



APUSIC
固若长城
睿比世界

API Documentation

Kingdee Apusic Application Server V10 Standard Edition

Copyright Notice

All intellectual property rights including software copyrights involved in this document have been legally registered and duly owned by Kingdee Apusic Cloud Computing Co., Ltd. Such rights are protected by Copyright Law of the People's Republic of China, Regulations on Protection of Computer Software, Regulations on Intellectual Property Protection, relevant international copyright conventions, laws, regulations and other intellectual property legislations. Unauthorized use is strictly prohibited.

Disclaimer

All copyright information contained herein is legally owned by Kingdee Apusic Cloud Computing Co., Ltd. and protected by applicable laws. The Company shall bear no liability for any information unrelated to Kingdee Apusic contained in this document. You may view, copy and print this document only within the scope permitted by Copyright Law of the People's Republic of China. No entity or individual may use, modify or republish any part or content of this document without written authorization from Kingdee Apusic Cloud Computing Co., Ltd. Any violation shall constitute infringement, and the Company reserves the right to pursue legal liabilities accordingly.

Updates to this document shall not be notified separately. You may consult the Company regarding issues related to this document. All rights not expressly granted by the Company are reserved.

Trademark Notice

 is a registered trademark filed with the National Intellectual Property Administration of PRC by Kingdee Apusic Cloud Computing Co., Ltd. Shenzhen. Exclusive rights to the registered trademark are legally owned by Kingdee Apusic and protected by law. Without written consent of Kingdee Apusic, no entity or individual shall use, copy, modify, disseminate, transcribe, or bundle and sell any part of this trademark with other products in any manner or for any reason. Kingdee Apusic shall pursue legal liabilities against any infringement of its trademark rights. All other trademarks or registered trademarks mentioned in this document belong to their respective owners.

Contents

- 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 Appendix: HTTP Method Description](#)

1 AlertConfigController

Base Path: `/api/alert-config`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/alert-config</code>	getConfig		Result<AlertConfigDTO>	Query alert configuration (switch/notification methods/threshold metrics)
PUT	<code>/api/alert-config</code>	saveConfig	@RequestBody AlertConfigDTO dto	Result<Void>	Save alert configuration; captures changes before saving for audit comparison
POST	<code>/api/alert-config/restore-defaults</code>	restoreDefaults		Result<AlertConfigDTO>	Restore alert configuration to defaults, with audit record

AlertConfigDTO Fields

Field	Type	Description
enabled	Integer	Alert master switch (0=off 1=on)
intervalMinutes	Integer	Alert interval (minutes, 1-527040)
emailEnabled	Integer	Email notification switch
emailRecipients	String	Recipient email addresses, multiple separated by semicolons
emailSmtpHost	String	SMTP server address
emailSmtpSsl	Integer	SMTP SSL switch
emailSmtpPort	Integer	SMTP port (1-65535)
emailSmtpUser	String	SMTP sender mailbox
emailSmtpPassword	String	SMTP password

snmpEnabled	Integer	SNMP notification switch
snmpVersion	String	SNMP version (v2c/v3)
snmpCommunity	String	SNMP community name
snmpAddress	String	SNMP target address
snmpSecurity	String	SNMPv3 security level
snmpUsername	String	SNMPv3 username
snmpAuthMethod	String	SNMPv3 authentication method
snmpAuthPassword	String	SNMPv3 authentication password
snmpCipher	String	SNMPv3 encryption algorithm
snmpPrivPassword	String	SNMPv3 encryption password
snmpSenderEngineId	String	SNMPv3 sender engine ID (read-only, for frontend display only)
platformEnabled	Integer	Alert platform notification switch
platformLevel	String	Alert level reported to the alert platform
platformUrl	String	Alert platform address (http/https)
metrics	List	Alert metric threshold configuration list (metricCode/threshold/consecutiveCount/unit)

2 AppClassLibController

Base Path: `/api/apps/{appId}/class-libs`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/apps/{appId}/class-libs</code>	list	<code>@PathVariable Long appId, @RequestParam int page, int size</code>	<code>Result<PageResult<Map<String, Object>>></code>	Paged query of an application's class library (JAR packages in the lib directory) list
POST	<code>/api/apps/{appId}/class-libs</code>	upload	<code>@PathVariable String appId, @RequestParam MultipartFile file</code>	<code>Result<Void></code>	Upload a JAR package to the application class library directory; the appId path segment accepts either a numeric ID or an application name (script replay)
DELETE	<code>/api/apps/{appId}/class-libs</code>	delete	<code>@PathVariable String appId, @RequestParam String jarName</code>	<code>Result<Void></code>	Delete the specified JAR package from the application class library

3 AppIncrementController

Base Path: `/api/apps/{appId}/increments`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/apps/{appId}/increments</code>	list	@PathVariable Long appId, @RequestParam int page, int size	Result<PageResult<AppIncrement>>	Paged query of an application's incremental update package list
POST	<code>/api/apps/{appId}/increments</code>	upload	@PathVariable String appId, @RequestParam MultipartFile file, String description	Result<Long>	Upload an incremental update package, returns the package ID; the appId path segment accepts either a numeric ID or an application name
DELETE	<code>/api/apps/{appId}/increments/{id}</code>	delete	@PathVariable String appId, Long id	Result<Void>	Delete the specified incremental update package
POST	<code>/api/apps/{appId}/increments/{id}/apply</code>	apply	@PathVariable String appId, Long id	Result<Void>	Apply the specified incremental update package to the application

4 ApplicationController

Base Path: `/api/apps`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/apps</code>	list	<code>AppQueryDTO query</code>	<code>Result<PageResult<ApplicationDTO>></code>	Paged query of a application list (filter by name/context path/virtual host/status)
GET	<code>/api/apps/{id}</code>	get	<code>@PathVariable Application app</code>	<code>Result<ApplicationDTO></code>	Get application details by ID
GET	<code>/api/apps/{id}/urls</code>	getAppUrls	<code>@PathVariable Application app</code>	<code>Result<List<AppURLDTO>></code>	Get the access URL list of the application
GET	<code>/api/apps/check-name</code>	checkAppName	<code>@RequestParam String appName, String virtualHost, Long excludeId</code>	<code>Result<Boolean></code>	Check whether the application name already exists (can exclude a specified ID)
GET	<code>/api/apps/check-context</code>	checkContextPath	<code>@RequestParam String contextPath, String virtualHost, Long excludeId</code>	<code>Result<Boolean></code>	Check whether the context path already exists (can exclude a specified ID)
GET	<code>/api/apps/check-context-backup</code>	checkContextPathForBackup	<code>@RequestParam String contextPath, String virtualHost, String appName</code>	<code>Result<Boolean></code>	Check context path conflicts during backup restore
POST	<code>/api/apps</code>	create	<code>@RequestBody ApplicationDTO dto</code>	<code>Result<Long></code>	Create an application, returns the new application ID
PUT	<code>/api/apps/{id}</code>	update	<code>@PathVariable Application app, @R</code>	<code>Result<Void></code>	Update application configuration (a

			requestBody ApplicationDTO dto		audit records before/after differences)
DELETE	/api/apps/{id}	delete	@PathVariable Application app, @RequestParam boolean force	Result<Void>	Undeploy/delete an application; force=true deletes forcibly (does not go to the recycle bin)
POST	/api/apps/{id}/start	start	@PathVariable Application app	Result<Void>	Start the application
POST	/api/apps/{id}/stop	stop	@PathVariable Application app	Result<Void>	Stop the application
POST	/api/apps/{id}/reload	reload	@PathVariable Application app	Result<Void>	Reload the application
POST	/api/apps/{id}/redeploy	redeploy	@PathVariable Application app, @RequestParam ApplicationDTO dto, MultipartFile file, String localWarPath, String localDirPath	Result<Void>	Redeploy the application (supports uploading a war file or specifying a local server path)
POST	/api/apps/{id}/deploy	deploy	@PathVariable Application app, @RequestParam MultipartFile file, String localWarPath, String localDirPath	Result<Void>	Deploy an application package (one of: war file / local war path / local directory path)
GET	/api/apps/war-browse	browseWarFiles	@RequestParam String path	Result<ServerFileListResponse>	Browse deployable war file directories on the server
GET	/api/apps/backups	getBackups		Result<List<Map<String, Object>>>	Get the backup list of all applications

POST	/api/apps/{id}/backups	create Backup	@PathVariable Application app	Result<Void>	Create a backup for the specified application
GET	/api/apps/backups/{backupName}/config	getBackupConfig	@PathVariable String backupName	Result<Map<String, Object>>	Get the application configuration in the backup package
POST	/api/apps/backups/{backupName}/restore	restore Backup	@PathVariable String backupName, @RequestBody ApplicationDTO dto	ResponseEntity<Result<Void>>	Restore an application from a backup; returns 409 on conflict
POST	/api/apps/backups/{backupName}/restore-force	restore BackupForce	@PathVariable String backupName, @RequestBody ApplicationDTO dto	Result<Void>	Forcibly restore an application from a backup (overwrites the existing application)
GET	/api/apps/backups/{backupName}/download	downloadBackup	@PathVariable String backupName	ResponseEntity<Resource>	Download the backup package file
DELETE	/api/apps/backups/{backupName}	delete Backup	@PathVariable String backupName	Result<Void>	Delete the backup package
GET	/api/apps/recycled	getRecycledApps	AppQueryDTO query	Result<List<RecycledAppDTO>>	Get the list of applications in the recycle bin
POST	/api/apps/recycle/{fileName}/restore	restore	@PathVariable String fileName, @RequestBody ApplicationDTO dto	Result<Void>	Restore an application from the recycle bin
GET	/api/apps/recycle/{fileName}/config	getRecycleArchiveConfig	@PathVariable String fileName	Result<Map<String, Object>>	Get the application configuration in the recycle bin archive
DELETE	/api/apps/recycle/{fileName}/permanent	permanently Delete	@PathVariable String fileName	Result<Void>	Permanently delete an application in

					the recycle bin (irrecoverable)
DELETE	/api/apps/external	delete External	@RequestBody ExternalAppInfo info	Result<Void>	Delete an externally deployed application (not deployed by the console)
POST	/api/apps/external/backups	create ExternalBackup	@RequestBody ExternalAppInfo info	Result<Map<String, Object>>	Create a backup for an externally deployed application
POST	/api/apps/external/redeploy	redeploy External	@RequestPart ExternalAppInfo info, MultipartFile file, String localWarPath	Result<Void>	Redeploy an external application
POST	/api/apps/external/reload	reload External	@RequestBody ExternalAppInfo info	Result<Void>	Reload an external application
POST	/api/apps/external/redeploy-with-props	redeploy ExternalWithProps	@RequestPart ExternalAppInfo info, ApplicationDTO dto, MultipartFile file, String localWarPath	Result<Void>	Redeploy an external application with configuration properties
GET	/api/apps/recycle/config	getRecycleConfig		Result<Map<String, Object>>	Get recycle bin configuration (retention path/retention days, etc.)
PUT	/api/apps/recycle/config	updateRecycleConfig	@RequestBody Map<String, Object> config	Result<?>	Save recycle bin configuration; returns 400 with field errors on validation failure
GET	/api/apps/backup/config	getBackupConfig		Result<Map<String, Object>>	Get backup configuration

PUT	/api/apps/backup/ config	updateBackupConfig	@RequestBody Map<String, Object> config	Result<?>	Save backup configuration; returns 400 with field errors on validation failure
-----	-----------------------------	--------------------	--	-----------	--

ApplicationDTO Fields

Field	Type	Description
id	Long	Application ID
appName	String	Application name
contextPath	String	Application context path
status	Integer	Application running status
classLoaderOrder	String	Class loading order
jspPrecompile	Boolean	Whether to precompile JSP
virtualHost	String	Owning virtual host
sessionManager	String	Session manager configuration
deployOrder	Integer	Deployment order
sharedLib	String	Shared library name
sharedLibId	Long	Shared library ID
sharedLibPriority	String	Shared library loading priority
deployTimeout	Integer	Deployment timeout (seconds)
classLevel	Integer	Classification level (secret/confidential/top secret)
description	String	Application description
deployTime	String	Deployment time
recycledTime	String	Recycle time
securityDomain	String	Security domain
deployProps	Map<String, String>	Deployment properties

managedByConsole	Boolean	Whether deployed by the console (false=externally deployed)
docBase	String	Application document root directory
deploySource	String	Deployment source: CONSOLE (deployed by console) / MANUAL_CONFIG (manual configuration) / AUTO_DEPLOY (auto-scan deployment)
originalVirtualHost	String	Original virtual host name before changing the virtual host (for migration)

AppQueryDTO Fields (inherits PageQueryDTO: page defaults to 1, size defaults to 20)

Field	Type	Description
appName	String	Application name (fuzzy query)
contextPath	String	Context path
virtualHost	String	Virtual host
status	Integer	Application status

5 AuditLogController

Base Path: `/api/audit/operation-logs`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/audit/operation-logs</code>	list	AuditLogQueryDTO query	Result<PageResult<AuditLogEntity>>	Paged query of operation audit logs (filter by operator/event type/operation target/result/time range)
GET	<code>/api/audit/operation-logs/{id}</code>	getById	@PathVariable Long id	Result<AuditLogEntity>	Get a single audit log by ID (if 18n key in description are translated according to the current Locale); returns 404 if not found
GET	<code>/api/audit/operation-logs/export</code>	export	AuditLogQueryDTO query, HttpServletResponse response	void (application/zip)	Export audit logs: export all records matching the query conditions, packaged as operation_audit.zip (containing operation_audit.csv with UTF-8 BOM)

AuditLogQueryDTO Fields (inherits PageQueryDTO: page defaults to 1, size defaults to 20)

