接口设计文档

# 一、文档概述

## 1.1 文档目的

本文档旨在规范本系统各模块接口的设计，明确接口的功能、参数、返回值、调用方式等信息，为开发人员提供清晰的开发指南，确保前后端开发工作的顺利对接，同时便于后期的接口维护和管理。

## 1.2 适用范围

本接口设计文档适用于管理系统的前后端开发人员、测试人员以及相关的项目管理人员。

## 1.3 术语与定义

* 接口：指系统中不同模块之间或不同系统之间进行数据交互的通道。
* 请求参数：前端发送给后端，用于描述请求信息的数据。
* 响应参数：后端返回给前端，包含请求处理结果的数据。
* Token：一种身份认证令牌，用于验证用户的身份合法性。

# 二、接口设计规范

## 2.1 接口命名规范

接口命名统一采用 RESTful 风格：以资源为核心，小写路径开头，通过标准 HTTP 动词或语义化后缀表达操作；保持层级简洁、含义单一，避免动词与资源混排，使路径即文档。

## 2.2 请求方法规范

* GET：用于获取资源，不改变服务器状态。
* POST：用于创建资源或提交数据，可能会改变服务器状态。
* PUT：用于更新资源的全部信息。
* DELETE：用于删除资源。

## 2.3 通用规范

* 传输协议：HTTP/HTTPS
* 数据格式：JSON（UTF-8）
* Content-Type：application/json; charset=utf-8
* 鉴权方式：登录前无鉴权；登录后请求使用 token
* 统一响应码：
  + 200 OK：业务成功
  + 400 Bad Request：参数错误
  + 401 Unauthorized：用户名/密码或 Token 无效
  + 403 Forbidden：权限不足
  + 404 Not Found：资源不存在
  + 429 Too Many Requests：触发限流
  + 500 Internal Server Error：服务端异常
* 错误码规范：
  + 1001：用户已存在
  + 1002：用户名或密码错误
  + 1003：用户不存在
  + 2000-2999：通用错误
  + 3000-3999：用户模块
  + 4000-4999：权限模块
  + 5000-5999：业务模块

## 2.4 响应规范

接口调用成功，后端应返回统一的格式，包含状态码、信息以及相关数据（data可选，为了格式统一，最好返回，无数据可返回null）。例如：

{

"code": 200,

"msg": "ok",

"data": {...}

}

当接口调用出现错误时，后端应返回统一的错误格式，包含错误码和错误信息，例如：

{

"code": 500,

"msg": "服务器内部错误"

}

# 三、登录模块接口设计

## 3.1 注册接口

* 接口URL：api/register
* 方法：POST
* 请求体：

{

"userName": "…",

"email": "…",

"phone": "…",

"password": "…"

"confirmPassword": "…"

…

}

* 成功响应：

{

"code": 0,

"msg": "ok",

"data": {...}

}

## 3.2 登录接口

* 接口URL：api/login
* 方法：POST
* 请求体：

{

"userName": "string", // 用户名

"password": "string", // 密码（建议前端加密后传输）

"captchaCode": "string", // 验证码（可选）

}

* 成功响应：

{

"code": 200,

"msg": "登录成功",

"data": {

"token": "string", // 访问令牌

"rules": ["string", …] // 用户权限列表

…

}

}

## 3.3 登出接口

* 接口URL：api/logout
* 方法：POST
* 请求体：无参数传递
* 成功响应：

{

"code": 200,

"msg": "登录成功",

"data": { 可选参数，可默认返回null

…

}

}

## 3.4 获取验证码接口

* 接口URL：api/sms/send
* 方法：POST
* 请求体：

{

"phone-number": "string" // 手机号

}

* 成功响应：

{

"code": 200,

"msg": "验证码已发送，有效期10分钟",

"data": {

"code": "…"

}

}

## 3.4 密码重置接口

* 接口URL：api/resetPassword
* 方法：POST
* 请求体：

{

"account": "string",

"verificationCode": "string", // 验证码

"newPassword": "string" // 新密码

}

* 成功响应：

{

"code": 200,

"msg": "密码重置成功"，

"data": null // 可选返回

}

# 四、用户信息接口

## 4.1 用户信息详情接口

* 接口URL：api/user/detail
* 方法：GET
* 请求体：

{

"userId": "…"

