软件设计说明书

SPIDER1.0.0.1（_XXX 子系统）

[填写说明：模板中用方括号括起来并以蓝色斜体显示的文本，用于向作者提供指导，在文档编辑完成后应该将其删除。文档正文应使用常规、黑色、五号字体即系统设置的“正文”样式]

文档页眉处的“xxxx 系统”和“版本号”仅为示例，请注意更新封页与页眉符合实际情况。此处的版本号指的是产品版本号。封页简要表中的产品名和子系统名，如无可以不填写。

当某一章节如没有内容时，必须注明 NA，同时标注理由。例如：本节内容无需考虑。特别说明：当本节内容参见其它文档内容，不能注明 NA，而应该写明参见某文档的具体章节。]

[特殊说明：产品对应的相关系统以及逻辑和物理结构划分参照下图]

文档版本号：		文档编号：	
文档密级：	保密	归属部门/项目：	
产品名：		子系统名：	
编写人：		编写日期：	

修订记录：

版本号	修订人	修订日期	修订描述
V1.0A	EPG 团队	2013-03-27	初稿
V1.0B	陈凌	2013-9-23	增加需求跟踪
V2.0	李雪峰	2023-03-10	参考公司的标准模板，对于动环项目进行补充和裁剪 2.1　概述，增加对重要需求点的列表阐述，避免设计时遗漏 2.4　模块间交互图，为了避免把大量的逻辑揉在一个时序图，分成子章节分别对业务流程进行设计
V2.1	孙科	2023-07-20	根据部门设计准入标准，增加执行检查点 增加 2.6 公共设计 增加 2.5 影响范围评估 增加 2.9.3 网络资源需求梳理 增加 2.9.4 安全性检查
V2.2	孙科	2024-03-27	增加 2.9.6 风险小节
V2.3	黄晓锋	2025-05-12	增加 2.9.4.9 垂直越权小节 增加 2.9.4.10 水平越权小节

派发清单：

发文人/部门	日期	电话/传真

受文人/部门	动作类型*	日期	电话/传真

*动作类型：批准、审核、通知、归档、参与会议，其它（请说明）

目 录

1 简介

1.1 目的

［本文档是对方案设计提供一个高层和底层的综合描述，有些系统无需将设计分为高层和底层两个阶段，而是直接用一个设计阶段完成高层和底层设计的工作，则可采用此模板。］

软件设计的目的是根据产品需求、软件需求文档中对需求的描述，结合架构设计文档中对方案的描述，对具体实现方案进行细化，精确到各模块的功能、具体的实现及其模块之间的交互，明确模块中如何通过源代码实现相关的接口和功能，描述其中关键的对外接口及内部函数的实现，关键的全局变量，使用到的数据库对象（包括表、数据、存储过程等）。本文档对编码和单元测试工作进行直接指导，并对测试工作起到辅导作用。

其他情况请另外说明。］

1.2 文档范围

［例如，本文档适用的产品、模块，覆盖的范围等，受这份文档影响的相关模块等，不在该文档覆盖范围内的但可能引起疑义的问题。］

1.3 预期的读者和阅读建议

［说明此文档的阅读对象，对读者的要求。简要说明此文档中其它章节包含的内容与文档组织方式，对于不同读者的阅读方式建议。

如：

目标读者是系统最终用户、系统分析员、项目经理、产品经理、市场人员等。

此文档的第 2 章描述…..］

本文档组织方式：

第一章 简介，描述文档的目的；
第二章 描述设计原则，开发人员在开发过程中必须遵循这些原则；
第三章 描述设计策略；
第四章 描述模块重用设计，设计人员和开发人员需要根据这部分的描述重用以前的设计并关注可供将来重用的设计部分；
第五章 方案设计部分描述设计方案，包括逻辑结构、模块划分、模块间关系及交互图、状态图等设计图例，共设计人员和开发人员了解方案；
第六章 模块设计，详细描述各个模块的具体设计及其单元设计，可指导开发人员进行模