Field	Type	Description
username	String	Operator username
operation	String	Event type (CREATE/EDIT/DELETE/START/STOP/AUTH/IMPORT/EXPORT/LICENSE/BACKUP)
target	String	Operation target
result	String	Operation result (SUCCESS/FAIL)

startTime	String	Start time
endTime	String	End time

6 AuthController

Base Path: `/api/auth`

Method	Path	Function	Parameters	Return Type	Description
POST	<code>/api/auth/login</code>	login	@RequestBody LoginRequest request, HttpServletRequest servletRequest	Result<LoginResponse>	User login: AES-decrypts username and password, sequentially validates captcha, account status (locked/disabled/expired), access time window and IP whitelist, OTP, and email code; issues a JWT on success and records the login log
POST	<code>/api/auth/cli-login</code>	cliLogin	@RequestBody LoginRequest request, HttpServletRequest servletRequest	Result<LoginResponse>	CLI login: same flow as login but does not validate the graphic captcha
POST	<code>/api/auth/logout</code>	logout	HttpServletRequest request	Result<Void>	Logout: adds the current token to the blacklist and records the logout log
GET	<code>/api/auth/captcha-status</code>	getCaptchaStatus		Result<Map<String, Object>>	Get the graphic captcha switch status (enabled)
GET	<code>/api/auth/otp-status</code>	getOtpStatus		Result<Map<String, Object>>	Get the global switch status of OTP two-factor authentication

GET	/api/auth/email-code/status	getEmailCodeStatus		Result<Map<String, Object>>	Get the switch status of email code login verification
POST	/api/auth/email-code/send	sendEmailCode	@RequestBody Map<String, String> body (username)	Result<Map<String, Object>>	Send login email verification code, with a 60-second resend limit
POST	/api/auth/otp/setup	setupOtp	@RequestBody Map<String, String> body	Result<Map<String, Object>>	Generate an OTP secret and QR code for the currently logged-in user (secret, otpAuthUri, qrCodeDataUri)
POST	/api/auth/otp/verify-setup	verifyOtpSetup	@RequestBody Map<String, String> body (secret, code)	Result<Void>	Verify the dynamic code and complete OTP binding
POST	/api/auth/otp/disable	disableOtp	@RequestBody Map<String, String> body (code)	Result<Void>	Disable OTP for the current user (requires entering the current dynamic code for confirmation)
PUT	/api/auth/profile	updateProfile	@RequestBody Map<String, String> body (phone/email/description)	Result<Void>	Update the personal information of the currently logged-in user
GET	/api/auth/me	me		Result<Map<String, Object>>	Get current logged-in user information (roles, classification level, contact info, session timeout, resource code list, etc.)

LoginRequest Fields

Field	Type	Description
username	String	Username (AES-encrypted in transit, decrypted on the server)
password	String	Password (AES-encrypted in transit, decrypted on the server)
captchaKey	String	Graphic captcha key
captchaCode	String	Graphic captcha answer
otpCode	String	OTP dynamic code
otpSecret	String	OTP secret (issued by the backend on first binding and echoed by the frontend)
emailCode	String	Email verification code

LoginResponse Fields

Field	Type	Description
token	String	JWT access token
tokenType	String	Token type, fixed as Bearer
username	String	Login username
roles	List	User role code list
firstLogin	boolean	Whether this is the first login (forced password change required)
passwordExpired	boolean	Password has expired and must be changed
requireOtp	boolean	Whether OTP verification is globally required
otpEnabled	boolean	Whether the current user has bound OTP
requireEmailCode	boolean	Whether this login requires an email verification code
emailCodeEnabled	boolean	Whether the email verification code feature is enabled
otpQrDataUri	String	QR code base64 dataUri for first-time OTP binding
otpSecret	String	OTP secret for first-time binding (echoed back on the second submission)
passwordExpireWarning	boolean	Password expiring soon warning (within the remaining 7 days)
resources	List	List of resource codes owned by the user (for frontend menu permissions)

7 BaselineCheckController

Base Path: `/api/baseline`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/baseline/rules</code>	listRules		Result<String>	List all baseline check rules (XML/JSON string)
GET	<code>/api/baseline/rules/{id}</code>	getRule	@PathVariable string id	Result<String>	Get a single rule's content by rule ID (id allows only letters, digits, underscores and hyphens, otherwise 400)
POST	<code>/api/baseline/rules</code>	saveRule	@RequestBody string ruleXml	Result<String>	Add a baseline check rule (raw rule XML as the input)
PUT	<code>/api/baseline/rules/{id}</code>	updateRule	@PathVariable string id, @RequestBody String ruleXml	Result<String>	Update the rule with the specified ID (saves the entire rule XML)
DELETE	<code>/api/baseline/rules/{id}</code>	deleteRule	@PathVariable string id	Result<String>	Delete the baseline rule with the specified ID
PUT	<code>/api/baseline/rules/{id}/enabled</code>	setRuleEnabled	@PathVariable string id, @RequestBody EnabledRequest request	Result<String>	Enable/disable the specified baseline rule
GET	<code>/api/baseline/script</code>	readScript	@RequestParam string fileName	Result<String>	Read the content of a check script file
POST	<code>/api/baseline/script</code>	saveScript	@RequestParam string fileName, @RequestBody string content	Result<String>	Save a check script file

POST	/api/baseline/checks	startCheck	@RequestBody startCheckRequest request	Result<String>	Manually trigger a baseline check, returns the task ID (empty ruleids means all rules)
GET	/api/baseline/checks/{taskId}	pollTask	@PathVariable String taskId	Result<String>	Poll the execution progress/result of a check task
GET	/api/baseline/reports	listReports	ReportQuery query (binds query parameters: name/overall/startTime/endTime/page/size)	Result<PageResult<BaselineReport>>	Paged query of baseline check report history
GET	/api/baseline/reports/{id}	getReport	@PathVariable Long id	Result<BaselineReport>	Get details of a single baseline report
DELETE	/api/baseline/reports/{id}	deleteReport	@PathVariable Long id	Result<Void>	Delete a single baseline report
DELETE	/api/baseline/reports	deleteReports	@RequestBody List<Long> ids	Result<Void>	Batch delete baseline reports
GET	/api/baseline/reports/{id}/download	downloadReport	@PathVariable Long id, @RequestParam(required=false) String format	ResponseEntity<ByteArrayResource>	Download a single report file (format optional, e.g. pdf/html, returned as an attachment stream)
POST	/api/baseline/reports/export	exportReports	@RequestBody List<Long> ids	ResponseEntity<ByteArrayResource>	Package multiple reports into a ZIP for export and download
GET	/api/baseline/settings	getSettings		Result<String>	Read all baseline check configuration items
PUT	/api/baseline/settings	saveSetting	@RequestBody configEntry entry	Result<String>	Modify a single baseline configuration item; cron/mast

					er switch changes publish events for hot-reloading scheduled tasks
GET	/api/baseline/report-config	getReportConfig		Result<String>	Read report-related configuration (same as getSettings)
PUT	/api/baseline/report-config	saveReportConfig	@RequestBody configEntry entry	Result<String>	Modify a report-related configuration item

StartCheckRequest Fields

Field	Type	Description
ruleIds	List	List of rule IDs to execute; empty means execute all rules

EnabledRequest Fields

Field	Type	Description
enabled	boolean	Rule enabled status (true enable / false disable)

ReportQuery Fields

Field	Type	Description
name	String	Report name fuzzy query
overall	String	Filter by report overall conclusion (pass/fail, etc.)
startTime	LocalDateTime	Check time from (ISO datetime format)
endTime	LocalDateTime	Check time to
page	int	Page number, default 1
size	int	Page size, default 10

ConfigEntry Fields

Field	Type	Description
-------	------	-------------

key	String	Configuration item key (e.g. baseline.schedule.cron, baseline.enabled)
value	String	Configuration item value

BaselineReport Fields (returned entity, inherits BaseEntity common fields id/createdTime/updatedTime/deleted)

Field	Type	Description
name	String	Report name
instance	String	Target instance identifier of the check
checkTime	LocalDateTime	Check execution time
overall	String	Overall conclusion (e.g. PASS/FAIL)
passCount	Integer	Number of passed items
minorCount	Integer	Number of minor issues
generalCount	Integer	Number of general issues
severeCount	Integer	Number of severe issues
ruleCount	Integer	Total number of rules involved in the check
reportFile	String	Report file storage path
format	String	Report file format
reportJson	String	Report detail JSON content
triggerType	String	Trigger type (MANUAL / SCHEDULED)
size	Long	Report file size (bytes); populated from the actual file at query time, not a database field

8 CaptchaController

Base Path: `/api/auth`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/auth/captcha</code>	getCaptcha		Result<Map<String, String>>	Generate a graphic captcha, returns the key and base64 image
GET	<code>/api/auth/captcha-test</code>	getCaptchaTest	@RequestParam String key	Result<Map<String, String>>	Test helper endpoint: get the captcha plaintext by key; only enabled in automated test environments (aas.security.captcha.test-enabled), Do not enable this in the production environment, otherwise the CAPTCHA mechanism will become ineffective

9 CertificateStoreController

Base Path: `/api/config/web/certs`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/web/certs</code>	list	@RequestParam int page, int size, String name, String storeType	Result<PageResult<CertificateStore>>	Paged query of certificate store list (password field returned masked with 15 asterisks); storeType supports only keystore/truststore
GET	<code>/api/config/web/certs/{id}</code>	get	@PathVariable Long id	Result<CertificateStore>	Get certificate store details by ID (password returned masked); returns 404 if not found
POST	<code>/api/config/web/certs</code>	create	@RequestBody CertificateStoreDTO dto	Result<Long>	Create a certificate store, returns the new record ID
PUT	<code>/api/config/web/certs/{idOrName}</code>	update	@PathVariable String idOrName, @RequestBody CertificateStoreDTO dto	Result<Void>	Update a certificate store by ID or name
DELETE	<code>/api/config/web/certs/{idOrName}</code>	delete	@PathVariable String idOrName	Result<Void>	Delete a certificate store by ID or name
GET	<code>/api/config/web/certs/{id}/entries</code>	listEntries	@PathVariable Long id, @RequestParam String password	Result<List<CertEntryDTO>>	List certificate entries in the store (optional password to unlock)

DELETE	/api/config/web/certs/{idOrName}/entries/{alias}	delete Entry	@PathVariable String idOrName, String alias, @RequestParam String password	Result<Void>	Delete the certificate entry with the specified alias from the store
POST	/api/config/web/certs/{idOrName}/import-entries	import Entries	@PathVariable String idOrName, @RequestParam MultipartFile file, String sourcePassword, String aliases, String targetPassword, String targetAlias, String entryPassword	Result<Void>	Import entries from an uploaded certificate store file (optional alias filtering and target alias/password)
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>	Import a certificate or private key by pasting PEM content (isPrivateKey marks a private key)
POST	/api/config/web/certs/{idOrName}/import	import Store	@PathVariable String idOrName, @RequestParam MultipartFile file	Result<Void>	Import an entire certificate store file (overwrite and replace)
GET	/api/config/web/certs/{id}/download	download	@PathVariable Long id	ResponseEntity<Resource>	Download the certificate store file; returns 404 if the file does not exist

CertificateStoreDTO Fields

Field	Type	Description
id	Long	Certificate store ID

name	String	Certificate store name (required on create, max 64 characters, letters/digits/underscores/hyphens/dots only)
storeType	String	Certificate store purpose (required on create, keystore or truststore)
type	String	Certificate store type (required on create, JKS or PKCS12)
password	String	Certificate store password (supports ENCRYPT* ciphertext, max 128/256 characters)
description	String	Description (max 255 characters)
enabled	Integer	Whether enabled

10 ClassAnalysisController

Base Path: `/api/class-analysis`

Method	Path	Function	Parameters	Return Type	Description
POST	<code>/api/class-analysis/upload</code>	upload	@RequestParam ("file") MultipartFile file	Result<Map<String, Object>>	Upload a file to be analyzed (jar/war, etc.), returns filePath/fileName/fileSize
POST	<code>/api/class-analysis/start</code>	start	@RequestBody classAnalysisStartRequest req	Result<Map<String, Object>>	Start a class analysis task, returns recordId (analysis record ID)
GET	<code>/api/class-analysis/records/{id}</code>	getRecord	@PathVariable Long id	Result<ClassAnalysisRecordDTO>	Query details of a single class analysis record (including the issue list)
GET	<code>/api/class-analysis/records</code>	listRecords	ClassAnalysisRecordQuery q (binds query parameters: startTime/endTime/target/appOrFileName/page/pageSize)	Result<PageResult<ClassAnalysisRecordDTO>>	Paged query of class analysis history records
DELETE	<code>/api/class-analysis/records/{id}</code>	delete Record	@PathVariable Long id	Result<Void>	Delete the specified class analysis record
GET	<code>/api/class-analysis/records/{id}/export</code>	export Report	@PathVariable Long id	ResponseEntity<byte[]>	Render the analysis record as a PDF report for download
GET	<code>/api/class-analysis/deployed-apps</code>	listDeployed Apps	@RequestParam(required=false) String keyword, @RequestParam(defaultValue=)	Result<PageResult<AppOptionDTO>>	Paged query of deployed application list (for selecting

			ItValue="1") <code>int</code> page, @RequestParam(defaultValue="20") <code>int</code> pageSize		the analysis target)
--	--	--	--	--	----------------------

ClassAnalysisStartRequest Fields

Field	Type	Description
name	String	Analysis task name
target	String	Analysis target type (e.g. uploaded file / deployed application)
appId	Long	Target application ID (fill in when analyzing a deployed application)
uploadFilePath	String	Storage path of the uploaded file (fill in when analyzing an uploaded file)
scope	String	Analysis scope
checkConflict	Boolean	Whether to perform class conflict check
checkRedundant	Boolean	Whether to perform redundant class check

