表
POST	/api/monitor/snapshots	createSnapshot	@RequestBody snapshotCreateDTO dto	Result<Void>	创建手动快照任务 (异步生成)
DELETE	/api/monitor/snapshots/{id}	deleteSnapshot	@PathVariable Long id	Result<Void>	删除指定快照
GET	/api/monitor/snapshots/{id}/download/file	downloadSnapshotFile	@PathVariable Long id, HttpServletResponse response	void	下载快照 zip 文件 (pending/running 状态禁止下载, 文件不存在报错)

SnapshotCreateDTO 字段说明

字段名	类型	中文说明
contentTypes	List	快照内容类型列表 (至少选择一项, 如线程栈/堆/进程等)
threadStackCount	Integer	线程栈抓取次数 (1~10)
heapdumpMode	String	Heapdump 模式 (all=全部对象 / live=仅存活对象)

40 SnmpConfigController

基础路径: /api/snmp-config

方法	路径	函数	参数	返回类型	说明
GET	/api/snmp-config	getConfig		Result<MonitorConfig>	获取 SNMP 代理配置（仅 SNMP 业务字段视图，已保存密码以 15 个星号占位，不含 Prometheus 等无关敏感字段）
PUT	/api/snmp-config	saveConfig	@RequestBody @Valid SnmpConfigDTO dto	Result<Void>	保存 SNMP 配置（密码留空表示保留原值，明文单次加密入库；审计携带 before/after 快照）
POST	/api/snmp-config/restore-defaults	restoreDefaults		Result<Void>	恢复 SNMP 配置为默认值

SnmpConfigDTO 字段说明

字段名	类型	中文说明
snmpEnabled	Integer	SNMP 开关（0 关 / 1 开）
snmpHost	String	SNMP 绑定地址（必填，最长 128）
snmpPort	Integer	SNMP 绑定端口（1-65535）
snmpProtocol	String	传输协议（udp / tcp）
snmpVersion	String	SNMP 版本（v2c / v3）
snmpCommunity	String	SNMP 团体名（v2c 用，最长 255）
snmpSecurity	String	SNMP v3 安全级别
snmpUsername	String	SNMP v3 用户名（最长 255）

snmpAuthMethod	String	SNMP v3 认证方式
snmpAuthPassword	String	SNMP v3 认证密码（最长 255，留空保留原值）
snmpCipher	String	SNMP v3 加密算法
snmpPrivPassword	String	SNMP v3 加密密码（最长 255，留空保留原值）

41 SqlTraceController

基础路径: /api/monitor/sql-trace

方法	路径	函数	参数	返回类型	说明
GET	/api/monitor/sql-trace/config	getConfig		Result<SqlTraceConfig>	获取 SQL 追踪配置
POST	/api/monitor/sql-trace/config	saveConfig	@RequestBody @Valid SqlTraceConfig config	Result<Void>	保存 SQL 追踪配置 (全局开关/阈值/堆栈开关/最大记录量/数据源配置)
POST	/api/monitor/sql-trace/start	start		Result<Void>	启动 SQL 追踪
POST	/api/monitor/sql-trace/stop	stop		Result<Void>	停止 SQL 追踪
GET	/api/monitor/sql-trace/records	query	SqlTraceQueryDTO query (绑定查询参数, @Valid)	Result<PageResult<SqlTraceRecord>>	分页查询 SQL 追踪记录
GET	/api/monitor/sql-trace/records/export	export	SqlTraceQueryDTO query, HttpServletResponse response	void	按查询条件导出 SQL 追踪记录 CSV 文件 (先校验查询条件)

SqlTraceConfig 字段说明 (@RequestBody 实体, id 为固定主键, 非业务语义)

字段名	类型	中文说明
enabled	Integer	全局开关 (0 关 / 1 开, 必填)
thresholdMs	Integer	执行时长阈值 (毫秒), 超过才记录 (必填, 非负)
stackEnabled	Integer	是否记录调用堆栈 (0 关 / 1 开, 必填)
maxRecordCount	Integer	最大 SQL 记录量 (必填, ≥1)
datasourceConfigs	String	各数据源独立配置 (JSON 串)

SqlTraceQueryDTO 字段说明 (继承 PageQueryDTO 的分页字段 page/size)

字段名	类型	中文说明
startTime	LocalDateTime	执行时间起 (yyyy-MM-dd HH:mm:ss)
endTime	LocalDateTime	执行时间止
datasource	String	数据源名称筛选
event	String	事件类型筛选
keyword	String	关键字筛选 (SQL 文本等模糊匹配)

SqlTraceRecord 字段说明 (返回实体)