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"userId ": "…",

"userName": "…",

"email": "….",

"createdAt": "…"

…

}

}

## 4.2 更新用户信息接口

* 接口URL：api/user/update
* 方法：PUT/POST
* 请求体：

{

" userId ": "…", // 必填

"userName": "…",

"email": "…",

"createdAt": "…"

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

# 五、用户管理接口

## 5.1 用户列表接口

* 接口URL：api/admin/user/list
* 方法：GET
* 请求体：

{

"page ": "…", // 页码，从1开始

"size": "…", // 每页条数

"userName": "….", // 用户名，模糊查询

"roleId": "…", // 角色ID，模糊查询

"status": "…", // 状态，0-禁用，1启用

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 5.2 用户审核接口

* 接口URL：api/admin/user/audit
* 方法：POST
* 请求体：

{

"userId": "…",

"status": "0"，// 审核状态

"reason": "…" // 审核拒绝时必填

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"userId": "…",

"auditStatus": 1,

…

}

}

## 5.3 用户删除接口

* 接口URL：api/admin/user/delete
* 方法：DELETE/POST
* 请求体：

{

"userId": "…"

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

## 5.4 用户权限

### 1. 列表接口

* 接口URL：api/admin/permissions/list
* 接口描述：查询系统所有权限，支持树形结构展示
* 方法：GET
* 请求体：

{

"page ": "…", // 页码，从1开始

"size": "…", // 每页条数

"permissionsName": "….", // 权限名，模糊查询

"permissionsId": "…", // 权限ID，模糊查询

"status": "…", // 状态，0-禁用，1启用

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/admin/permissions/add
* 方法：POST
* 请求体：

{

"permissionsName": "…",

"parentId ": "…",

"status": "…", // 状态，0-禁用，1启用

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/admin/permissions/update
* 方法：PUT/POST
* 请求体：

{

"permissionsId": "…", // 必填

"permissionsName": "…",

"parentId ": "…",

"status": "…", // 状态，0-禁用，1启用

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/admin/permissions/delete
* 方法：DELETE/POST
* 请求体：

{

"permissionsId": "…",

"parentId": "…",

"permissionsName": "…"

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5. 详情接口

* 接口URL：api/admin/permissions/detail
* 方法：GET
* 请求体：

{

"permissionsId": "…",

"parentId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"permissionsId": "…",

"permissionsName": "…",

"permCode": "user:manage",

…

}

}

## 5.5 用户角色

### 1. 列表接口

* 接口URL：api/admin/role/list
* 方法：GET
* 请求体：

{

"page": "…", // 页码，从1开始

"size": "…", // 每页条数

"roleName": "….", // 角色名，模糊查询

"roleId": "…", // 角色ID，模糊查询

"status": "…", // 状态，0-禁用，1启用

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/admin/role/add
* 方法：POST
* 请求体：

{

"roleName": "…",

"roleCode": "…",

"status": "…", // 状态，0-禁用，1启用

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/admin/role/update
* 方法：PUT/POST
* 请求体：

{

"roleId": "…", // 必填

"roleName": "…",

"roleCode ": "…",

"status": "…", // 状态，0-禁用，1启用

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/admin/role/delete
* 方法：DELETE/POST
* 请求体：

{

"roleId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5. 详情接口

* 接口URL：api/admin/role/detail
* 接口描述：根据角色 ID 查询角色详情及关联权限
* 方法：GET
* 请求体：

{

"roleId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"roleId": 1,

"roleName": "超级管理员",

"permissions": [

{ … },

…

],

…

}

}

### 6. 分配角色权限接口

* 接口URL：api/admin/role/assignPermissions
* 接口描述：为指定角色分配权限
* 方法：POST
* 请求体：

{

"roleId": 5,

"permissionIds": [101, 102, 103, 104, 105]

}

* 成功响应：

{

"code": 200,

"msg": "角色权限分配成功",

"data": {

"updateTime": "2023-08-04 17:20:00"