11 ClassLoaderController

Base Path: `/api/classloader`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/classloader/tree</code>	getClassLoaderTree		Result<Map<String, Object>>	Get the class loader tree (data.classLoaders is a tree-structured list of ClassLoaderNode)
GET	<code>/api/classloader/detail/{id}</code>	getClassLoaderDetail	@PathVariable String id	Result<ClassLoaderDetail>	Get class loader details by ID (class name, instance, hash, class path, etc.)
GET	<code>/api/classloader/search</code>	searchClass	@RequestParam String className	Result<List<ClassSearchResult>>	Search by class name for how a class is loaded across class loaders (name/status/owning application/loader/location)

ClassLoaderNode Fields

Field	Type	Description
id	String	Node unique identifier
name	String	Class loader display name
className	String	Full name of the class loader implementation class
instance	String	Instance identifier
hash	String	Instance hash
app	String	Owning application
isAppLoader	boolean	Whether it is an application class loader

paths	List	Class search path list
children	List	Child class loader nodes

12 CracController

Base Path: `/api/config/jvm/crac`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/jvm/crac/env-check</code>	envCheck		Result<CracEnvCheckReport>	Synchronously run 7 CRaC environment compatibility checks (triggered when the CRaC switch is turned on)
GET	<code>/api/config/jvm/crac/status</code>	status		Result<CracStatusDTO>	Get the enabled/disabled linkage status of the "Make Image" button: image making is allowed only when enabled and saved + no pending restart + current JVM supports CRaC + the process is not in a restored state + the runtime directory matches the fixed value; reason values: notEnabled/pendingRestart/notCracJdk/restored/dirMismatch; also returns the crac.log path
POST	<code>/api/config/jvm/crac/images</code>	makeImage	@RequestBody cracMakeImageRequest request	Result<Void>	Make a CRaC image: after validation passes, spawns a detached helper to asynchronously complete image generation and ser

					ver startup (see logs/crac.log for results); returns business code 409 when image data already exists in the checkpoint directory, retry with force=true after frontend confirmation
--	--	--	--	--	--

CracMakeImageRequest Fields

Field	Type	Description
name	String	Image name (required, letters/digits/underscores/hyphens/dots only, max 64)
desc	String	Image description (optional, max 200)
force	boolean	Whether to forcibly overwrite when image data already exists in the checkpoint directory

13 DashboardController

Base Path: `/api/dashboard`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/dashboard/overview</code>	getOverview		Result<DashboardOverviewDTO>	Get dashboard home overview data (system statistics, application status, quick info, etc.)

14 DatasourceController

Base Path: `/api/resources/datasources`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/resources/datasources</code>	list	<code>DatasourceQueryDTO query</code>	<code>Result<PageResult<Datasource>></code>	Paged query of datasource list
GET	<code>/api/resources/datasources/{id}</code>	get	<code>@PathVariable Long id</code>	<code>Result<Datasource></code>	Get datasource details by ID
POST	<code>/api/resources/datasources (JSON)</code>	create ByJson	<code>@Valid @RequestBody Datasource ds</code>	<code>Result<Datasource></code>	Create a datasource via JSON (using an already uploaded driver path)
POST	<code>/api/resources/datasources (multipart)</code>	create	<code>@Valid @RequestPart Datasource ds, MultipartFile driverFile, boolean Overwrite</code>	<code>Result<Datasource></code>	Create a datasource via multipart, can upload a driver package at the same time (overwrite controls overwriting a driver with the same name)
PUT	<code>/api/resources/datasources/{idOrName}</code>	update	<code>@PathVariable String idOrName, @Valid @RequestBody Datasource ds</code>	<code>Result<Void></code>	Update a datasource by ID or name
DELETE	<code>/api/resources/datasources/{idOrName}</code>	delete	<code>@PathVariable String idOrName</code>	<code>Result<Void></code>	Delete a datasource by ID or name
POST	<code>/api/resources/datasources/{idOrName}/status</code>	toggle Status	<code>@PathVariable String idOrName, @RequestParam Integer status</code>	<code>Result<Void></code>	Enable/disable a datasource (status allows only 0 or 1)
POST	<code>/api/resources/datasources/ping</code>	pingBy Params	<code>@RequestBody Datasource ds</code>	<code>Result<Map<String, Object>></code>	Test datasource connectivity with the

					given parameters (url is required)
POST	/api/resources/datasources/{idOrName}/ping	ping	@PathVariable String idOrName	Result<Map<String, Object>>	Test connectivity of an existing datasource
POST	/api/resources/datasources/{idOrName}/driver	uploadDriver	@PathVariable String idOrName, @RequestParam MultipartFile driverFile, boolean overwrite	Result<Void>	Upload a JDBC driver package for the datasource
GET	/api/resources/datasources/driver-exists	checkDriverExists	@RequestParam String filename	Result<Map<String, Object>>	Check whether a driver package with the same name already exists in the driver directory (pre-upload check)
POST	/api/resources/datasources/check-driver-path	checkDriverPath	@RequestBody Map<String, String> body (driverPath)	Result<Map<String, Object>>	Check whether the specified driver path exists

Datasource Fields (create/update request body, only main business fields listed)

Field	Type	Description
name	String	Datasource name (max 128 characters)
jndiName	String	JNDI name
dbType	String	Database type
driverClass	String	JDBC driver class name (max 128 characters)
url	String	Connection URL (max 500 characters)
username	String	Connection username
password	String	Connection password
maxConnections	Integer	Maximum connections (0-10000)
initialConnections	Integer	Initial connections (0-10000)

maxWaitMs	Integer	Maximum wait time for acquiring a connection (milliseconds, -1 means wait indefinitely)
validationQuery	String	Connection validation SQL
classLevel	Integer	Classification level
status	Integer	Status
driverJarPath	String	Driver JAR path
driverPath	String	Driver path
slowSqlMs	Integer	Slow SQL threshold (milliseconds)
leakDetectionThresholdMs	Integer	Connection leak detection threshold (milliseconds)
suspectTimeoutMs	Integer	Suspected leak determination timeout (milliseconds)
testOnBorrow	Integer	Whether to validate when borrowing a connection (0/1)
testOnReturn	Integer	Whether to validate when returning a connection (0/1)
testWhileIdle	Integer	Whether to validate while idle (0/1)
testOnConnect	Integer	Whether to validate when establishing a connection (0/1)
resourceType	String	Resource type
host	String	Host name
port	Integer	Port number (1-65535)
databaseName	String	Database name
customProperties	String	Custom properties
idleTimeoutMs	Integer	Idle timeout (milliseconds)
idleCheckPeriodMs	Integer	Idle check period (milliseconds)
maxCheckConnections	Integer	Maximum checked connections
minIdle	Integer	Minimum idle connections
maxIdle	Integer	Maximum idle connections
validationIntervalMs	Integer	Connection validation interval (milliseconds)
validationTimeoutSeconds	Integer	Validation timeout (seconds, 0-3600)

maxLifetimeMs	Integer	Maximum connection lifetime (milliseconds)
leakRecover	Integer	Whether to reclaim leaked connections (0/1)
connectionAutoReclaim	Integer	Automatic connection reclaim (0/1)
removeAbandonedOnMaintenance	Integer	Remove abandoned connections during maintenance (0/1)
logAbandoned	Integer	Log abandoned connections (0/1)
abandonWhenPercentageFull	Integer	Abandon percentage threshold (0-100)
autoCommit	Integer	Auto commit (0/1)
sqlTraceEnabled	Integer	SQL trace switch (0/1)
failedSqlAlertEnabled	Integer	Failed SQL alert switch (0/1)
haProbeEnabled	Integer	HA probe switch (0/1)

15 HttpChannelController

Base Path: `/api/config/web/channels`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/web/channels</code>	list	@RequestParam int page, int size, String name, String protocol, Integer enabled	Result<PageResult<HttpChannel>>	Paged query of HTTP channel list
GET	<code>/api/config/web/channels/{id}</code>	get	@PathVariable Long id	Result<HttpChannel>	Get HTTP channel details by ID; returns 404 if not found
POST	<code>/api/config/web/channels</code>	create	@RequestBody HttpChannel channel	Result<Void>	Add an HTTP channel (requires restart to take effect)
PUT	<code>/api/config/web/channels/{idOrName}</code>	update	@PathVariable String idOrName, @RequestBody HttpChannel channel	Result<Void>	Update an HTTP channel by ID or name (requires restart to take effect)
DELETE	<code>/api/config/web/channels/{idOrName}</code>	delete	@PathVariable String idOrName	Result<Void>	Delete an HTTP channel by ID or name (requires restart to take effect)
GET	<code>/api/config/web/channels/cipher-suites</code>	getSupportedCipherSuites		Result<List<String>>	Get the list of strong cipher suites supported by the JVM (weak CBC/RSAsuites filtered out)

HttpChannel Fields (@RequestBody, many fields, summarized by group)

Field	Type	Description
-------	------	-------------

name	String	Channel name
protocol	String	Protocol (HTTP/HTTPS)
host	String	Listening host
port	Integer	Listening port
executor	String	Associated thread pool executor name
enabled	Integer	Whether enabled
sslEnabled	Integer	Whether SSL is enabled
redirectPort	Integer	Redirect port
acceptCount	Integer	Waiting queue length
connectionTimeout	Integer	Connection timeout (milliseconds)
proxyName / proxyPort	String / Integer	Reverse proxy host name/port
receiveBufferSize / sendBufferSize	Integer	Receive/send buffer size
processorCache	Integer	Processor cache size
maxConnections	Integer	Maximum connections
tcpNoDelay	Integer	Whether to enable TCP_NODELAY
maxKeepAliveRequests / keepAliveTimeout	Integer	Maximum requests per keep-alive connection/timeout
useIPVHosts	Integer	Whether to match virtual hosts by IP
longThreadEnabled and other longThread*	-	Long thread detection switch and threshold/period/log directory
sdosEnabled and other sdos*	-	Slow denial of service (SDoS) protection switch, mode, thresholds, blacklist TTL, etc.
generalProperties	String	General custom properties
gracefulStopAwaitMillis	Integer	Graceful stop wait time (milliseconds)
uriEncoding / responseCharacterEncoding	String	URI/response character encoding
enableLookups	Integer	Whether DNS reverse lookup is allowed

compression and other compression*	-	Response compression switch, compressible MIME types, minimum compression size
httpTrace	Integer	HTTP access trace switch
maxPostSize / maxParameterCount / maxHeaderCount / maxHttpRequestSize	Integer	Limits on POST size/parameter count/header count and size
poweredBy	Integer	Whether to show the Server response header
disableUploadTimeout / connectionUploadTimeout	Integer	Upload timeout configuration
useBodyEncodingForURI / asyncTimeout	-	URI encoding source / async request timeout
responseHeaders	String	Custom response headers
gmEnabled	Integer	Chinese cryptography (GM) SSL switch
clientAuth	String	SSL client authentication mode
keyPassword / certificateFormat / tlsProtocols / sslProtocol	String	SSL key password/certificate format/TLS protocol versions
certificateAlias	String	Certificate alias
keystorePath / keystorePassword	String	Keystore path/password
truststorePath / truststorePassword / truststoreType	String	Truststore path/password/type
sslCipherSuites	String	SSL cipher suite list
certificateType	String	Certificate type (JSON property name type)
certificateCustom / keystoreName / keystoreEntry / truststoreName / truststoreEntry	-	Certificate store reference mode and entries
http2Enabled and other http2*	-	HTTP/2 switch and compression, concurrent streams, header/frame size, read/write timeouts, etc.
http2Properties	String	HTTP/2 custom properties

16 InspectionController

Base Path: `/api/inspection/reports`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/inspection/reports</code>	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>>	Paged query of inspection report list (time format yyyy-MM-dd HH:mm:ss)
GET	<code>/api/inspection/reports/has-running</code>	hasRunning		Result<Boolean>	Query whether there is an inspection task currently running
GET	<code>/api/inspection/reports/{id}</code>	get	@PathVariable Long id	Result<InspectionReportDTO>	Get details of a single inspection report (including summary, instance list, baseline items, monitor data); returns 404 if not found
GET	<code>/api/inspection/reports/{id}/status</code>	status	@PathVariable Long id	Result<String>	Query the execution status of an inspection report (RUNNING/SUCCESS/FAILED, etc.)
GET	<code>/api/inspection/reports/{id}/summary</code>	summary	@PathVariable Long id	Result<InspectionReportDTO.InspectionSummary>	Get the report summary (counts of CPU/memory/hea

				tionSummaryDTO >	p/disk exceeding 80%)
POST	/api/inspection/reports	trigger	@RequestBody @Valid InspectionTriggerDTO dto	Result<InspectionReportDTO>	Trigger a new inspection, returns the report (dto.timeoutSeconds is the collection timeout in seconds)
POST	/api/inspection/reports/{id}/rerun	rerun	@PathVariable Long id	Result<InspectionReportDTO>	Re-run an inspection based on a historical report
GET	/api/inspection/reports/{id}/download	download	@PathVariable Long id	ResponseEntity<Resource>	Download the inspection report ZIP package (including all snapshot files)
DELETE	/api/inspection/reports/{id}	delete	@PathVariable Long id	Result<Void>	Delete the specified inspection report
GET	/api/inspection/reports/{id}/snapshots/{type}	snapshotText	@PathVariable Long id, @PathVariable String type	Result<String>	View the text content of a specified type of system snapshot (jstack/lsof/netstat/top) in the report online
GET	/api/inspection/reports/{id}/snapshots/{type}/download	downloadSnapshot	@PathVariable Long id, @PathVariable String type	ResponseEntity<Resource>	Download the specified type of snapshot XML file

InspectionTriggerDTO Fields

Field	Type	Description
timeoutSeconds	Integer	Inspection collection timeout (seconds), required, 1~172800

InspectionReportDTO Fields (response DTO)