字段名	类型	中文说明
id	Long	记录 ID
event	String	事件类型
sqlText	String	SQL 文本
stackTrace	String	调用堆栈
datasource	String	所属数据源
executeCount	Integer	执行次数
avgTimeMs	Integer	平均执行耗时 (毫秒)
maxTimeMs	Integer	最大执行耗时 (毫秒)
executeTime	LocalDateTime	SQL 执行时间
createdTime	LocalDateTime	记录创建时间

42 SystemController

基础路径: /api/system

方法	路径	函数	参数	返回类型	说明
GET	/api/system/version	getVersion		Result<Map<String, String>>	获取版本信息 (currentVersion/buildTime/javaVersion)
GET	/api/system/upgradeHistory	getUpgradeHistory		Result<List<Map<String, Object>>>	实例升级历史列表 (占位实现, 当前返回空列表)
POST	/api/system/upgrade	upgrade	@RequestBody Map<String, Object> body	Result<Map<String, Object>>	提交实例升级请求 (占位实现, 返回 upgradeId/status/message)
POST	/api/system/restart	restart		Result<Map<String, Object>>	重启 AAS 服务器, 返回 status=RESTARTING
GET	/api/system/restart-pending	getRestartPending		Result<Map<String, Object>>	查询待生效的配置变更 (needsRestart 与 changes 列表)

43 ThreadDiagnosisController

基础路径: `/api/monitor/thread-diagnosis`

方法	路径	函数	参数	返回类型	说明
POST	<code>/api/monitor/thread-diagnosis/start</code>	start	@RequestBody ThreadDiagnosisStartRequest request	Result<Void>	启动线程诊断采样 (intervalSec 必填, 1~3600 秒; maxDataMb 控制采样数据上限)
POST	<code>/api/monitor/thread-diagnosis/stop</code>	stop		Result<Void>	停止线程诊断采样
POST	<code>/api/monitor/thread-diagnosis/session-end</code>	endSession		Result<Void>	会话结束 (跳走/关闭页面/session过期) 时安静停止监控并清理诊断任务, 不生成数据文件
GET	<code>/api/monitor/thread-diagnosis/snapshot</code>	getSnapshot		Result<Map<String, Object>>	获取当前线程诊断快照 (线程列表及状态数据)
GET	<code>/api/monitor/thread-diagnosis/thread/{id}/stack</code>	getThreadStack	@PathVariable("id") long threadId	Result<Map<String, Object>>	获取指定线程的堆栈信息
POST	<code>/api/monitor/thread-diagnosis/thread/{id}/stop</code>	stopThread	@PathVariable("id") long threadId	Result<Void>	停止指定线程
POST	<code>/api/monitor/thread-diagnosis/analyze</code>	analyze	@RequestParam (defaultValue="live") String source	Result<Map<String, Object>>	同步执行线程诊断分析 (source 默认 live 实时数据)
POST	<code>/api/monitor/thread-diagnosis/analyze/async</code>	startAnalyzeJob	@RequestParam (defaultValue="live") String source	Result<Map<String, Object>>	异步启动分析任务, 返回 jobId
GET	<code>/api/monitor/thread-diagnosis/analyze/{jobId}</code>	getAnalyzeJob	@PathVariable("jobId") String jobId	Result<Map<String, Object>>	查询异步分析任务进度/结果

	alyze/async/{jobId}	ob			
POST	/api/monitor/thread-diagnosis/analyze/async/{jobId}/cancel	cancel AnalyzeJob	@PathVariable("jobId") String jobId	Result<Void>	取消异步分析任务
GET	/api/monitor/thread-diagnosis/export	export Data	@RequestParam(defaultValue="live") String source, @RequestParam(required=false) Long startTime, Long endTime, HttpServletResponse response	void	导出线程诊断采样数据文件（可指定时间范围）
GET	/api/monitor/thread-diagnosis/export-report	export Report	@RequestParam(defaultValue="live") String source, @RequestParam(defaultValue="html") String format, HttpServletResponse response	void	导出诊断分析报告（format 默认 html）
POST	/api/monitor/thread-diagnosis/import/probe	probel mport	@RequestParam("file") MultipartFile file	Result<Map<String, Object>>	上传线程诊断数据文件并预检（探测文件内容/时间范围）
POST	/api/monitor/thread-diagnosis/import/confirm	confirm mport	@RequestParam(required=false) Long startTime, Long endTime	Result<Map<String, Object>>	确认导入已预检的诊断数据（可选裁剪时间范围）
GET	/api/monitor/thread-diagnosis/imported	getIm ported Data		Result<Map<String, Object>>	获取当前已导入的诊断数据

ThreadDiagnosisStartRequest 字段说明

字段名	类型	中文说明
intervalSec	Integer	采样间隔（秒），必填，取值 1~3600

maxDataMb	Long	采样数据大小上限（MB），1~100，达到后自动丢弃最早数据，默认 10
-----------	------	--------------------------------------