}

}

### 7. 分配用户角色接口

* 接口URL：api/admin/role/assignUserRoles
* 接口描述：为指定用户分配一个或多个角色，实现用户与角色的关联。若用户已拥有某些角色，将更新为新的角色集合（即先移除原有角色关联，再建立新的关联）。
* 方法：POST
* 请求体：

{

"userId": "…", // 用户ID，必填

"roleIds": [1, 2, 3] // 角色ID列表，数组，必填，至少包含一个角色ID

}

* 成功响应：

{

"code": 200,

"msg": "用户角色分配成功",

"data": {

"userId": "…",

"roleIds": [1, 2, 3],

…

}

}

# 六、仪表盘（首页）接口

## 6.1 告警量、阻断量等数据接口

* 接口URL：api/getDashboardData
* 接口描述：返回在线数据源数量、告警量、阻断量、接入总量等数据
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"onlineVolume": 20,

"alarmVolume": 10,

…

}

}

## 6.2 每日接入/转发的流量数据（折线图）

* 接口URL：api/getTrafficData
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"accessData": [],

"forwardData": [],

…

}

}

## 6.3 正在接入的数据源及传输速度接口

* 接口URL：api/getAccessData
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

"page ": 1, // 页码，从1开始

"size": 5, // 每页条数

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 6.4 正在转发的数据源及传输速度接口

* 接口URL：api/getForwardData
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

"page ": 1, // 页码，从1开始

"size": 5, // 每页条数

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 6.5 告警等级及数量接口

* 接口URL：api/getAlarmLevel
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": { // 告警等级（1低，2中，3高，4致命）

"low":11,

"middle":6,

"high":2,

…

}

}

## 6.6 告警类型及比例分布（饼图）

* 接口URL：api/getAlarmTypeRatio
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"alarmData": [],

…

}

}

## 6.7 告警列表

* 接口URL：api/alertEvent/list
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

"page": 1, // 页码，从1开始

"size": 5, // 每页条数

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 6.8 每日阻断的可疑访问统计情况接口

* 接口URL：api/get**SuspiciousBehavior**
* 方法：GET
* 请求体：

{

"period": "…", // 用户选择的时间周期，默认值3个月

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"**suspicious**Data": [],

…

}

}

# 七、原始日志列表（接入/转发流量列表）

## 7.1 列表接口

* 接口URL：api/originLog/list
* 方法：GET
* 请求体：

{

"sourceIp": "…", // 源IP地址

"protocol": "…", // 协议类型（HTTP/HTTPS/WS等）

"page ": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 7.2 详情接口

* 接口URL：api/originLog**/detail**
* 方法：GET
* 请求体：

{

"originLogId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"ip": "…",

…

}

}

## 7.3 删除接口

* 接口URL：api/originLog**/delete**
* 方法：DELETE/POST
* 请求体：

{

"originLogId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

## 7.4 导出接口

* 接口URL：api/originLog**/export**
* 方法：GET
* 请求体：

{

"destIp": "…", // 目标IP地址

"payloadSize": "…", // 数据包大小（字节）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}

# 八、端侧访问日志

## 8.1 列表接口

* 接口URL：api/accessLog/list
* 方法：GET
* 请求体：

{

"ip": "…",

"url": "…",

"page ": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 8.2 详情接口

* 接口URL：api/accessLog**/detail**
* 方法：GET
* 请求体：

{

"accessLogId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"ip": "…",

…

}

}

## 8.3 删除接口

* 接口URL：api/accessLog**/delete**
* 方法：DELETE/POST
* 请求体：

{

"accessLogId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

## 8.4 导出接口

* 接口URL：api/accessLog**/export**
* 方法：GET
* 请求体：

{

"referer": "…", // 来源地址

"userAgent": "…", // 浏览器信息

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}

# 九、端侧用户指纹

## 9.1 列表接口

* 接口URL：api/**userFingerprint/list**
* 方法：GET
* 请求体：

{

"browserInfo": "…", // 浏览器类型和版本

"osInfo": "…", // 操作系统信息

"page ": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 9.2 详情接口

* 接口URL：api/**userFingerprint/detail**
* 方法：GET
* 请求体：

{

"fingerprintId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"browserInfo": "…",