Field	Type	Description
-------	------	-------------

id	Long	Report ID
name	String	Report name
timeoutSeconds	Integer	Collection timeout in seconds
status	String	Execution status
startedAt	LocalDateTime	Start time
finishedAt	LocalDateTime	Finish time
createdTime	LocalDateTime	Creation time
summary	InspectionSummaryDTO	Report summary (returned only by the detail endpoint)
instances	List	Inspection instance list (returned only by the detail endpoint)
baselineItems	List	Security baseline check item list
monitorData	Map<String, Object>	Monitor data (applications/HTTP channels/datasources/thread pools/OS/JVM)
snapshotStatuses	Map<String, SnapshotStatus>	System snapshot collection status (key: jstack/lsof/netstat/top), including status (OK/NOT_INSTALLED/FALLBACK/SKIPPED/FAILED/TIMEOUT) and message

17 InvokeChainTraceController

Base Path: `/api/monitor/invoke-chain`

Method	Path	Function	Parameters	Return Type	Description
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>>	Paged query of invocation chain trace history records (time format yyyy-MM-dd HH:mm:ss)
DELETE	<code>/api/monitor/invoke-chain/records/{id}</code>	deleteRecord	@PathVariable Long id	Result<Void>	Delete the specified trace record
GET	<code>/api/monitor/invoke-chain/settings</code>	getSettings		Result<Map<String, Object>>	Get invocation chain analysis configuration (including apmHome, etc.)
PUT	<code>/api/monitor/invoke-chain/settings/apm-home</code>	saveApmHome	@RequestBody Map<String, String> body (takes the apmHome key)	Result<Map<String, Object>>	Save the APM agent installation directory (apmHome); requires restart to take effect
POST	<code>/api/monitor/invoke-chain/trace/start</code>	startTrace	@RequestBody @Valid InvokeChainTraceStartRequest request	Result<Map<String, Object>>	Start invocation chain tracing (parameters such as cost threshold/package filtering)

POST	/api/monitor/invoke-chain/trace/stop	stopTrace		Result<Map<String, Object>>	Stop the current invocation chain trace
GET	/api/monitor/invoke-chain/trace/status	status		Result<Map<String, Object>>	Query the trace running status
GET	/api/monitor/invoke-chain/records/{id}/xml	downloadXml	@PathVariable Long id	ResponseEntity<Resource>	Download the trace result XML file
GET	/api/monitor/invoke-chain/records/{id}/flame-graph	flameGraph	@PathVariable Long id	ResponseEntity<Resource>	Generate and download the flame graph HTML file of the record

InvokeChainTraceStartRequest Fields

Field	Type	Description
cost	Long	Minimum request cost (milliseconds), required, positive integer
methodMinCost	Long	Minimum method cost (milliseconds), required, positive integer and must be less than cost
excPkgs	String	Excluded package names, multiple separated by commas (no spaces allowed)
incPkgs	String	Included package names, multiple separated by commas (no spaces allowed)
appContext	String	Target application context path (e.g. /myapp), optional; empty means no application filtering
deadline	LocalDateTime	Trace deadline (yyyy-MM-dd HH:mm:ss), optional; empty means no automatic stop

InvokeChainTrace Fields (returned entity, inherits BaseEntity common fields id/createdTime/updatedTime/deleted)

Field	Type	Description
startTime	LocalDateTime	Trace start time
endTime	LocalDateTime	Trace end time

status	String	Status: SUCCESS / FAILED
params	String	Trace parameter snapshot JSON (cost/methodMinCost/incPkgs/excPkgs/appContext/deadline)
fileName	String	Trace result XML file name (used for download)
filePath	String	Absolute path of the trace result XML file (path on the machine where the agent resides)
dataSize	Long	Trace result XML data size (bytes)
errorMsg	String	Failure/exception information

18 JmxConfigController

Base Path: `/api/config/jmx`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/jmx</code>	getConfig		Result<JmxConfig>	Get JMX configuration (keyStorePassword/keyPassword returned masked with asterisks)
GET	<code>/api/config/jmx/cipher-suites</code>	getSupportedCipherSuites		Result<List<String>>	Get the list of strong cipher suites supported by the JVM (weak suites filtered out)
PUT	<code>/api/config/jmx</code>	saveConfig	@RequestBody JmxConfig config	Result<Void>	Save JMX configuration (string fields trimmed, requires restart to take effect)

JmxConfig Fields

Field	Type	Description
enabled	Integer	Whether to enable JMX
host	String	JMX listening host
port	Integer	JMX listening port
authenticate	Integer	Whether to enable authentication
passwordFile	String	Password file path
accessFile	String	Access control file path
sslEnabled	Integer	Whether to enable SSL
clientAuth	Integer	Whether client certificate authentication is required

keyStoreFile	String	Keystore file path
keyPassword	String	Key password
keyStorePassword	String	Keystore password
keyStoreType	String	Keystore type
trustStoreFile	String	Truststore file path
trustStorePassword	String	Truststore password
trustStoreType	String	Truststore type
enabledCipherSuites	String	Enabled cipher suite list
certificateCustom	Integer	Whether to use a custom certificate store reference
keyStoreName	String	Keystore name (certificate store reference)
trustStoreName	String	Truststore name (certificate store reference)
keyStoreEntry	String	Keystore certificate entry alias

19 JndiResourceController

Base Path: `/api/resources/jndi`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/resources/jndi</code>	list	<code>PageQueryDTO query</code>	<code>Result<PageResult<JndiResource>></code>	Paged query of JNDI resource list
GET	<code>/api/resources/jndi/{id}</code>	get	<code>@PathVariable Long id</code>	<code>Result<JndiResource></code>	Get JNDI resource details by ID; returns 404 if not found
POST	<code>/api/resources/jndi</code>	create	<code>@RequestBody JndiResourceDTO dto</code>	<code>Result<Void></code>	Create a JNDI resource
PUT	<code>/api/resources/jndi/{id}</code>	update	<code>@PathVariable Long id, @RequestBody JndiResourceDTO dto</code>	<code>Result<Void></code>	Update a JNDI resource
DELETE	<code>/api/resources/jndi/{id}</code>	delete	<code>@PathVariable Long id</code>	<code>Result<Void></code>	Delete a JNDI resource
POST	<code>/api/resources/jndi/{id}/status</code>	toggle Status	<code>@PathVariable Long id, @RequestParam Integer status</code>	<code>Result<Void></code>	Enable/disable a JNDI resource (status can only be 0 or 1)

JndiResourceDTO Fields

Field	Type	Description
<code>jndiName</code>	String	JNDI name (required on create, max 256 characters)
<code>resourceType</code>	String	Resource type (required on create)
<code>factoryClass</code>	String	Fully qualified factory class name (required on create, max 512 characters)
<code>properties</code>	String	Custom properties (JSON object string, max 65536 characters)
<code>description</code>	String	Resource description (max 256 characters)

20 JvmConfigController

Base Path: `/api/config/jvm`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/jvm</code>	getConfig		Result<JvmConfig>	Get JVM configuration
PUT	<code>/api/config/jvm</code>	saveConfig	@RequestBody JvmConfig config	Result<?>	Save JVM configuration: validates heap size (Xms<=Xmx, positive integers), metaspace (-1 or positive, initial <=max), JVM option format (must start with -D/-X/-XX:/-javaagent: and not be duplicated), native library path existence, and debug option format; when CRaC is enabled, validates that fdPolicy is required and is an existing file. On 400, data carries field names such as cracFdPolicy for frontend display. Requires restart to take effect

JvmConfig Fields

Field	Type	Description
jdkPath	String	JDK path
javaVersion	String	Java version

xms	String	Minimum heap (MB, positive integer)
xmx	String	Maximum heap (MB, not less than xms)
xmn	String	Young generation size
permSize / maxPermSize	String	Initial/maximum permanent generation size (legacy configuration compatibility)
metaspaceSize	String	Initial metaspace size (-1 or positive integer)
maxMetaspaceSize	String	Maximum metaspace size (-1 or positive integer, not less than the initial value)
gcType	String	Garbage collector type
gcLogPath	String	GC log path
debugEnabled	Integer	Whether to enable remote debugging
debugOptions	String	Debug options (only tokens starting with -D/-X/-XX:/-agentlib: are allowed)
nativeLibraryPath	String	Native library path (multiple paths separated by the path separator, all must exist)
jvmOptions	String	Additional JVM options (one per line, starting with -D/-X/-XX:/-javaagent: and not duplicated)
oomAction	String	OOM handling policy
cracEnabled	Integer	Whether to enable the CRaC feature
cracCheckpointDir	String	CRaC checkpoint directory (fixed by the server, request value ignored)
cracFdPolicy	String	CRaC file descriptor policy file path (required and must exist when CRaC is enabled)
cracTraceStartupTime	Integer	CRaC startup time tracing
cracPrintResources	Integer	CRaC print resource information
cracCollectFdStacktraces	Integer	CRaC collect fd stack traces
cracFdPolicyDefault	String	Default fdPolicy path (resolved at runtime, not persisted, for frontend default restoration)

21 LicenseController

Base Path: `/api/license`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/license/info</code>	info		<code>Result<Map<String, Object>></code>	Get the currently effective license information (including machine identifier iden, whether licensed, feature codes, licensed object, validity dates, license type, whether perpetual, remaining days, product name/version, whether cloud environment)
GET	<code>/api/license/days-remaining</code>	daysRemaining		<code>Result<Map<String, Object>></code>	Get the remaining days of the license
POST	<code>/api/license/update</code>	update	<code>@RequestParam multipartFile file, String content, String fileName</code>	<code>Result<License></code>	Update the license (upload a .lic/.xml file or directly submit license text content, with file size/file name/content length validation)

22 LogController

Base Path: `/api/logs`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/logs/files</code>	getLogFiles		Result<List<Map<String, Object>>>	Get the log file list (scans the system log directory and access log directory, including file name, path, size, last modified time, type)
GET	<code>/api/logs/content</code>	getLogContent	@RequestParam String fileName, long offset, String direction, int pageSize, String keyword, String logLevel, String startTime, String endTime	Result<LogContentResponse>	Streaming bidirectional cursor-paged reading of log content (supports forward/backward/last/tail/raw-* directions and keyword, level, time range filtering)
GET	<code>/api/logs/raw</code>	getLogRawContent	@RequestParam String fileName	Result<LogRawContentResponse>	Read the complete raw content of a log file (falls back to GBK when UTF-8 fails, returns content and file size)
GET	<code>/api/logs/config</code>	getLogConfig		Result<Map<String, Object>>	Get system log and access log configuration (two parts: systemLog/accessLog)
PUT	<code>/api/logs/config</code>	updateLogConfig	@RequestBody Map<String, Object> config	Result<?>	Update log configuration (after validation, saves the system/access log c

					onfiguration and pushes it to the AAS instance application; requires restart to take effect)
POST	/api/logs/rotate	rotateLog	@RequestParam string fileName	Result<Void>	Rotate the specified log file
DELETE	/api/logs/clear	clearLogs	@RequestParam string fileName	Result<Void>	Clear the content of the specified log file
POST	/api/logs/export	exportLog	@RequestBody Map<String, String> body	Result<Map<String, Object>>	Generate a log export download link (returns downloadUrl and the zip file name)
GET	/api/logs/download	downloadLog	@RequestParam string fileName, string logType, String keyword, String logLevel, String startTime, String endTime, HttpServletResponse response	void	Download the log zip file (only matching log entries are exported when filter conditions are present)
POST	/api/logs/config/sync	syncLogConfig		Result<Map<String, Object>>	Synchronize log configuration from the AAS instance back to the console and return the latest configuration

23 LoginLogController

Base Path: `/api/audit/login-logs`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/audit/login-logs</code>	list	LoginLogQueryDTO query	Result<PageResult<LoginLog>>	Paged query of login audit logs
GET	<code>/api/audit/login-logs/export</code>	export	LoginLogQueryDTO query, HttpServletResponse response	void	Export login audit logs as a CSV wrapped in a zip (including event time, account, event type, IP, description, result)

24 MetricsController

Base Path: `/metrics`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/metrics</code>	metrics	HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	Prometheus metrics endpoint (text/plain), 404 when the switch is off; supports Basic Auth / Access Token + IP whitelist
GET	<code>/metrics/collectors</code>	listCollectors	HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	List all collector names and category mappings (JSON, requires the Restful switch enabled and authentication passed)
GET	<code>/metrics/data</code>	collectAll	HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	Get all monitor data (merging all collectors), returns RestfulMetricsResponse
GET	<code>/metrics/data/{category}</code>	collectCategory	@PathVariable String category, HttpServletRequest servletRequest, @RequestHeader String authHeader	ResponseEntity<?>	Get monitor data of a single category; unknown category returns 404
GET	<code>/metrics/data/{category}/{name}</code>	collectOne	@PathVariable String category, @PathVariable String name, HttpServletRequest servletRequest, @Request	ResponseEntity<?>	Get monitor data of a single collector (category/name); unknown collector returns 404

			Header string authenticationHeader		
--	--	--	------------------------------------	--	--

25 MonitorConfigController

Base Path: `/api/monitor/config`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/monitor/config</code>	getConfig		Result<MonitorConfig>	Get monitor service configuration (SNMP v3 passwords returned masked with asterisks; Prometheus credentials decrypted and returned with password masked)
PUT	<code>/api/monitor/config</code>	saveConfig	@RequestBody MonitorConfigDTO	Result<Void>	Save monitor service/alert/SNMP configuration (excluding interface service fields; stops the collector when Full GC monitoring is turned off)
POST	<code>/api/monitor/config/restore-defaults</code>	restoreDefaults		Result<MonitorConfig>	Restore monitor service default configuration
POST	<code>/api/monitor/config/prometheus-auth-token/refresh</code>	refreshPrometheusAuthToken		Result<Map<String, String>>	Regenerate the Prometheus Authorization Bearer token (plaintext returned only in this response, stored encrypted)
PUT	<code>/api/monitor/config/interface-service</code>	updateInterfaceService	@RequestBody InterfaceServiceConfigDTO	Result<MonitorConfig>	Update only the Prometheus/Restful interface service configuration (switches, auth method,