块设计和相关测试工作；
第七章 系统容错处理，描述可能的错误信息和处理措施；
第八章 描述系统可维护性设计；
第九章 描述设计的假设前提；
第十章 描述设计中的风险，相关人员应当关注风险对产品可能造成的影响；
第十一章 附录。

1.4 参考文档

[适当时，提供相关的包含文档及参考文档。

软件设计说明书的参考文档应当包括但不限于：产品需求说明书，软件需求说明书，架构设计说明书，数据库设计说明书，高层设计说明书、设计规范、编码规范等；

同时，文档中说明为引用、参考的文档也应该在这里列出。

参考文档请按包含、相关的关系分别在下面列出。]

1.4.1 包含文档

包含的文档名称及其版本	文档路径	对应的章节名称
内部接口设计文档		

[包含文档：作为本软件设计的一部分，不可分割的组成部分，读者阅读本设计说明书时必须同时也阅读的文档。如数据库设计说明书（如果数据库设计有变更）。

通常情况下，软件设计说明书没有包含文档。]

如有接口文档该章节为必填，请补充接口文档名称及 SVN 归档路径

【检查点：系统内部接口】是否涵盖内部接口调用设计内容，该设计内容是否清晰合理，依赖耦合性是否较低，是否存在相互依赖的网状调用、是否存在同在同层级调用、是否存在不该依赖的接口调用等

【检查点：外部系统接口】是否涵盖外部系统接口交互设计内容，该设计内容是否清晰合理，是否复用当前接口，是否迫于压力以对方接口为准新增接口，是否有考虑推动以我方为主提供接口规范，接口协议是否合理，接口安全性是否有考虑，接口性能是否有考虑

1.4.2 相关文档【必填】

参考的文档名称及其版本	文档路径	对应的章节名称

[相关文档：作为引用而包含的文档。读者在阅读本设计说明书时如果有必要可以参考阅读的文档。如产品需求说明书、设计规范文档等]

如有该章节为必填，请补充需求文档名称及 SVN 归档路径，如非独立需求文档请补充具体章节；如涉及外部接口文档请一并补充

2 方案设计

2.1 概述

[描述本系统或子系统的设计概述，可参考或者引用架构设计中的相关内容，以达到从架构设计到高层设计之间的上下关联作用]

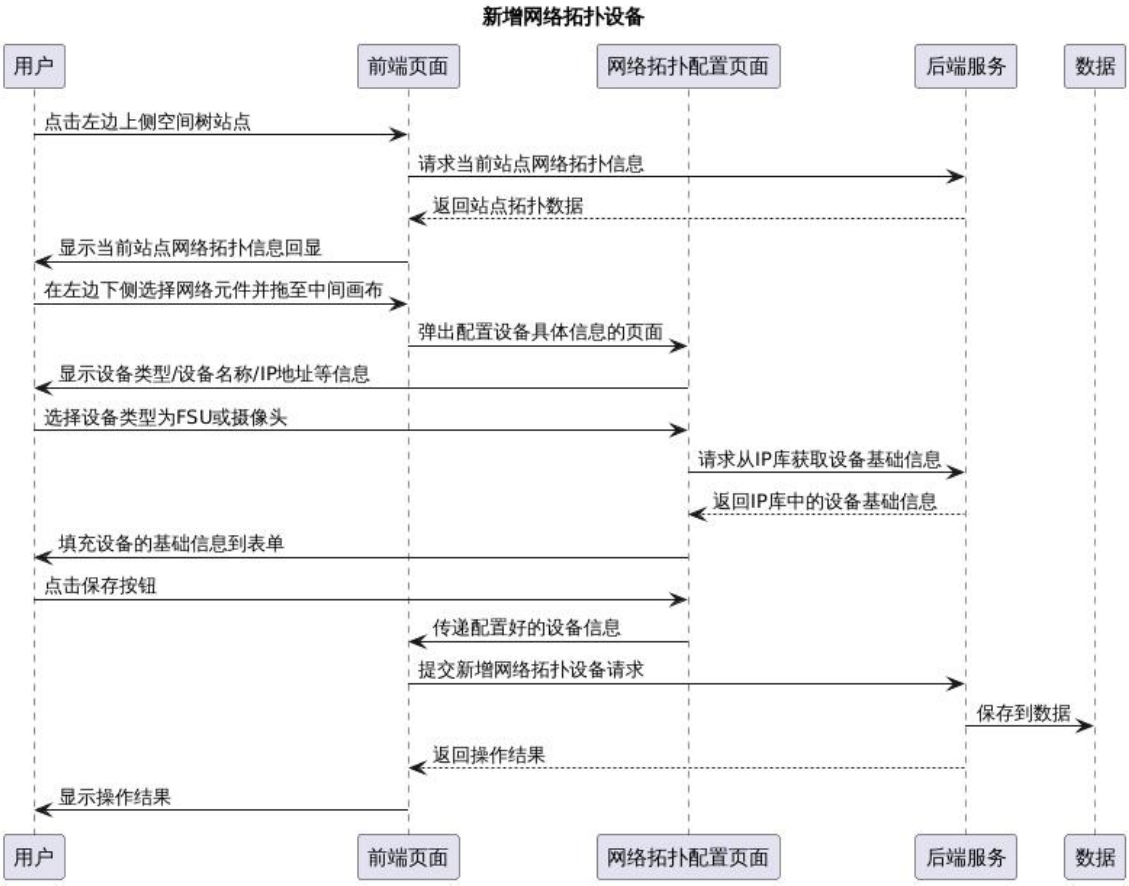
蓄电池充放电管理页面支持设备充放电策略的查询、新增、修改、删除和策略的启停，策略调控日志的查询。