…

}

}

## 9.3 删除接口

* 接口URL：api/**userFingerprint/delete**
* 方法：DELETE/POST
* 请求体：

{

"fingerprintId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

## 9.4 导出接口

* 接口URL：api/**userFingerprint/export**
* 方法：GET
* 请求体：

{

"deviceHash": "…", // 设备唯一标识hash

"canvasHash": "…", // Canvas指纹Hash值

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}

## 9.5 用户行为查询接口

* 接口URL：api/**userFingerprint/userBehavior**
* 方法：GET
* 请求体：

{

"fingerprintId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"behaviorType": "…", // 行为类型（页面浏览、按钮点击、下载等）

"targetObject": "…", // 行为目标对象（页面名称、API地址、文件名等）

…

}

}

# 十、告警列表

## 10.1 列表接口

* 接口URL：api/alertEvent/list
* 方法：GET
* 请求体：

{

"eventType": "…", // 告警类型（异常流量、攻击检测等）

"severity": "…", //告警等级（1低，2中，3高）

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 10.2 详情接口

* 接口URL：api/alertEvent/**detail**
* 方法：GET
* 请求体：

{

"alertEventId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"eventType": "…",

…

}

}

## 10.3 删除接口

* 接口URL：api/alertEvent/delete
* 方法：DELETE/POST
* 请求体：

{

"alertEventId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

## 10.4 导出接口

* 接口URL：api/alertEvent/export
* 接口描述：根据筛选条件导出告警列表数据，支持导出为 Excel 格式，包含告警的详细信息。接口采用同步方式处理，直接返回文件流。
* 方法：GET
* 请求体：

{

"eventType": "…", // 告警类型（异常流量、攻击检测等）

"severity": "…", // 告警等级（1低，2中，3高）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}

# 十一、阻断列表

## 11.1 列表接口

* 接口URL：api/block/list
* 方法：GET
* 请求体：

{

"ip": "…", // 被阻断IP地址

"reason": "…", // 阻断原因（高频访问等）

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

## 11.2 详情接口

* 接口URL：api/block/**detail**
* 方法：GET
* 请求体：

{

"blockId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"ip": "…",

…

}

}

## 11.3 删除接口

* 接口URL：api/block/delete
* 方法：DELETE/POST
* 请求体：

{

"blockId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

## 11.4 导出接口

* 接口URL：api/block/export
* 方法：GET
* 请求体：

{

"ip": "…", // 被阻断IP地址

"reason": "…", // 阻断原因（高频访问等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}

# 十二、系统配置

## 12.1 接入配置

### 1. 列表接口

* 接口URL：api/access/list
* 方法：GET
* 请求体：

{

"ip": "…", // 节点IP

"region": "…", //节点所在区域

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/access/add
* 方法：POST
* 请求体：

{

"nodeName ": "…",

"ip": "…",

"region": "…",

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/access/update
* 方法：PUT/POST
* 请求体：

{

"accessId": "…", // 必填

"nodeName": "…",

"ip": "…",

"region": "…",

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/access/delete
* 方法：DELETE/POST
* 请求体：

{

"accessId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5. 详情接口

* 接口URL：api/access**/detail**
* 方法：GET
* 请求体：

{

"accessId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"nodeName": "…",

…

}

}

### 6. 操作记录接口

* 接口URL：api/access**/operationRecords**
* 方法：GET
* 请求体：

{

"accessId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"operationType": "…",

…

}

}

## 12.2 转发配置

### 1. 列表接口

* 接口URL：api/forward/list
* 方法：GET
* 请求体：

{

"ruleName": "…", // 规则名称

"conditions": "…", //转发条件定义（如指定协议、特定URL、指定处理类型等）

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/forward/add
* 方法：POST
* 请求体：

{

…

"forwardMode": "…", // 转发方式（SINGLE=单目标，MULTI=多目标，BROADCAST等）

"pathPriority": "…", // 转发路径优先级（数值越小优先）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/forward/update
* 方法：PUT/POST
* 请求体：