					Basic Auth credentials, Token, whitelist; validates complexity when the Basic Auth password is non-empty)
POST	/api/monitor/config/interface-service/restore-defaults	restore InterfaceServiceDefaults		Result<MonitorConfig>	Restore only the interface service related fields to defaults

MonitorConfigDTO Fields (only business fields listed; repeated validation notes for 0/1 switches omitted)

Field	Type	Description
monitorEnabled	Integer	Monitor service master switch (0/1)
mbeanMonitorEnabled	Integer	MBean monitoring switch (0/1)
collectIntervalSeconds	Integer	Collection interval (seconds, 10~3600)
jvmMonitorEnabled	Integer	JVM monitoring switch (0/1)
threadPoolMonitorEnabled	Integer	Thread pool monitoring switch (0/1)
dsMonitorEnabled	Integer	JDBC connection pool monitoring switch (0/1)
httpMonitorEnabled	Integer	HTTP channel monitoring switch (0/1)
sessionMonitorEnabled	Integer	Session monitoring switch (0/1)
systemResourceMonitorEnabled	Integer	System resource monitoring switch (0/1)
fullGcMonitorEnabled	Integer	Full GC monitoring switch (0/1)
cpuMonitorEnabled	Integer	CPU monitoring switch (0/1)
memoryMonitorEnabled	Integer	Memory monitoring switch (0/1)
maxDataPoints	Integer	Maximum number of data points
replaySampleInterval	Integer	Replay sampling interval (seconds, 1~86400)
alertEnabled	Integer	Alert switch (0/1)
alertIntervalMinutes	Integer	Alert interval (minutes, 1~1440)

alertCpuThreshold	Integer	CPU alert threshold (0~100)
alertMemoryThreshold	Integer	Memory alert threshold (0~100)
alertDiskThreshold	Integer	Disk alert threshold (0~100)
httpThreadAlertEnabled	Integer	HTTP thread alert switch (0/1)
httpThreadThreshold	Integer	HTTP thread alert threshold
cpuAlertEnabled	Integer	CPU alert notification switch (0/1)
virtualMemoryAlertEnabled	Integer	Virtual memory alert switch (0/1)
virtualMemoryThreshold	Integer	Virtual memory alert threshold (0~100)
physicalMemoryAlertEnabled	Integer	Physical memory alert switch (0/1)
physicalMemoryThreshold	Integer	Physical memory alert threshold (0~100)
fullGcAlertEnabled	Integer	Full GC alert switch (0/1)
fullGcDurationSeconds	Integer	Full GC duration threshold (seconds)
fullGcFrequencyLimit	Integer	Full GC frequency limit (times)
jdbcAlertEnabled	Integer	JDBC alert switch (0/1)
jdbcErrorThreshold	Integer	JDBC error threshold
alertSendMessageEnabled	Integer	Alert send message switch (0/1)
alertGenerateSnapshotEnabled	Integer	Alert generate snapshot switch (0/1)
snapshotThreadEnabled	Integer	Thread snapshot switch (0/1)
snapshotHeapEnabled	Integer	Heap snapshot switch (0/1)
snapshotProcessEnabled	Integer	Process snapshot switch (0/1)
sendEmailEnabled	Integer	Email notification switch (0/1)
sendSnmpEnabled	Integer	SNMP notification switch (0/1)
sendPlatformEnabled	Integer	Platform notification switch (0/1)
snmpEnabled	Integer	SNMP switch (0/1)
snmpHost	String	SNMP Agent bind address (IPv4/IPv6 literal, max 128)
snmpPort	Integer	SNMP port (1~65535)

snmpCommunity	String	SNMP community name
snmpProtocol	String	SNMP protocol
snmpVersion	String	SNMP version
snmpAddress	String	NMS-side address (IP/port format)
snmpSecurity	String	SNMP v3 security level (none/auth-no-priv/auth-priv)
snmpUsername	String	SNMP v3 username
snmpAuthMethod	String	SNMP v3 authentication method (MD5/SHA)
snmpAuthPassword	String	SNMP v3 authentication password (plaintext input, encrypted on the server before storage)
snmpCipher	String	SNMP v3 encryption algorithm (DES/3DES/AES128/AES192/AES256)
snmpPrivPassword	String	SNMP v3 encryption password
prometheusEnabled	Integer	Prometheus interface switch (0/1)
restfulEnabled	Integer	Restful monitor data interface switch (0/1)
prometheusAuthMethod	String	Authentication method (none/basic/authorization)
prometheusAuthUsername	String	Basic Auth username (plaintext input, encrypted on the server)
prometheusAuthPassword	String	Basic Auth password (plaintext input, encrypted on the server)
prometheusAuthToken	String	Authorization Bearer value (plaintext input, encrypted on the server)
prometheusWhitelist	String	Whitelist raw text (multi-line/comma-separated IPs or CIDRs, including IPv6)

InterfaceServiceConfigDTO Fields

Field	Type	Description
prometheusEnabled	Integer	Prometheus switch (0=off 1=on)
restfulEnabled	Integer	Restful monitor data interface switch (0=off 1=on)
prometheusAuthMethod	String	Authentication method (none/basic/authorization)

prometheusAuthUsername	String	Basic Auth username (plaintext input, encrypted on the server; letters, digits and ._- only)
prometheusAuthPassword	String	Basic Auth password (plaintext input, must not contain <>; validates password policy complexity when non-empty)
prometheusAuthToken	String	Authorization Bearer value (plaintext input, must not contain <>)
prometheusWhitelist	String	Whitelist raw text (multi-line/comma-separated IPs or CIDRs, including IPv6, max 512)

26 MonitorDataController

Base Path: `/api/monitor/data`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/monitor/data/system</code>	getSystemResource		Result<Map<String, Object>>	Get real-time system resource monitor data (CPU/memory/disk, etc.; returns null when the module switch is off)
GET	<code>/api/monitor/data/system-resource/history</code>	getSystemResourceHistory	@RequestParam <code>LocalDateTime</code> startTime, <code>LocalDateTime</code> endTime	Result<SystemResourceTrendDTO>	Query system resource historical trend data (default last 1 hour, max 30 days)
GET	<code>/api/monitor/data/jvm</code>	getJvmMonitor		Result<Map<String, Object>>	Get JVM memory/garbage collection real-time monitor data
GET	<code>/api/monitor/data/jvm/history</code>	getJvmHistory	@RequestParam <code>LocalDateTime</code> startTime, <code>LocalDateTime</code> endTime	Result<JvmHistoryDTO>	Query JVM monitor history data (memory, GC trends, etc.)
GET	<code>/api/monitor/data/threads</code>	getThreadInfo		Result<Map<String, Object>>	Get JVM thread real-time monitor information
GET	<code>/api/monitor/data/fullgc/records</code>	fullGcRecords	@RequestParam <code>LocalDateTime</code> startTime, <code>LocalDateTime</code> endTime, <code>Integer</code> current, <code>Integer</code> size	Result<Page<FullGcRecordDTO>>	Paged query of Full GC records (default last 1 day, max 30 days)

GET	/api/monitor/data/fullgc/trend	fullGcTrend	@RequestParam LocalDateTime startTime, LocalDateTime endTime	Result<FullGcTrendDTO>	Query Full GC trend statistics
GET	/api/monitor/data/fullgc/status	fullGcStatus		Result<Map<String, Object>>	Get Full GC collector status (supported: whether supported, listening; whether listening)
POST	/api/monitor/data/fullgc/switch/on	fullGcSwitchOn		Result<Void>	Start Full GC collection listening
POST	/api/monitor/data/fullgc/switch/off	fullGcSwitchOff		Result<Void>	Stop Full GC collection listening
GET	/api/monitor/data/fullgc/export	exportFullGcRecords	@RequestParam LocalDateTime startTime, LocalDateTime endTime, HttpServletResponse response	void	Export Full GC records as a CSV file
GET	/api/monitor/data/jndi	getJndiTree		Result<Map<String, Object>>	Get the JNDI tree structure
GET	/api/monitor/data/sql	getSqlTrace		Result<Map<String, Object>>	Get SQL trace monitor data
GET	/api/monitor/data/perf	getPerfData		Result<Map<String, Object>>	Get performance collection data
GET	/api/monitor/data/datasource	getDataSourceMonitor		Result<Map<String, Object>>	Get datasource connection pool real-time monitor data
GET	/api/monitor/data/datasource-replay	getDataSourceReplay		Result<Map<String, Object>>	Get datasource monitor data (replay scenario, not restricted by switches)

POST	/api/monitor/data/datasource/test	testDataSource	@RequestParam String jndiName	Result<Boolean>	Test connectivity of the specified JNDI datasource
GET	/api/monitor/data/datasource/history	getDataSourceHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime, String jndiName	Result<DataSourceTrendDTO>	Query datasource historical trend (filterable by JNDI name)
GET	/api/monitor/data/http	getHttpChannelMonitor		Result<Map<String, Object>>	Get HTTP channel real-time monitor data
GET	/api/monitor/data/http-replay	getHttpChannelReplay		Result<Map<String, Object>>	Get HTTP channel monitor data (replay scenario)
GET	/api/monitor/data/threadpool	getThreadPoolMonitor		Result<Map<String, Object>>	Get thread pool real-time monitor data
GET	/api/monitor/data/threadpool-replay	getThreadPoolReplay		Result<Map<String, Object>>	Get thread pool monitor data (replay scenario)
GET	/api/monitor/data/threadpool/history	getThreadPoolHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime, String poolName	Result<ThreadPoolHistoryDTO>	Query thread pool history data (filterable by thread pool name)
GET	/api/monitor/data/threadpool/{name}/detail	getThreadPoolDetail	@PathVariable String name	Result<Map<String, Object>>	Get detailed information of the specified thread pool
GET	/api/monitor/data/app-sessions	getAppSessionMonitor		Result<Map<String, Object>>	Get application session real-time monitor data

GET	/api/monitor/data/app-sessions-replay	getAppSessionReplay		Result<Map<String, Object>>	Get application session monitor data (replay scenario)
GET	/api/monitor/data/app-sessions/history	getAppSessionHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime, String contextPath	Result<AppSessionTrendDTO>	Query application session historical trend (filterable by application context path)
GET	/api/monitor/data/http/trend	getHttpChannelTrend	@RequestParam String port	Result<Map<String, Object>>	Get HTTP channel request trend data (filterable by port)
GET	/api/monitor/data/http/history	getHttpChannelHistory	@RequestParam LocalDateTime startTime, LocalDateTime endTime, String port	Result<HttpChannelHistoryDTO>	Query HTTP channel history data (filterable by port)
GET	/api/monitor/data/http-connections	getHttpConnections		Result<Map<String, Object>>	Get HTTP connection count data (shared by dashboard/monitor)

27 NotificationController

Base Path: `/api/notifications`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/notifications</code>	list	@RequestParam Integer page, Integer size, Integer readStatus	Result<PageResult<Notification>>	Paged query of the current user's notification list (filterable by read status)
GET	<code>/api/notifications/unread-count</code>	unreadCount		Result<Map<String, Object>>	Get the current user's unread notification count
PUT	<code>/api/notifications/{id}/read</code>	markRead	@PathVariable Long id	Result<Void>	Mark the specified notification as read
PUT	<code>/api/notifications/read-all</code>	markAllRead		Result<Void>	Mark all notifications of the current user as read
POST	<code>/api/notifications</code>	create	@RequestBody Map<String, String> body (title/type/content/severity/targetUser)	Result<Notification>	Create a notification (a target user can be specified; a audit recorded)
DELETE	<code>/api/notifications</code>	deleteBatch	@RequestBody List<Long> ids	Result<Integer>	Batch delete notifications, returns the number deleted

28 PasswordController

Base Path: `/api/auth`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/auth/password-policy</code>	getPasswordPolicy		Result<PasswordPolicyDTO>	Get the current system password policy
POST	<code>/api/auth/change-password</code>	changePassword	@RequestBody changePasswordRequest request	Result<Void>	Change the current logged-in user's password (AES-encrypted in transit; validates the old password, password policy, and historical password duplication; on success, all sessions of the user are forcibly invalidated)

ChangePasswordRequest Fields

Field	Type	Description
oldPassword	String	Old password (AES-encrypted ciphertext)
newPassword	String	New password (AES-encrypted ciphertext)
confirmPassword	String	Confirm new password (AES-encrypted ciphertext, must match newPassword)

29 PatchController

Base Path: `/api/system/patches`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/system/patches</code>	list	<code>PageQueryDTO query</code>	<code>Result<PageResult<Patch>></code>	Paged query of patch list
POST	<code>/api/system/patches/upload</code>	upload	<code>@RequestParam String name, String version, String platform, String description, MultipartFile file</code>	<code>Result<Void></code>	Upload a patch package (multipart form)
POST	<code>/api/system/patches/{id}/apply</code>	apply	<code>@PathVariable Long id, String upgradeNote, Integer forceUpgrade</code>	<code>Result<Void></code>	Apply (upgrade) the specified patch, with optional upgrade note and force upgrade flag
POST	<code>/api/system/patches/{id}/rollback</code>	rollback	<code>@PathVariable Long id, String rollbackNote</code>	<code>Result<Void></code>	Roll back the specified patch
DELETE	<code>/api/system/patches/{id}</code>	delete	<code>@PathVariable Long id</code>	<code>Result<Void></code>	Delete the specified patch
GET	<code>/api/system/patches/{id}/progress</code>	getProgress	<code>@PathVariable Long id</code>	<code>Result<Map<String, Object>></code>	Get patch apply/rollback progress

Patch Fields

Field	Type	Description
id	Long	Patch ID
patchName	String	Patch name
version	String	Patch version
platform	String	Applicable platform version

description	String	Patch description
filePath	String	Patch file path
status	Integer	Patch status (0=not applied, 1=applied, 2=failed, 3=rolled back)
progress	Integer	Installation progress percentage
installTime	LocalDateTime	Installation time
rolledBackTime	LocalDateTime	Rollback time
upgradeNote	String	Upgrade note
rollbackNote	String	Rollback note
forceUpgrade	Integer	Force upgrade flag
progressDetail	String	Progress details (JSON format)