2.2 逻辑结构

[使用框线图或者 UML 描述本系统或子系统的逻辑结构，包括模块的划分与各模块间的依赖关系。]

程序设计规范性：是否按照公司标准模板进行设计（包括**对象、时序、包、类、方法**等），是否有遗漏，同时满足公司软件设计规范要求

2.2.1 新增网络拓扑设备：



用户点击左边上侧空间树站点会进入绘制画布页面，前端页面显示当前站点网络拓扑信息回显，用户在左边下侧选择网络元件并拖中间绘制画布，然后用户点击设备时前端页面会侧滑出配置设备具体信息的页面，显示字段有设备类型、设备名称、IP 地址、MAC 地址、设备型号、所属专业、设备厂家等信息，具体字段信息在接口文档中体现，用户如果选择设备类型为 FSU 或摄像头时，设备名信息从 IP 库选择填充设备的基础信息，当设备是跨站点时，也可以进行连接，只是连接线的颜色要有不一样的，填写完信息后点击保存，前端页面提交新增网络拓扑设备请求，保存到数据，后端服务返回操作结果，前端页面显示操作结果。
注：在新增的时候，如果上游设备不存在，那么就不会画出指向线，如果创建时上游设备不存在，后面添加了该设备元件，那么也要在拓扑上体现。