{

"forwardId": "…"，// 必填

"ruleName": "…", // 规则名称

"conditions": "…", //转发条件定义（如指定协议、特定URL、指定处理类型等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/forward/delete
* 方法：DELETE/POST
* 请求体：

{

"forwardId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5. 详情接口

* 接口URL：api/forward**/detail**
* 方法：GET
* 请求体：

{

"forwardId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"ruleName": "…",

…

}

}

### 6. 操作记录接口

* 接口URL：api/forward**/operationRecords**
* 方法：GET
* 请求体：

{

"forwardId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"operationType": "…",

…

}

}

## 12.3 黑/白名单配置

### 1. 列表接口

* 接口URL：api/accessControl/list
* 方法：GET
* 请求体：

{

"listType": "…", // 列表类型（1.BLACK=黑名单，2.WHITE=白名单）

"matchType": "…",// 匹配对象类型（1.IP、2.FINGERPRINT、3.UA、4.DOMAIN）

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/accessControl/add
* 方法：POST
* 请求体：

{

"matchValue": "…", // 匹配值（如具体IP、指纹ID、UA关键字、域名等）

"priority": "…", // 优先级（数值越小优先级越高，黑白名单规则冲突时生效顺序）

"source\_url": "…",

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/accessControl/update
* 方法：PUT/POST
* 请求体：

{

"accessControlId": "…", // 必填

"matchValue": "…", // 匹配值（如具体IP、指纹ID、UA关键字、域名等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/accessControl/delete
* 方法：DELETE/POST
* 请求体：

{

"accessControlId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5. 详情接口

* 接口URL：api/accessControl/**detail**
* 方法：GET
* 请求体：

{

"accessControlId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"matchValue": "…",

…

}

}

### 6. 操作记录接口

* 接口URL：api/accessControl/**operationRecords**
* 方法：GET
* 请求体：

{

"accessControlId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"operationType": "…",

…

}

}

## 12.4 协议解析规则配置

### 1. 列表接口

* 接口URL：api/protocolAnalysis/list
* 方法：GET
* 请求体：

{

"ruleName": "…", // 规则名称

"sourceId": "…", // 处理类型(NORMAL/BLOCK/ALERT/VERIFY)

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/protocolAnalysis/add
* 方法：POST
* 请求体：

{

"protocolType": "…", // 协议类型（HTTP、HTTPS、TCP、UDP、自定义等）

"dataFormat": "…", // 数据格式（JSON、XML、RAW、BINARY等）

"parseStructure": "…", // 解析结构定义（例如字段提取规则、正则表达式等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/protocolAnalysis/update
* 方法：PUT/POST
* 请求体：

{

"protocolAnalysisId": "…", // 必填

"dataMapping": "…", // 解析后字段与系统内部字段的映射配置

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/protocolAnalysis/delete
* 方法：DELETE/POST
* 请求体：

{

"protocolAnalysisId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5．详情接口

* 接口URL：api/protocolAnalysis/**detail**
* 方法：GET
* 请求体：

{

"protocolAnalysisId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"dataMapping": "…",

…

}

}

### 6. 操作记录接口

* 接口URL：api/protocolAnalysis/**operationRecords**
* 方法：GET
* 请求体：

{

"protocolAnalysisId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"operationType": "…",

…

}

}

## 12.5 流量转发规则配置

### 1. 列表接口

* 接口URL：api/trafficForwarding/list
* 方法：GET
* 请求体：

{

"ruleName": "…", // 规则名称

"conditions": "…", // 转发条件定义（如指定协议、特定URL、指定处理类型等）

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/trafficForwarding/add
* 方法：POST
* 请求体：

{

…

"forwardMode": "…", // 转发方式（SINGLE=单目标，MULTI=多目标，BROADCAST等）

"pathPriority": "…", // 转发路径优先级（数值越小优先）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/trafficForwarding/update
* 方法：PUT/POST
* 请求体：