30 ReplayController

Base Path: `/api/monitor/replay`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/monitor/replay/system</code>	system	@RequestParam string start, string end	Result<Map<String, Object>>	Query system-dimension (host/system resource) historical sampling point replay data
GET	<code>/api/monitor/replay/jvm</code>	jvm	@RequestParam string start, string end	Result<Map<String, Object>>	Query JVM-dimension historical sampling point replay data
GET	<code>/api/monitor/replay/datasource</code>	datasource	@RequestParam string start, string end, string target	Result<Map<String, Object>>	Query historical sampling point replay data of the specified datasource (target)
GET	<code>/api/monitor/replay/app-session</code>	appSession	@RequestParam string start, string end, string target	Result<Map<String, Object>>	Query session count historical sampling point replay data of the specified application (target)
GET	<code>/api/monitor/replay/http-channel</code>	httpChannel	@RequestParam string start, string end, string target	Result<Map<String, Object>>	Query historical sampling point replay data of the specified HTTP channel (target)
GET	<code>/api/monitor/replay/thread-pool</code>	threadPool	@RequestParam string start, string end, string target	Result<Map<String, Object>>	Query historical sampling point replay data of the specified thread pool (target)

31 ResourceController

Base Path: `/api/resources`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/resources/permission-tree</code>	permissionTree		Result<List<PermissionTreeNode>>	Get the permission tree (three-level structure: module -> page -> operation), used for role permission configuration

PermissionTreeNode Fields

Field	Type	Description
code	String	Node code (menu code or API permission code, e.g. app_list:create)
name	String	Node display name
nameKey	String	i18n name key (menu:xxx or permission:xxx)
resourceId	Long	Associated resource ID (only for menu nodes)
children	List	Child node list (pages/operations)

32 RoleController

Base Path: `/api/roles`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/roles</code>	list	<code>PageQueryDTO query, @RequestParam String roleCode, Integer builtin</code>	<code>Result<PageResult<RoleDTO>></code>	Paged query of role list (filterable by role code and whether built-in)
GET	<code>/api/roles/all</code>	listAll		<code>Result<List<RoleDTO>></code>	Query all roles (no pagination)
GET	<code>/api/roles/{id}</code>	get	<code>@PathVariable Long id</code>	<code>Result<RoleDTO></code>	Get role details by ID
POST	<code>/api/roles</code>	create	<code>@RequestBody RoleDTO dto</code>	<code>Result<Void></code>	Create a role (Create validation group, empty string fields normalized to null)
PUT	<code>/api/roles/{id}</code>	update	<code>@PathVariable Long id, @RequestBody RoleDTO dto</code>	<code>Result<Void></code>	Update the specified role
DELETE	<code>/api/roles/{id}</code>	delete	<code>@PathVariable Long id</code>	<code>Result<Void></code>	Delete the specified role
DELETE	<code>/api/roles/batch</code>	deleteBatch	<code>@RequestBody List<Long> ids</code>	<code>Result<Integer></code>	Batch delete roles, returns the number deleted

RoleDTO Fields

Field	Type	Description
id	Long	Role ID
roleCode	String	Role code (required on create, letters/digits/underscores/hyphens only, max 64)
roleName	String	Role display name (max 64)

description	String	Role description (max 255)
builtin	Integer	Whether it is a built-in role (1=built-in, cannot be modified or deleted)
classLevel	Integer	Role classification level
resourceIds	List	Associated resource ID list
resourceCodes	List	Associated resource code list (for display)
userNames	List	Associated username list
userCount	Integer	Number of associated users (for list display)
defaultRole	Integer	Whether it is the default role (only one role can be the default)
permissions	List	Permission code list (including menu and operation permissions)

33 ScriptRecordingController

Base Path: `/api/script-recording`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/script-recording</code>	list	@RequestParam s tring name, String status, LocalDate Time startTime, Lo calDateTime endTi me, long page, lo ng size	Result<PageResult <ScriptRecording VO>>	Paged query of sc ript recording task list, supports nam e/status/time rang e filtering
GET	<code>/api/script-recording/active</code>	active	Authentication aut hentication	Result<Boolean>	Whether the curre nt user has a reco rding task in progr ess (used for logo ut confirmation, re turns only their o wn status)
POST	<code>/api/script-recording</code>	create	@RequestBody sc riptRecordingCrea teDTO dto, Authen tication authentica tion	Result<ScriptReco rdingVO>	Create a script rec ording task (with t he current user as creator), audit rec orded
POST	<code>/api/script-recording/{id}/stop</code>	stop	@PathVariable Lon g id, Authentica tion authentication	Result<ScriptReco rdingStopResult>	Stop recording an d generate scripts (aascli/postman/c url)
POST	<code>/api/script-recording/{id}/stop-empty</code>	stopE mpty	@PathVariable Lon g id, Authentica tion authentication	Result<Void>	Stop recording wit hout generating s cripts (empty task discarded), audit r ecorded
DELETE	<code>/api/script-recording/{id}</code>	delete	@PathVariable Lon g id, Authentica tion authentication	Result<Void>	Delete a recording task (creator only)

GET	/api/script-recording/{id}/download	download	@PathVariable Long id, @RequestParam List<String> formats	ResponseEntity<byte[]>	Download the script ZIP package generated by recording (packages all three formats aasc i/postman/curl by default)
-----	-------------------------------------	----------	---	------------------------	--

ScriptRecordingCreateDTO Fields

Field	Type	Description
name	String	Script name (letters/digits/underscores/hyphens/dots, 1-255 characters)
timeoutMinutes	Integer	Timeout (minutes), 0=unlimited
description	String	Description, optional, max 1000 characters

ScriptRecordingVO Fields

Field	Type	Description
id	Long	Recording task ID
name	String	Script name
status	String	Task status (recording/stopped, etc.)
description	String	Description
timeoutMinutes	Integer	Timeout (minutes)
creator	String	Creator username
startTime	LocalDateTime	Recording start time
endTime	LocalDateTime	Recording end time
opCount	Integer	Number of operations recorded
elapsedSeconds	long	Elapsed recording duration (seconds), calculated in real time while recording

34 SecurityConfigController

Base Path: `/api/security/config`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/security/config</code>	getConfigs		<code>Result<Map<String, String>></code>	Get all security configuration items (smtp_password uniformly returned as an encrypted string)
PUT	<code>/api/security/config</code>	updateConfigs	<code>@RequestBody Map<String, String> configs, @RequestHeader String X-Current-User-Email</code>	<code>Result<Map<String, Object>></code>	Batch update security configuration (when backup_enabled changes from off to on, a backup is executed immediately; when smtp_verify_enabled is turned on, a test email is automatically sent and the result returned)
POST	<code>/api/security/config/test-mail</code>	testMail	<code>@RequestBody Map<String, String> body (to)</code>	<code>Result<Map<String, Object>></code>	Send an SMTP test email to the specified recipient
GET	<code>/api/security/config/smtp</code>	getSmtpEnabled		<code>Result<Map<String, Object>></code>	Get the email verification switch status
PUT	<code>/api/security/config/smtp</code>	updateSmtpEnabled	<code>@RequestBody Map<String, Object> body (smtpEnabled)</code>	<code>Result<Void></code>	Update the email verification switch
GET	<code>/api/security/config/auth-policy</code>	getAuthPolicy		<code>Result<Map<String, Object>></code>	Get authentication policy configuration

					n
PUT	/api/security/config/auth-policy	updateAuth Policy	@RequestBody Map<String, Object> policy	Result<Void>	Save authentication policy configuration
POST	/api/security/config/backup-now	backupNow		Result<Map<String, Object>>	Immediately execute a manual backup of business configuration
GET	/api/security/config/backup-list	listBackups		Result<List<Map<String, Object>>>	Query the business configuration backup file list
POST	/api/security/config/backup-restore	restoreBackup	@RequestBody Map<String, String> body (fileName)	Result<Map<String, Object>>	Restore a business configuration backup by file name
DELETE	/api/security/config/backup/{fileName}	deleteBackup	@PathVariable String fileName	Result<Void>	Delete the specified business configuration backup file
POST	/api/security/config/restart	restart		Result<Map<String, Object>>	Restart the AAS server (returns RESTARTING status)

SecurityConfig Fields

Field	Type	Description
id	Long	Configuration ID
configKey	String	Configuration item key
configValue	String	Configuration item value
description	String	Configuration description

35 SecurityRealmController

Base Path: `/api/security/realms`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/security/realms</code>	list	<code>SecurityRealmQueryDTO</code> query	<code>Result<PageResult<SecurityRealmDTO>></code>	Paged query of security realm list
GET	<code>/api/security/realms/all</code>	listAll		<code>Result<List<SecurityRealmDTO>></code>	Query all security realms (no pagination)
GET	<code>/api/security/realms/{id}</code>	get	<code>@PathVariable</code> Long id	<code>Result<SecurityRealmDTO></code>	Get security realm details by ID (returns 404 if not found)
POST	<code>/api/security/realms</code>	create	<code>@RequestBody</code> <code>SecurityRealmDTO</code> dto	<code>Result<Void></code>	Create a security realm (requires restart to take effect)
PUT	<code>/api/security/realms/{id}</code>	update	<code>@PathVariable</code> Long id, <code>@RequestBody</code> <code>SecurityRealmDTO</code> dto	<code>Result<Void></code>	Update a security realm (requires restart to take effect)
DELETE	<code>/api/security/realms/{id}</code>	delete	<code>@PathVariable</code> Long id	<code>Result<Void></code>	Delete a security realm (requires restart to take effect)
DELETE	<code>/api/security/realms/batch</code>	delete Batch	<code>@RequestBody</code> <code>List<Long></code> ids	<code>Result<Integer></code>	Batch delete security realms, returns the number deleted
GET	<code>/api/security/realms/user-database-resources</code>	getUserDatabaseResources		<code>Result<List<String>></code>	Get the list of user database resource names available in the AAS instance

SecurityRealmDTO Fields

Field	Type	Description
id	Long	Security realm ID
name	String	Security realm name
type	String	Type (DataSourceRealm/UserDatabaseRealm/JAASRealm/JNDIRealm/Cust
config	String	Configuration content (JSON or custom format)
xmlConfig	String	XML configuration output (for server.xml)
description	String	Description
passwordHashAlgorithm	String	Password hash algorithm (none/MD5/SHA-1/SHA-256/SHA-512)
boundApps	List	Bound application name list
lockEnabled	Boolean	Whether to enable the account lockout mechanism
maxFailedAttempts	Integer	Allowed number of wrong password attempts
lockTimeout	Integer	Lockout timeout (seconds)
cacheSize	Integer	Cache size
cacheRemovalWarningTime	Integer	Cache removal warning time (seconds)
userTable	String	User table name (DataSource Realm)
usernameField	String	Username field
passwordField	String	Password field
roleTable	String	Role table name
roleField	String	Role field
dataSourceJndi	String	Datasource JNDI name (DataSource Realm)
resourceName	String	Resource name (UserDatabase Realm)
jaasLoginModules	List	JAAS LoginModule configuration list (including loginModuleClass/controlFlag/options)
userPrincipalClass	String	User Principal implementation class (JAAS Realm)
rolePrincipalClass	String	Role Principal implementation class

contextFactory	String	Context factory class name (JNDI Realm)
jndiConnectionUrl	String	Connection URL
alternateUrl	String	Alternate connection URL
connectionUsername	String	Connection username
connectionPassword	String	Connection password
userSearchBase	String	User search base
userSearchFilter	String	User search filter
userSubtree	Boolean	User subtree search
userPattern	String	User DN pattern
userSearchAsUser	Boolean	Search for users as the user
userPasswordAttr	String	User password attribute name
userRoleName	String	Role attribute name in the user entry
userRoleAttribute	String	User role attribute (used for roleSearch substitution)
roleSearchBase	String	Role search base
roleSearchFilter	String	Role search filter
roleNameAttribute	String	Role name attribute
roleSubtree	Boolean	Role subtree search
roleNested	Boolean	Nested role search
roleSearchAsUser	Boolean	Search for roles as the user
commonRole	String	Common role
connectionTimeout	String	Connection timeout (milliseconds)
readTimeout	String	Read timeout (milliseconds)
connectionPoolSize	Integer	Connection pool size (must be greater than 0)

36 SessionManagerController

Base Path: `/api/config/web/sessions`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/web/sessions/all</code>	listAll		Result<List<SessionManager>>	Query all session managers (no pagination)
GET	<code>/api/config/web/sessions/redis-plugin-status</code>	redisPluginStatus		Result<Map<String, Boolean>>	Get Redis session plugin status (loaded/jarsInstalled/commonLoaderConfigured)
GET	<code>/api/config/web/sessions</code>	list	@RequestParam int page, int size, String name, String type	Result<PageResult<SessionManagerListItem>>	Paged query of session manager list (filterable by name and type)
GET	<code>/api/config/web/sessions/{id}</code>	get	@PathVariable Long id	Result<SessionManagerListItem>	Get session manager details by ID (returns 404 if not found)
POST	<code>/api/config/web/sessions</code>	create	@RequestBody SessionManager sm	Result<Void>	Create a session manager (field-level validation of name/type/Redis/path, etc.; error on duplicate name)
PUT	<code>/api/config/web/sessions/{id}</code>	update	@PathVariable Long id, @RequestBody SessionManager sm	Result<Void>	Update a session manager (same validation as create, excluding itself from duplicate check)
DELETE	<code>/api/config/web/sessions/{id}</code>	delete	@PathVariable Long id	Result<Void>	Delete a session manager (logical)

					delete, name preserved for audit)
--	--	--	--	--	-----------------------------------

SessionManager Fields (excluding BaseEntity common fields)