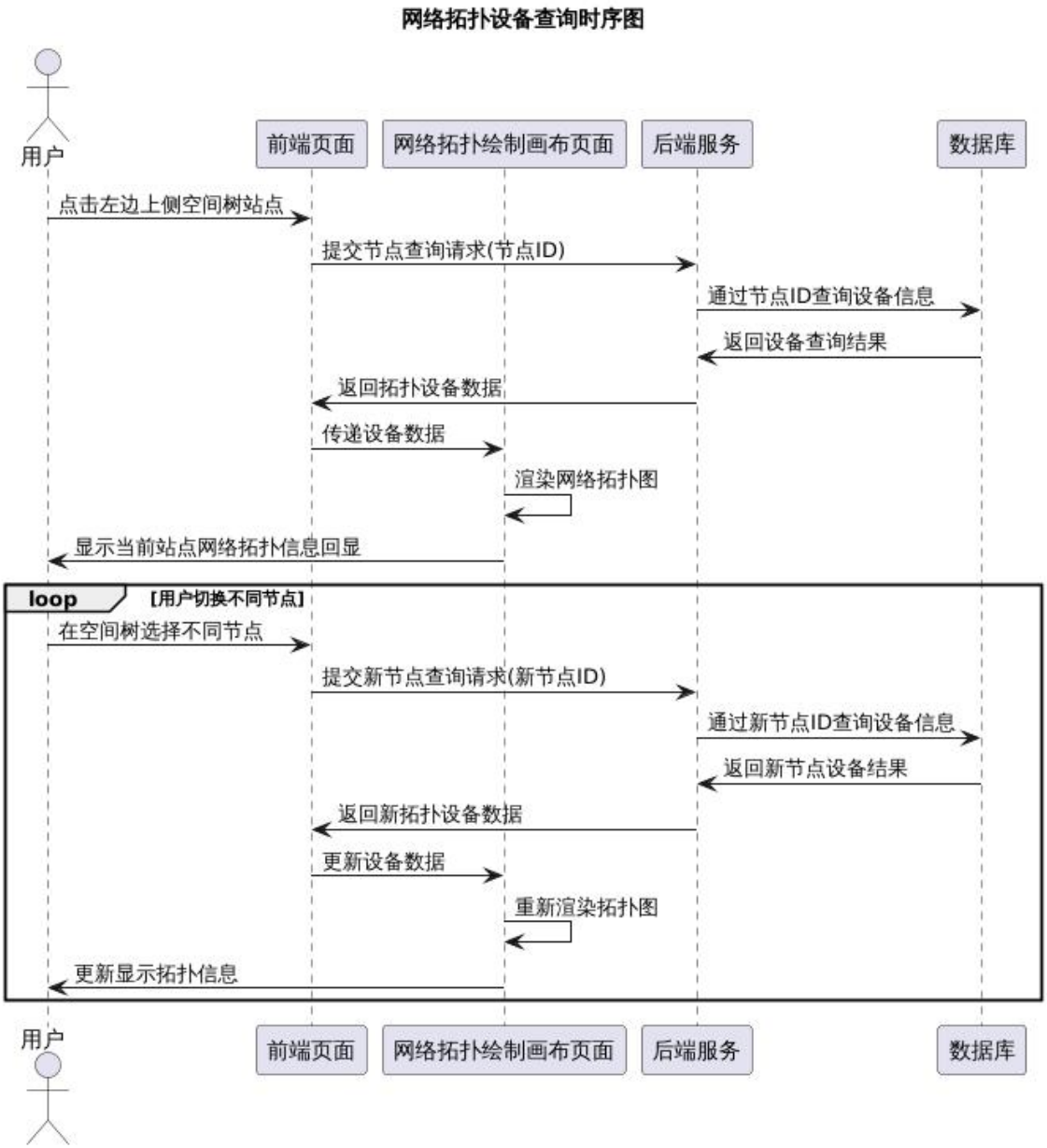
2.2.2 编辑网络拓扑设备：

编辑的新增的操作是类似的，用户点击左边上侧空间树站点会进入绘制画布页面，前端页面显示当前站点网络拓扑信息回显，然后用户在中间绘制画布点击要修改的设备时前端页面会侧滑出配置设备具体信息的页面，显示字段有设备类型、设备名称、IP 地址、MAC 地址、设备型号、所属专业、设备厂家等信息，具体字段信息在接口文档中体现，用户如果选择设备类型为 FSU 或摄像头时，设备名信息从 IP 库选择填充设备的基础信息，当设备是跨站点时，也可以进行连接，只是连接线的颜色要有不一样的，填写完信息后点击保存，前端页面提交修改网络拓扑设备请求，保存到数据，后端服务返回操作结果，前端页面显示操作结果。
注：要注意在添加或编辑时上游设备不存在的情况，有可能是后续添加的。

2.2.3 删除网络拓扑设备：

用户点击左边上侧空间树站点会进入绘制画布页面，前端页面显示当前站点网络拓扑信息回显，然后用户在中间绘制画布点击要删除的设备时前端页面会侧滑出配置设备具体信息的页面，点击删除网络拓扑设备，就可以删除掉这条网络拓扑设备信息。