{

"tfId": "…", // 必填

"failoverStrategy": "…", // 失败处理（RETRY、BACKUP、DROP等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/trafficForwarding/delete
* 方法：DELETE/POST
* 请求体：

{

"tfId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5. 详情接口

* 接口URL：api/trafficForwarding/**detail**
* 方法：GET
* 请求体：

{

"tfId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"failoverStrategy": "…",

…

}

}

### 6. 操作记录接口

* 接口URL：api/trafficForwarding/**operationRecords**
* 方法：GET
* 请求体：

{

"tfId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"operationType": "…",

…

}

}

## 12.6 异常告警规则配置

### 1. 列表接口

* 接口URL：api/alertRules/list
* 方法：GET
* 请求体：

{

"ruleName": "…", // 规则名称

"metricType": "…", // 监控指标类型（QPS、BLOCK率、状态码5xx、延迟、爬虫频率等）

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 增加接口

* 接口URL：api/alertRules/add
* 方法：POST
* 请求体：

{

"ruleName": "…", // 规则名称

"metricType": "…", // 监控指标类型（QPS、BLOCK率、状态码5xx、延迟、爬虫频率等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 编辑接口

* 接口URL：api/alertRules/update
* 方法：PUT/POST
* 请求体：

{

"alertRulesId": "…", // 必填

"alertLevel": "…", // 告警级别（1.INFO、2.WARNING、3.CRITICAL等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 4. 删除接口

* 接口URL：api/alertRules/delete
* 方法：DELETE/POST
* 请求体：

{

"alertRulesId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 5. 详情接口

* 接口URL：api/alertRules/**detail**
* 方法：GET
* 请求体：

{

"alertRulesId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"alertLevel": "…",

…

}

}

### 6. 操作记录接口

* 接口URL：api/alertRules/**operationRecords**
* 方法：GET
* 请求体：

{

"alertRulesId": "…",

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": {

"operationType": "…",

…

}

}

# 十三、日志管理

## 13.1 登录日志

### 1. 列表接口

* 接口URL：api/log/login/list
* 方法：GET
* 请求体：

{

"userName": "…",

"userAgent": "…", // 浏览器或访问工具UA字符串

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 删除接口

* 接口URL：api/log/login/delete
* 方法：DELETE/POST
* 请求体：

{

"loginLogId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 导出接口

* 接口URL：api/log/login/export
* 方法：GET
* 请求体：

{

"userName": "…",

"loginStatus": "…", // 登录结果（1=成功，0=失败）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}

## 13.2 操作日志

### 1. 列表接口

* 接口URL：api/log/operation/list
* 方法：GET
* 请求体：

{

"userName": "…",

"operationType": "…", // 操作类型（页面访问、配置修改、规则新增、规则删除等）

"page": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 删除接口

* 接口URL：api/log/operation/delete
* 方法：DELETE/POST
* 请求体：

{

"operationLogId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 导出接口

* 接口URL：api/log/operation/export
* 方法：GET
* 请求体：

{

"userName": "…",

"ipAddress": "…", // 操作来源IP

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}

## 13.3 系统日志接口

### 1. 列表接口

* 接口URL：api/log/system/list
* 方法：GET
* 请求体：

{

"packet\_id": "…", // 一条流量记录

"eventType": "…", // 系统事件类型（流量转发、流量清洗、节点切换等）

"page ": 1, // 页码，从1开始

"size": 10, // 每页条数

…

}

* 成功响应：

{

"code": 200,

"msg": "ok",

"data": {

"total": 100, // 总条数

"records": [ // 列表数组

{…},

…

],

…

}

}

### 2. 删除接口

* 接口URL：api/log/system/delete
* 方法：DELETE/POST
* 请求体：

{

"systemLogId": "…", // 如果需要批量删除，值改用数组类型

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": null // 可选返回

}

### 3. 导出接口

* 接口URL：api/log/system/export
* 方法：GET
* 请求体：

{

"packet\_id": "…", // 一条流量记录

"eventType": "…", // 系统事件类型（流量转发、流量清洗、节点切换等）

…

}

* 成功响应：

{

"code": 200,

"msg": "ok"，

"data": [Excel文件二进制流]

}