Field	Type	Description
name	String	Session manager name (required, max 128)
type	String	Type (memory/file/redis/database)
maxSessions	Integer	Maximum active sessions (-1 means unlimited, otherwise >=1)
persistPath	String	Session persistence file path (memory type, relative paths only)
saveOnRestart	Integer	Whether to save sessions on restart
maxIdleBackup	Integer	Idle backup threshold (seconds, -1 or >=0)
minIdleSwap	Integer	Minimum idle swap threshold (seconds)
maxIdleSwap	Integer	Maximum idle swap threshold (seconds)
persistenceDir	String	Persistence directory (file type, relative paths only)
redisMode	String	Redis mode (standalone/sentinel)
redisNodes	String	Redis nodes (host:port, multiple nodes separated by commas)
redisMasterName	String	Sentinel mode master node name
redisPassword	String	Redis password (max 256)
redisDatabase	Integer	Redis database index (0~15)
redisConnectTimeout	Integer	Redis connection timeout (milliseconds)
redisMaxConn	Integer	Redis maximum connections (>=1)
redisMinIdle	Integer	Redis minimum idle connections (>=0)
redisMaxIdle	Integer	Redis maximum idle connections
redisMaxWait	Integer	Redis maximum wait time (milliseconds)
sessionPersistPolicies	String	Session persistence policies (SAVE_ON_CHANGE/ALWAYS_SAVE_AFTER_REQUEST,

		comma-combined)
sessionAttributeNameFilter	String	Session attribute name filter (regular expression)
sessionAttributeValueClassNameFilter	String	Session attribute value class name filter (regular expression)
warnOnFilterFailure	Integer	Whether to alert on filter failure
notifyAttributeOnUnchanged	Integer	Whether to notify when an attribute is unchanged
notifyBindingOnUnchanged	Integer	Whether to notify when a binding is unchanged
dataSourceName	String	Datasource name (required for database type)
localDataSource	Integer	Whether to use a local datasource

37 SharedLibController

Base Path: `/api/resources/sharedlibs`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/resources/sharedlibs</code>	list	SharedLibQueryDTO query	Result<PageResult<SharedLib>>	Paged query of shared library list
GET	<code>/api/resources/sharedlibs/{id}</code>	get	@PathVariable Long id	Result<SharedLib>	Get shared library details by ID
POST	<code>/api/resources/sharedlibs</code>	create	@RequestBody SharedLib lib	Result<Long>	Create a shared library, returns the new record ID
PUT	<code>/api/resources/sharedlibs/{id}</code>	update	@PathVariable Long id, @RequestBody SharedLib lib	Result<Void>	Update a shared library
DELETE	<code>/api/resources/sharedlibs/{id}</code>	delete	@PathVariable Long id	Result<Void>	Delete a shared library
POST	<code>/api/resources/sharedlibs/{id}/upload-jar</code>	uploadJar	@PathVariable Long id, @RequestParam MultipartFile file	Result<Void>	Upload a JAR file to the shared library
GET	<code>/api/resources/sharedlibs/{id}/jars</code>	listJarFiles	@PathVariable Long id	Result<List<Map<String, Object>>>	List the uploaded JAR files in the shared library
GET	<code>/api/resources/sharedlibs/{id}/jars/{fileName}/download</code>	downloadJar	@PathVariable Long id, @PathVariable String fileName, HttpServletResponse response	void	Download the specified JAR file in the shared library
DELETE	<code>/api/resources/sharedlibs/{id}/jars/{fileName}</code>	deleteJarFile	@PathVariable Long id, @PathVariable String fileName	Result<Void>	Delete the specified JAR file in the shared library

POST	/api/resources/sharedlibs/{id}/jars/delete-batch	deleteJarFiles	@PathVariable Long id, @RequestBody List<String> fileNamees	Result<Void>	Batch delete JAR files in the shared library
POST	/api/resources/sharedlibs/{id}/apps	bindApp	@PathVariable Long id, @RequestParam Long appld	Result<Void>	Bind the shared library to the specified application
DELETE	/api/resources/sharedlibs/{id}/apps	unbindApp	@PathVariable Long id, @RequestParam Long appld	Result<Void>	Unbind the shared library from the application
GET	/api/resources/sharedlibs/{id}/apps	listBoundApps	@PathVariable Long id	Result<List<Map<String, Object>>>	Query the list of applications bound to the shared library
PUT	/api/resources/sharedlibs/{id}/priority	updatePriority	@PathVariable Long id, @RequestParam Integer priority	Result<Void>	Update the shared library loading priority (0~9999, smaller value means higher priority)

SharedLib Fields (excluding BaseEntity common fields)

Field	Type	Description
name	String	Shared library name (required, max 128)
path	String	Shared library path (max 512)
version	String	Version number (max 64)
description	String	Description
classLevel	Integer	Class level (>=0)
status	Integer	Status (0/1)
loadStrategy	String	Class loading strategy (application-first/classes-first/shared-first)
sortOrder	Integer	Sort order (>=0)
priority	Integer	Loading priority (0~9999, smaller value means higher priority, default 0)
fileCount	Integer	File count (non-persistent field, for list display)

refCount	Integer	Reference count (non-persistent field, for list display)
----------	---------	--

38 SnapshotConfigController

Base Path: `/api/monitor/snapshot-config`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/monitor/snapshot-config</code>	getConfig		Result<SnapshotConfig>	Query automatic snapshot configuration (currently effective configuration)
PUT	<code>/api/monitor/snapshot-config</code>	saveConfig	@RequestBody snapshotConfigDTO	Result<Void>	Save automatic snapshot configuration (including content and trigger rules), audit log recorded
POST	<code>/api/monitor/snapshot-config/restore-defaults</code>	restoreDefaults		Result<SnapshotConfig>	Restore automatic snapshot configuration to defaults

SnapshotConfigDTO Fields

Field	Type	Description
id	Long	Configuration record ID
autoEnabled	Integer	Automatic snapshot switch (0=off 1=on)
contentAccessLog	Integer	Snapshot content: access log switch
contentGcLog	Integer	Snapshot content: GC log switch
contentSystemConfig	Integer	Snapshot content: system configuration switch
contentSystemLog	Integer	Snapshot content: system log switch
contentHostMonitor	Integer	Snapshot content: host monitor data switch
contentInstanceMetrics	Integer	Snapshot content: instance metrics switch
contentCoreDump	Integer	Snapshot content: system core dump switch

contentThreadDump	Integer	Snapshot content: thread stack switch
threadStackCount	Integer	Thread stack capture count (1-10)
contentHeapDump	Integer	Snapshot content: heap dump switch
heapdumpMode	String	Heap dump mode (all/live)
maxSnapshots	Integer	Maximum number of snapshots to keep (1-1000)
intervalMinutes	Integer	Generation interval (minutes, 1-1440)
ruleCpuThreshold	Double	CPU usage threshold (0-100%)
ruleCpuOccurrences	Integer	CPU consecutive occurrences
ruleMemoryThreshold	Double	Memory usage threshold (0-100%)
ruleMemoryOccurrences	Integer	Memory consecutive occurrences
ruleFullGcCountThreshold	Integer	Full GC count threshold
ruleFullGcCountOccurrences	Integer	Full GC count consecutive occurrences
ruleFullGcTimeThreshold	Integer	Full GC time threshold
ruleFullGcTimeOccurrences	Integer	Full GC time consecutive occurrences
ruleHttpTimeoutThreshold	Integer	HTTP slow request count threshold
ruleHttpTimeoutOccurrences	Integer	HTTP slow request count consecutive occurrences
ruleHttpWaitThreshold	Integer	HTTP waiting task count threshold
ruleHttpWaitOccurrences	Integer	HTTP waiting task count consecutive occurrences
ruleHttpErrorThreshold	Integer	HTTP error count threshold
ruleHttpErrorOccurrences	Integer	HTTP error count consecutive occurrences
ruleJdbcWaitThreshold	Integer	JDBC waiting request count threshold
ruleJdbcWaitOccurrences	Integer	JDBC waiting request count consecutive occurrences
ruleJdbcLeakThreshold	Integer	JDBC suspected leaked connection count threshold
ruleJdbcLeakOccurrences	Integer	JDBC suspected leaked connection count consecutive occurrences

39 SnapshotController

Base Path: `/api/monitor/snapshots`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/monitor/snapshots</code>	getSnapshotList	SnapshotQueryDTO query	Result<PageResult<Snapshot>>	Paged query of snapshot list
POST	<code>/api/monitor/snapshots</code>	createSnapshot	@RequestBody SnapshotCreateDTO dto	Result<Void>	Create a manual snapshot task (generated asynchronously)
DELETE	<code>/api/monitor/snapshots/{id}</code>	deleteSnapshot	@PathVariable Long id	Result<Void>	Delete the specified snapshot
GET	<code>/api/monitor/snapshots/{id}/download/file</code>	downloadSnapshotFile	@PathVariable Long id, HttpServletResponse response	void	Download the snapshot zip file (download forbidden in pending/running status, error if the file does not exist)

SnapshotCreateDTO Fields

Field	Type	Description
contentTypes	List	Snapshot content type list (at least one must be selected, e.g. thread stack/heap/process)
threadStackCount	Integer	Thread stack capture count (1~10)
heapdumpMode	String	Heap dump mode (all=all objects / live=live objects only)

40 SnmpConfigController

Base Path: `/api/snmp-config`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/snmp-config</code>	getConfig		Result<MonitorConfig>	Get SNMP agent configuration (SNMP business fields view only; saved passwords replaced with 15 asterisks; unrelated sensitive fields such as Prometheus excluded)
PUT	<code>/api/snmp-config</code>	saveConfig	@RequestBody @Valid SnmpConfigDTO	Result<Void>	Save SNMP configuration (an empty password means keep the original value; plaintext encrypted once before storage; audit carries before/after snapshots)
POST	<code>/api/snmp-config/restore-defaults</code>	restoreDefaults		Result<Void>	Restore SNMP configuration to defaults

SnmpConfigDTO Fields

Field	Type	Description
snmpEnabled	Integer	SNMP switch (0 off / 1 on)
snmpHost	String	SNMP bind address (required, max 128)
snmpPort	Integer	SNMP bind port (1-65535)
snmpProtocol	String	Transport protocol (udp / tcp)
snmpVersion	String	SNMP version (v2c / v3)

snmpCommunity	String	SNMP community name (for v2c, max 255)
snmpSecurity	String	SNMP v3 security level
snmpUsername	String	SNMP v3 username (max 255)
snmpAuthMethod	String	SNMP v3 authentication method
snmpAuthPassword	String	SNMP v3 authentication password (max 255, empty keeps the original value)
snmpCipher	String	SNMP v3 encryption algorithm
snmpPrivPassword	String	SNMP v3 encryption password (max 255, empty keeps the original value)

41 SqlTraceController

Base Path: `/api/monitor/sql-trace`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/monitor/sql-trace/config</code>	getConfig		Result<SqlTraceConfig>	Get SQL trace configuration
POST	<code>/api/monitor/sql-trace/config</code>	saveConfig	@RequestBody @Valid SqlTraceConfig config	Result<Void>	Save SQL trace configuration (global switch/threshold/stack switch/max record count/datasource configuration)
POST	<code>/api/monitor/sql-trace/start</code>	start		Result<Void>	Start SQL tracing
POST	<code>/api/monitor/sql-trace/stop</code>	stop		Result<Void>	Stop SQL tracing
GET	<code>/api/monitor/sql-trace/records</code>	query	SqlTraceQueryDTO query (binds query parameters, @Valid)	Result<PageResult<SqlTraceRecord>>	Paged query of SQL trace records
GET	<code>/api/monitor/sql-trace/records/export</code>	export	SqlTraceQueryDTO query, HttpServletResponse response	void	Export SQL trace records as a CSV file by query conditions (query conditions validated first)

SqlTraceConfig Fields (@RequestBody entity, id is a fixed primary key, no business semantics)

Field	Type	Description
enabled	Integer	Global switch (0 off / 1 on, required)
thresholdMs	Integer	Execution duration threshold (milliseconds); only records exceeding it (required, non-negative)

stackEnabled	Integer	Whether to record the call stack (0 off / 1 on, required)
maxRecordCount	Integer	Maximum SQL record count (required, >=1)
datasourceConfigs	String	Per-datasource independent configuration (JSON string)

SqlTraceQueryDTO Fields (inherits PageQueryDTO pagination fields page/size)

Field	Type	Description
startTime	LocalDateTime	Execution time from (yyyy-MM-dd HH:mm:ss)
endTime	LocalDateTime	Execution time to
datasource	String	Datasource name filter
event	String	Event type filter
keyword	String	Keyword filter (fuzzy match on SQL text, etc.)