2.2.4 网络拓扑设备查询：



这块是一个网络拓扑展示的页面，用户选择不同的空间树节点，展示对应节点的网络拓扑信息，中间绘制画布展示网络拓扑信息， 右边滑出一个层按设备分类展示设备信息。需要连表查询涉及到的表有：

t_cfg_precinct：区域基本表

t_cfg_device：设备基本表

t_cfg_ip：ip 配置表

t_cfg_topology：设备网络拓扑配置表

t_cfg_topology_config_part：拓扑配置-元件（设备）

区域表和设备表是通过 precinct_id 区域编号连接的

2.2.5 网络拓扑设备查询（站点以上）

用户点击左边上侧空间树站点，前端绘制画布页面显示当前站点网络拓扑信息回显 ，用户在空间树选择站点级以上节点进行查询 ， 前端页面提交到后端，后端服务通过节点 Id，和节点类型进行查询设备信息，返回查询结果， 当节点类型为站点及以上时，只站点级的拓扑，不做具体的拓扑展现 ， 前端页面显示操作结果。

2.3 模块定义

［描述模块的划分及各模块的总体功能］

模块的外部接口和入参/出参，基于接口的设计，就先不强制在设计文档中体现，但是需要在代码中把外部接口、外部接口涉及的 view 对象和内部的分离，并且添加注释

模块名称	模块描述*[主要描述模块的功能及其意义]*	模块状态*[重用/变更/新增]*
composite-scada	*暴漏 Feign 接口，实现服务间的交互*	*重用*
enerApp-scada	*业务逻辑在此模块实现*	*重用*

2.4 模块间交互图

[列出本系统或子系统所有模块的交互（SEQUENCE）图，并进行必要的文字描述，需要反映出对需求关键点的设计实现分解，如果涉及到多个关键流程，那么需要多个时序图]

2.5 影响范围评估

程序设计是否完全覆盖了版本需求内容，改造开发点是否有考虑对其他程序的影响并给出合理的设计

特别是已有功能优化，老功能改造该章节为必填。

2.6 公共设计

2.6.1 关键技术考虑

难点、重点技术是否有考虑、是否有研究、是否有针对该技术业务的设计、是否可行能落地、是否提供实现方式

2.6.2 公共机制考虑

是否有考虑公用抽象工具类、公共类/组件，是否存在为了避免麻烦将可公共的地方写在了私有包或者项目工程中

动环项目新需求开发，**禁止复制 V 版本中的代码到项目中**，组件及工具类不满足的应进行扩展

2.6.3 重用性考虑

系统重用、组件复用考虑，是否已考虑、进行系统、组件重用、复用设计

动环所有新项目都必须复用能源平台 V 版本，新需求开发都需要考虑与 V 版本的关系，产品化要求及组件化可行性

2.7 数据流图

[列出本系统或子系统所有模块的数据流图，并进行必要的文字描述。]

2.8 数据存储设计

2.8.1 逻辑模型

[这里要用框线图按照业务涉及的业务实体（非物理模型的表），列出不同业务实体的关系（几对几），业务实体要包括主要的业务属性]

2.8.2 物理模型列表

【重要约束】：对于估计累计数据量超过一千万的表不容许放在 mysql，否则需要拆成多表

检查点:数据结构规范性考虑

表名	*t_cfg_topology_v3_configuration*
表注释	*网络拓扑配置表*
设计考虑	*数据存储在 MYSQL*
数据操作类型	*增加数据/修改数据/删除数据/查询单体数据详情/查询数据列表/查询聚合数据*
数据访问频率	[检查点：数据量性能考虑]*数据访问频次 xx 次/秒*
数据增长量（每月）	[检查点：数据量考虑]*数据增量的估算公式*
数据量峰值（数量级）	[检查点：数据量考虑　对数据库表的数量级有清晰的评估，并且针对数据量较大的表进行了高性能、扩展性设计，确保数据读取性能]*数据生命周期内的累计量*
数据生命周期/淘汰归档规则	[检查点：数据量性能及扩展性考虑]*合数据的生命周期是几年/无限制，生命周期到达后，是否支持自动数据清理/运维手工清*