44 ThreadPoolController

基础路径: /api/config/web/pools

方法	路径	函数	参数	返回类型	说明
GET	/api/config/web/pools	list	@RequestParam int page, int size, String name, Integer maxThreads, Integer priority	Result<PageResult<ThreadPool>>	分页查询线程池列表，支持名称/最大线程数/优先级过滤
GET	/api/config/web/pools/{id}	get	@PathVariable Long id	Result<ThreadPool>	按 ID 查询线程池详情，不存在则抛异常
GET	/api/config/web/pools/defaults	getDefaults		Result<Map<String, Object>>	获取线程池默认参数（按 CPU 核数计算 minThreads 等默认值）
POST	/api/config/web/pools	create	@RequestBody ThreadPool pool	Result<Void>	新增线程池（校验名称唯一），需重启生效，记审计
PUT	/api/config/web/pools/{idOrName}	update	@PathVariable String idOrName, @RequestBody ThreadPool pool	Result<Void>	按 ID 或名称更新线程池（校验名称唯一），需重启生效
DELETE	/api/config/web/pools/{idOrName}	delete	@PathVariable String idOrName	Result<Void>	按 ID 或名称删除线程池，需重启生效，记审计

ThreadPool 字段说明

字段名	类型	中文说明
name	String	线程池名称
maxThreads	Integer	最大线程数
minThreads	Integer	最小线程数

maxIdleTime	Integer	线程最大空闲时间（毫秒）
queueSize	Integer	等待队列大小
priority	Integer	线程优先级
threadStepReclaim	Integer	线程阶梯回收开关/值
threadStepSize	Integer	线程阶梯步长（每批创建线程数）
threadStepTime	Integer	线程阶梯时间间隔
connectionAutoReclaim	String	连接自动回收（"true"/"false"）
enabled	Integer	是否启用（0=禁用 1=启用）

45 ThreadTraceController

基础路径: /api/monitor/thread-trace

方法	路径	函数	参数	返回类型	说明
POST	/api/monitor/thread-trace/long/start	startLongTrace	@RequestBody LongTraceStartRequest request	Result<Void>	启动长请求线程追踪 (thresholdMs/intervalMs 均必填, 否则 400)
POST	/api/monitor/thread-trace/long/stop	stopLongTrace		Result<Void>	停止长请求线程追踪
GET	/api/monitor/thread-trace/long/snapshot	getLongTraceSnapshot		Result<Map<String, Object>>	获取长请求追踪快照 (执行时长超过阈值的线程数据)

LongTraceStartRequest 字段说明

字段名	类型	中文说明
thresholdMs	Long	阈值 (毫秒), 执行时长超过该值的请求判定为长请求, 必填
intervalMs	Long	采样频率 (毫秒), 必填

46 UpgradeHistoryController

基础路径: /api/system/upgrade-history

方法	路径	函数	参数	返回类型	说明
GET	/api/system/upgrade-history	list	PageQueryDTO query (page/size 分页参数)	Result<PageResult<UpgradeHistory>>	分页查询实例升级历史记录

UpgradeHistory 字段说明

字段名	类型	中文说明
version	String	升级到的版本号
upgradeTime	LocalDateTime	升级时间
status	Integer	状态: 0=失败 1=成功 2=进行中
note	String	备注

47 UserController

基础路径: /api/users

方法	路径	函数	参数	返回类型	说明
GET	/api/users	list	UserQueryDTO query (username/status/roleType/builtin等过滤+分页)	Result<PageResult<UserDTO>>	分页查询用户列表
GET	/api/users/{id}	get	@PathVariable Long id	Result<UserDTO>	按 ID 查询用户详情
POST	/api/users	create	@RequestBody UserDTO dto	Result<Void>	创建用户 (Create 校验分组, 含初始密码), 记审计
PUT	/api/users/{id}	update	@PathVariable Long id, @RequestBody UserDTO dto	Result<Void>	更新用户信息, 记审计
DELETE	/api/users/{id}	delete	@PathVariable Long id	Result<Void>	删除单个用户, 记审计
DELETE	/api/users/batch	deleteBatch	@RequestBody List<Long> ids	Result<Integer>	批量删除用户, 返回删除数量
POST	/api/users/{id}/status	toggleStatus	@PathVariable Long id, @RequestParam Integer status	Result<Void>	启用/禁用用户 (status: 0=禁用 1=启用)
POST	/api/users/{id}/unlock	unlock	@PathVariable Long id	Result<Void>	解锁用户 (清除登录锁定状态)
POST	/api/users/{id}/reset-password	resetPassword	@PathVariable Long id, @RequestBody Map<String, String> body (password)	Result<Void>	重置用户密码