SqlTraceRecord Fields (returned entity)

Field	Type	Description
id	Long	Record ID
event	String	Event type
sqlText	String	SQL text
stackTrace	String	Call stack
datasource	String	Owning datasource
executeCount	Integer	Execution count
avgTimeMs	Integer	Average execution time (milliseconds)
maxTimeMs	Integer	Maximum execution time (milliseconds)
executeTime	LocalDateTime	SQL execution time
createdTime	LocalDateTime	Record creation time

42 SystemController

Base Path: `/api/system`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/system/version</code>	getVersion		Result<Map<String, String>>	Get version information (currentVersion/buildTime/javaVersion)
GET	<code>/api/system/upgrade</code>	getUpgradeHistory		Result<List<Map<String, Object>>>	Instance upgrade history list (placeholder implementation, currently returns an empty list)
POST	<code>/api/system/upgrade</code>	upgrade	@RequestBody Map<String, Object> body	Result<Map<String, Object>>	Submit an instance upgrade request (placeholder implementation, returns upgradeId/status/message)
POST	<code>/api/system/restart</code>	restart		Result<Map<String, Object>>	Restart the AAS server, returns status =RESTARTING
GET	<code>/api/system/restart-pending</code>	getRestartPending		Result<Map<String, Object>>	Query pending configuration changes (needsRestart and the changes list)

43 ThreadDiagnosisController

Base Path: `/api/monitor/thread-diagnosis`

Method	Path	Function	Parameters	Return Type	Description
POST	<code>/api/monitor/thread-diagnosis/start</code>	start	@RequestBody ThreadDiagnosisStartRequest request	Result<Void>	Start thread diagnosis sampling (intervalSec required, 1 ~ 3600 seconds; maxDataMb controls the sampling data limit)
POST	<code>/api/monitor/thread-diagnosis/stop</code>	stop		Result<Void>	Stop thread diagnosis sampling
POST	<code>/api/monitor/thread-diagnosis/session-end</code>	endSession		Result<Void>	When the session ends (navigating away/closing the page/session expiry), quietly stop monitoring and clean up the diagnosis task without generating a data file
GET	<code>/api/monitor/thread-diagnosis/snapshot</code>	getSnapshot		Result<Map<String, Object>>	Get the current thread diagnosis snapshot (thread list and status data)
GET	<code>/api/monitor/thread-diagnosis/thread/{id}/stack</code>	getThreadStack	@PathVariable("id") long threadId	Result<Map<String, Object>>	Get the stack information of the specified thread
POST	<code>/api/monitor/thread-diagnosis/thread/{id}/stop</code>	stopThread	@PathVariable("id") long threadId	Result<Void>	Stop the specified thread

POST	/api/monitor/thread-diagnosis/analyze	analyze	@RequestParam (defaultValue="live") String source	Result<Map<String, Object>>	Synchronously run thread diagnosis analysis (source defaults to live real-time data)
POST	/api/monitor/thread-diagnosis/analyze/async	startAnalyzeJob	@RequestParam (defaultValue="live") String source	Result<Map<String, Object>>	Asynchronously start an analysis task, returns jobId
GET	/api/monitor/thread-diagnosis/analyze/async/{jobId}	getAnalyzeJob	@PathVariable("jobId") String jobId	Result<Map<String, Object>>	Query the progress/result of an async analysis task
POST	/api/monitor/thread-diagnosis/analyze/async/{jobId}/cancel	cancelAnalyzeJob	@PathVariable("jobId") String jobId	Result<Void>	Cancel an async analysis task
GET	/api/monitor/thread-diagnosis/export	exportData	@RequestParam (defaultValue="live") String source, @RequestParam(required=false) Long startTime, Long endTime, HttpServletResponse response	void	Export the thread diagnosis sampling data file (a time range can be specified)
GET	/api/monitor/thread-diagnosis/export-report	exportReport	@RequestParam (defaultValue="live") String source, @RequestParam (defaultValue="html") String format, HttpServletResponse response	void	Export the diagnosis analysis report (format defaults to html)
POST	/api/monitor/thread-diagnosis/import/probe	probeImport	@RequestParam ("file") MultipartFile file	Result<Map<String, Object>>	Upload a thread diagnosis data file and pre-check it (probe file content/time range)

POST	/api/monitor/thread-diagnosis/import/confirm	confirmImport	@RequestParam(required=false) Long startTime, Long endTime	Result<Map<String, Object>>	Confirm importing the pre-checked diagnosis data (optionally trimming the time range)
GET	/api/monitor/thread-diagnosis/imported	getImportedData		Result<Map<String, Object>>	Get the currently imported diagnosis data

ThreadDiagnosisStartRequest Fields

Field	Type	Description
intervalSec	Integer	Sampling interval (seconds), required, value range 1~3600
maxDataMb	Long	Sampling data size limit (MB), 1~100; earliest data automatically discarded when reached, default 10

44 ThreadPoolController

Base Path: `/api/config/web/pools`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/web/pools</code>	list	@RequestParam int page, int size, String name, Integer maxThreads, Integer priority	Result<PageResult<ThreadPool>>	Paged query of thread pool list, supports name/max threads/priority filtering
GET	<code>/api/config/web/pools/{id}</code>	get	@PathVariable Long id	Result<ThreadPool>	Get thread pool details by ID; throws an exception if not found
GET	<code>/api/config/web/pools/defaults</code>	getDefaults		Result<Map<String, Object>>	Get thread pool default parameters (defaults such as minThreads calculated by CPU core count)
POST	<code>/api/config/web/pools</code>	create	@RequestBody ThreadPool pool	Result<Void>	Add a thread pool (validates name uniqueness), requires restart to take effect, audit recorded
PUT	<code>/api/config/web/pools/{idOrName}</code>	update	@PathVariable String idOrName, @RequestBody ThreadPool pool	Result<Void>	Update a thread pool by ID or name (validates name uniqueness), requires restart to take effect
DELETE	<code>/api/config/web/pools/{idOrName}</code>	delete	@PathVariable String idOrName	Result<Void>	Delete a thread pool by ID or name, requires restart

					to take effect, and it recorded
--	--	--	--	--	---------------------------------

ThreadPool Fields

Field	Type	Description
name	String	Thread pool name
maxThreads	Integer	Maximum threads
minThreads	Integer	Minimum threads
maxIdleTime	Integer	Maximum thread idle time (milliseconds)
queueSize	Integer	Waiting queue size
priority	Integer	Thread priority
threadStepReclaim	Integer	Thread step reclaim switch/value
threadStepSize	Integer	Thread step size (number of threads created per batch)
threadStepTime	Integer	Thread step time interval
connectionAutoReclaim	String	Automatic connection reclaim ("true"/"false")
enabled	Integer	Whether enabled (0=disabled 1=enabled)

45 ThreadTraceController

Base Path: `/api/monitor/thread-trace`

Method	Path	Function	Parameters	Return Type	Description
POST	<code>/api/monitor/thread-trace/long/start</code>	<code>startLongTrace</code>	<code>@RequestBody LongTraceStartRequest</code>	<code>Result<Void></code>	Start long-request thread tracing (both <code>thresholdMs</code> / <code>intervalMs</code> are required, otherwise 400)
POST	<code>/api/monitor/thread-trace/long/stop</code>	<code>stopLongTrace</code>		<code>Result<Void></code>	Stop long-request thread tracing
GET	<code>/api/monitor/thread-trace/long/snapshot</code>	<code>getLongTraceSnapshot</code>		<code>Result<Map<String, Object>></code>	Get the long-request trace snapshot (data of threads whose execution time exceeds the threshold)

LongTraceStartRequest Fields

Field	Type	Description
<code>thresholdMs</code>	Long	Threshold (milliseconds); requests whose execution time exceeds this value are judged as long requests, required
<code>intervalMs</code>	Long	Sampling frequency (milliseconds), required

46 UpgradeHistoryController

Base Path: `/api/system/upgrade-history`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/system/upgrade-history</code>	list	PageQueryDTO query (page/size pagination parameters)	Result<PageResult<UpgradeHistory>>	Paged query of instance upgrade history records

UpgradeHistory Fields

Field	Type	Description
version	String	Version number upgraded to
upgradeTime	LocalDateTime	Upgrade time
status	Integer	Status: 0=failed 1=success 2=in progress
note	String	Note

47 UserController

Base Path: `/api/users`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/users</code>	list	UserQueryDTO query (filters such as username/status/roleType/builtin + pagination)	Result<PageResult<UserDTO>>	Paged query of user list
GET	<code>/api/users/{id}</code>	get	@PathVariable Long id	Result<UserDTO>	Get user details by ID
POST	<code>/api/users</code>	create	@RequestBody UserDTO dto	Result<Void>	Create a user (Create validation group, including initial password), audit recorded
PUT	<code>/api/users/{id}</code>	update	@PathVariable Long id, @RequestBody UserDTO dto	Result<Void>	Update user information, audit recorded
DELETE	<code>/api/users/{id}</code>	delete	@PathVariable Long id	Result<Void>	Delete a single user, audit recorded
DELETE	<code>/api/users/batch</code>	delete Batch	@RequestBody List<Long> ids	Result<Integer>	Batch delete users, returns the number deleted
POST	<code>/api/users/{id}/status</code>	toggle Status	@PathVariable Long id, @RequestParam Integer status	Result<Void>	Enable/disable a user (status: 0=disable 1=enable)
POST	<code>/api/users/{id}/unlock</code>	unlock	@PathVariable Long id	Result<Void>	Unlock a user (clears the login lockout state)
POST	<code>/api/users/{id}/reset-password</code>	resetPassword	@PathVariable Long id, @RequestBody Map<String, String> params	Result<Void>	Reset a user's password

```
ring> body (password)
```

UserDTO Fields

Field	Type	Description
id	Long	User ID
username	String	Username (letters/digits/underscores/hyphens)
email	String	Email
phone	String	Phone
status	Integer	Status (0-2)
firstLogin	Integer	Whether first login
classLevel	Integer	Classification level
roles	List	Role code list
password	String	Initial password when creating a user (write-only, not serialized into responses)
builtin	Integer	Whether a built-in default user (admin/secure/audit), response field only
otpEnabled	Integer	Whether OTP two-factor authentication is enabled, response field only
expireDate	LocalDateTime	Account expiration time; empty means permanently valid
createdTime	LocalDateTime	Creation time
description	String	Remarks/description

48 UserPermissionController

Base Path: `/api/users`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/users/{userId}/permission</code>	getByUserId	@PathVariable Long userId	Result<UserPermissionDTO>	Get the permission configuration of the specified user (IP whitelist/time/classification level/extra roles)
PUT	<code>/api/users/{userId}/permission</code>	saveOrUpdate	@PathVariable Long userId, @RequestBody UserPermissionDTO dto	Result<Void>	Save/update user permission configuration, audit recorded (AUTH operation)

UserPermissionDTO Fields

Field	Type	Description
id	Long	Permission configuration record ID
userId	Long	User ID
allowedIps	String	IP whitelist allowed for access, multiple separated by commas
accessStartTime	String	Allowed access start time (HH:mm:ss)
accessEndTime	String	Allowed access end time (HH:mm:ss)
accessWeekdays	String	Allowed access weekdays, comma-separated 1-7
dataClassLevel	Integer	Data classification level (0=public 1=internal 2=secret 3=confidential 4=top secret)
extraRoleCodes	String	Additionally granted role codes, comma-separated
role	String	Permission assignment role (frontend field, mapped to extraRoleCodes)
ipWhitelist	String	IP whitelist (frontend field, mapped to allowedIps)
startTime	String	Access start time (frontend field, mapped to accessStartTime)

endTime	String	Access end time (frontend field, mapped to accessEndTime)
classLevel	Integer	Data classification level (frontend field, mapped to dataClassLevel)
timeLimitEnabled	Integer	Whether to enable time restriction

49 VirtualHostController

Base Path: `/api/config/web/hosts`

Method	Path	Function	Parameters	Return Type	Description
GET	<code>/api/config/web/hosts/all</code>	listAll		<code>Result<List<VirtualHost>></code>	Query all virtual hosts (no pagination, for dropdown selection)
GET	<code>/api/config/web/hosts</code>	list	<code>@RequestParam int page, int size, String name, String alias, String dcBase</code>	<code>Result<PageResult<VirtualHost>></code>	Paged query of virtual host list
GET	<code>/api/config/web/hosts/{id}</code>	get	<code>@PathVariable Long id</code>	<code>Result<VirtualHost></code>	Get virtual host details by ID; returns 404 if not found
POST	<code>/api/config/web/hosts</code>	create	<code>@RequestBody VirtualHost host</code>	<code>Result<Void></code>	Add a virtual host, requires restart to take effect, audit recorded
PUT	<code>/api/config/web/hosts/{idOrName}</code>	update	<code>@PathVariable String idOrName, @RequestBody VirtualHost host</code>	<code>Result<Void></code>	Update a virtual host by ID or name, requires restart to take effect
DELETE	<code>/api/config/web/hosts/{idOrName}</code>	delete	<code>@PathVariable String idOrName</code>	<code>Result<Void></code>	Delete a virtual host by ID or name, requires restart to take effect
PUT	<code>/api/config/web/hosts/{idOrName}/default</code>	setDefault	<code>@PathVariable String idOrName</code>	<code>Result<Void></code>	Set the specified virtual host as the default host, requires restart to take effect

VirtualHost Fields

Field	Type	Description
name	String	Virtual host name
domain	String	Domain name
alias	String	Alias
docBase	String	Application root directory (docBase)
enabled	Integer	Whether enabled
isDefault	Integer	Whether it is the default host
serviceName	String	Owning service (Service) name
logFile	String	Log file path
accessIpWhitelist	String	Access IP whitelist
accessIpBlacklist	String	Access IP blacklist
accessDomainWhitelist	String	Access domain whitelist
accessDomainBlacklist	String	Access domain blacklist
accessTime	String	Access time restriction
sso	String	Single sign-on configuration
ssoHttpOnly	Integer	Whether the SSO Cookie is HttpOnly
extraProps	String	Extended properties (custom property string)

50 Appendix: HTTP Method Description

Method	Description
GET	Retrieve a resource
POST	Create a resource
PUT	Update a resource
DELETE	Delete a resource
PATCH	Partially update a resource
REQUEST	Generic request

UNIFIED SERVICE HOTLINE

4008-555-800

Kingdee Apusic Cloud Computing Co., Ltd. (referred to as “Kingdee Apusic”) was founded in 2000. Its predecessor was “Kingdee Middleware Company.” It is a basic software platform provider under Kingdee Group, a national standards-setting enterprise for cloud computing, and a core software enterprise in the national IT application innovation industry. Kingdee Apusic is an enterprise undertaking the National 863 Key R&D Program and the National Major Science and Technology Project for Core Electronic Components, High-end General-purpose Chips and Fundamental Software, and is also a basic platform provider for the national key projects of “Two Networks, One Website, Four Databases, and Twelve Golden Projects.” Its products are widely used in key industries such as government, military, finance, and energy, and it has cumulatively served more than 100,000 customers.



National standard setter for cloud computing
Basic software enterprise under Kingdee Group
Core enterprise for IT application innovation
Website: www.apusic.com