	理		
Field Name	Field Type	Key	Description
id	bigint		主键 Id
device_name	varchar		设备名称
ip	bigint		IP 地址,数据库唯一
mac	varchar		MAC 地址
device_model	varchar		设备型号
up_device_name	varchar		上游网络设备
up_device_ip	varchar		上游设备 IP
cross_site	int		是否跨站点
ip_v6	bigint		IPv6 地址
gateway	bigint		网关
virtual_ip	bigint		虚拟 IP
port_num	int		端口号
major	varchar		所属专业
device_vender	varchar		设备厂家
site_id	varchar		站点 id
site_name	varchar		站点名
room_name	varchar		机房名
room_id	varchar		机房 id
location_desc	varchar		具体位置
version_num	varchar		版本号
serial_num	varchar		序列号
port_count	int		端口数量
draw_content	text		绘制内容(json)
precinct_id	varchar		节点 Id
create_by	varchar		创建账号
create_time	datetime		创建时间
update_by	varchar		更新账号
update_time	datetime		更新时间
主键	PRIMARY KEY (id)		
外键			
索引	idx_ip(ip)、idx_site_id (site_id)、idx_up_device_ip (up_device_ip)、idx_precinct_id (precinct_id)		
分区	[检查点：数据量性能及扩展性考虑]*分区的算法，分区的字段及其顺序*		
分桶	*分桶的算法，分桶的字段及其顺序，只适用于 mpp*		
序列	N/A		

［数据库设计采用数据库设计模板，并编写成单独文档。在这里请说明引用的数据库设计文档。

此节从本系统或子系统的角度来列举与本高层设计有关的数据库对象变更等。］

2.9 非功能设计

2.9.1 指标

按照当前系统的规模和容量，估算并承诺一个季度内的设计目标

非功能指标分类	非功能指标项	指标值	补充说明
安全	是否涉及外部接口		*是否涉及外部接口*
	是否包含身份认证/报文防篡改		*外部接口必须有 appID/appKey 身份认证，外部接口必须有报文的散列值防篡改*
	是否包含铭感数据		*外部接口涉及敏感数据必须用 https 协议*
性能	接口 TPS(每秒)		
	接口响应时间（毫秒）		
	风险规模考虑	*是否需要水平/垂直拆涉及存储的表，是否需要换存储*	
	高性能高并发大数据量读写场景	**检查点：如不涉及填不涉及**	*是否有识别高性能高并发大数据量读写场景，并且对该场景进行专项设计，并且就设计技术合理性、工作量、设计内容均具备可落地性*
	高可用场景考虑	**检查点：如不涉及填不涉及**	*服务是否具备高可用、弹性扩展能力，是否更多地以集群热备为主而非冷备，同时在服务在高可用环境下是否存在多程序运行冲突*

2.9.2 第三方引用

[描述各模块所引用的第 3 方库、工具等，包括库、工具的名称、开发者/商、版本等信息，并描述其引用方式、接口等。]

2.9.3 网络资源需求梳理

网络资源规划是否涵盖，是否合理，涉及的对外网络开通策略需要给出，并且需要满足正确性、可行性、完整性

【检查点】如不涉及对接接口，且无需额外网络策略写不涉及；如需开通网络策略，参考下表补充网络策略信息

源地址	源端口	源地址所属业务系统	目的 IP 地址	目的端口	目的地址所属业务系统	协议（TCP，UDP）	动作（开通，关闭）	开通策略的用途
10.242.135.0/24（原有） 10.242.238.0/24（新增）	ALL	网络线 4A	188.0.211.40	8080	卓望动环工作台	TCP	开通	4A 图形堡垒机访问资源

2.9.4 安全性

2.9.4.1 公司安全开发规范满足度

是否满足公司安全开发规范要求

原则上必须满足公司安全开发规范要求，如因可观因素暂不满足或需延迟满足的，说明不满足的点、不满足原因、可预期风险及应对措施、后续满足计划

2.9.4.2 垂直越权考虑及实现

垂直越权考虑及实现是否完成

以下为动环垂直越权处理方案

1. 新增 ResAction 注解类

```
@Documented
@Retention(RetentionPolicy.RUNTIME)
@Target(ElementType.METHOD)
public @interface ResAction {
  String resType();
  String action();
```

```
  boolean loadResFilter() default false;
}
```