UserDTO 字段说明

字段名	类型	中文说明
-----	----	------

id	Long	用户 ID
username	String	用户名（字母/数字/下划线/短横线）
email	String	邮箱
phone	String	电话
status	Integer	状态（0-2）
firstLogin	Integer	是否首次登录
classLevel	Integer	密级标识
roles	List	角色编码列表
password	String	创建用户时的初始密码（仅写入，不序列化到响应）
builtin	Integer	是否内置默认用户（admin/secure/audit），仅响应字段
otpEnabled	Integer	是否启用 OTP 双因子认证，仅响应字段
expireDate	LocalDateTime	账号过期时间，空表示永久有效
createdTime	LocalDateTime	创建时间
description	String	备注/说明

48 UserPermissionController

基础路径: /api/users

方法	路径	函数	参数	返回类型	说明
GET	/api/users/{userId}/permission	getByUserId	@PathVariable Long userId	Result<UserPermissionDTO>	获取指定用户的权限配置 (IP白名单/时间/密级/额外角色)
PUT	/api/users/{userId}/permission	saveOrUpdate	@PathVariable Long userId, @RequestBody UserPermissionDTO dto	Result<Void>	保存/更新用户权限配置, 记审计 (AUTH 操作)

UserPermissionDTO 字段说明

字段名	类型	中文说明
id	Long	权限配置记录 ID
userId	Long	用户 ID
allowedIps	String	允许访问的 IP 白名单, 多个用逗号分隔
accessStartTime	String	允许访问的开始时间 (HH:mm:ss)
accessEndTime	String	允许访问的结束时间 (HH:mm:ss)
accessWeekdays	String	允许访问的星期, 逗号分隔 1-7
dataClassLevel	Integer	数据密级 (0=公开 1=内部 2=秘密 3=机密 4=绝密)
extraRoleCodes	String	额外授予的角色编码, 逗号分隔
role	String	权限分配角色 (前端字段, 映射到 extraRoleCodes)
ipWhitelist	String	IP 白名单 (前端字段, 映射到 allowedIps)
startTime	String	开始访问时间 (前端字段, 映射到 accessStartTime)
endTime	String	结束访问时间 (前端字段, 映射到 accessEndTime)
classLevel	Integer	数据密级 (前端字段, 映射到 dataClassLevel)
timeLimitEnabled	Integer	是否启用时间限制

49 VirtualHostController

基础路径: /api/config/web/hosts

方法	路径	函数	参数	返回类型	说明
GET	/api/config/web/hosts/all	listAll		Result<List<VirtualHost>>	查询全部虚拟主机（不分页，供下拉选择）
GET	/api/config/web/hosts	list	@RequestParam int page, int size, String name, String alias, String dcBase	Result<PageResult<VirtualHost>>	分页查询虚拟主机列表
GET	/api/config/web/hosts/{id}	get	@PathVariable Long id	Result<VirtualHost>	按 ID 查询虚拟主机详情，不存在返回 404
POST	/api/config/web/hosts	create	@RequestBody VirtualHost host	Result<Void>	新增虚拟主机，需重启生效，记审计
PUT	/api/config/web/hosts/{idOrName}	update	@PathVariable String idOrName, @RequestBody VirtualHost host	Result<Void>	按 ID 或名称更新虚拟主机，需重启生效
DELETE	/api/config/web/hosts/{idOrName}	delete	@PathVariable String idOrName	Result<Void>	按 ID 或名称删除虚拟主机，需重启生效
PUT	/api/config/web/hosts/{idOrName}/default	setDefault	@PathVariable String idOrName	Result<Void>	将指定虚拟主机设为默认主机，需重启生效

VirtualHost 字段说明

字段名	类型	中文说明
name	String	虚拟主机名称
domain	String	域名

alias	String	别名
docBase	String	应用根目录 (docBase)
enabled	Integer	是否启用
isDefault	Integer	是否为默认主机
serviceName	String	所属服务 (Service) 名称
logFile	String	日志文件路径
accessIpWhitelist	String	访问 IP 白名单
accessIpBlacklist	String	访问 IP 黑名单
accessDomainWhitelist	String	访问域名白名单
accessDomainBlacklist	String	访问域名黑名单
accessTime	String	访问时间限制
sso	String	单点登录配置
ssoHttpOnly	Integer	SSO Cookie 是否 HttpOnly
extraProps	String	扩展属性 (自定义属性串)

50 附录： HTTP 方法说明

方法	说明
GET	获取资源
POST	创建资源
PUT	更新资源
DELETE	删除资源
PATCH	部分更新资源
REQUEST	通用请求

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年，前身为“金蝶中间件公司”，是金蝶集团旗下新一代软件基础云平台服务商，云计算国家标准制定企业，国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业，也是“两网一站四库十二金”国家重点工程的基础平台提供商，产品广泛应用于政府、军工、金融、能源等关键行业，累计服务客户总数超过10万家。



云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网：www.apusic.com