2. 在 ControllerAuthAspect 增加 ResAction 解析

在 ControllerAuthAspect 类的 initCurrRequestUserAuth方法的最后增加以下代码块

```
  // 解析方法的注解 ResAction,填充  resourceType 和 action
final MethodSignature signature = (MethodSignature) joinPoint.getSignature
();
final Method method = signature.getMethod();
final ResAction anno = method.getAnnotation(ResAction.class);
if (anno != null) {
authCtx.setResType(anno.resType());
authCtx.setRawAction(anno.action());
  // 在 controller 嵌套调用时，只要有@ResAction 注解中配置了需要加载资源，则该标记适
用于所有 controller
if (anno.loadResFilter()) {
authCtx.setLoadResFilter(anno.loadResFilter());
  }
}
```

3. 在 Controller 层增加注解

在所有的 Controller 类的方法上，增加 ResAction 注解

```
@ResAction(resType="cmdb", action="view")
public InstanceVO resActionTest(@RequestParams("id") String resId) {
  // todo
  return instanceService.getInstanceById(resId);
}
```

4. 前端 VUE 集成，控制按钮不显示

前端通过获取 session 中存放的权限列表，进行遍历判断，关键代码如下

```
<button id = "cmdbCreateBtnDisplay" v-if="cmdbCreateBtnDisplay">资源新增</bu
tton>
let permissions = sessionStorage.getItem('permissions');
permissions && permissions.forEach((item) => {
  if (item.indexOf("cmdb:create") > -1) {
  this.cmdbCreateBtnDisplay = true
  }
})
```

5、ResAction 注解中 resType 定义原则

涉及数据库表 resource_schema，要遵守如下原则：

1、一级菜单（模块）：resource 遵守驼峰命名规则，不允许出现下划线“_”，parent_resource 为 all；

2、其它级别的菜单（除一级菜单外）：resource 的定义规则为，父级菜单编码

（parent_resource）_XXXXX，其中 XXXXX 遵守驼峰命名规则，不允许出现下划线“_”。菜单编码规则样例如下：

级别	菜单编码（resource）	菜单名称（name）	父级菜单（parent_resource）
一级菜单	alarm	告警	all
二级菜单	alarm_filter	筛选器	alarm
三级菜单	alarm_filter_view	视图	alarm_filter
三级菜单	alarm_filter_config	设置	alarm_filter
三级菜单	alarm_filter_analysis	分析	alarm_filter

6、ResAction 注解中 action 定义原则

涉及数据库表 resource_schema_actions，要遵守如下原则，{resource}:{action}，{resource}为 resource_schema 最后层级的 resource，aciont 遵守如下原则，如（alarm_filter_analysis:create）：

1、create：新增；
2、update：修改
3、view：查看
4、delete：删除
5、upload：上传
6、download：下载
7、import：导入
8、export：导出

2.9.4.3 水平越权考虑及实现

水平越权考虑及实现是否完成

水平越权主要是指用户可看的数据权限，包括资源权限和站点权限的数据控制。

1、封装层根据用户取得资源权限和站点权限并组装为查询参数

```
public void setAuth(EnergyComplianceRequest request){
    RequestAuthContext authCtx = RequestAuthContext.currentRequestAuthContext();
    RbacResource rbacData = new RbacResource();
    rbacData.setResTypeAction(authCtx.getResAction());
    String userName = authCtx.getUser().getUsername();

    //获取资源权限并组装为查询参数
    if (!authCtx.getUser().isSuperUser()) {
        Map<String, Set<String>> totalConstraints =
resHelper.dataConstraints(authCtx.getUser());
        if (null != totalConstraints) {
            Set<String> precintIdList = totalConstraints.get(ResourceAuthHelper.AREA);
            Set<String> authContainsSubList =
totalConstraints.get(ResourceAuthHelper.AREA_LIKE);
            if (!CollectionUtils.isEmpty(precintIdList)) {
                request.setAuthPrecinctList(new ArrayList<>(precintIdList));
            }
```

```
            if (!CollectionUtils.isEmpty(authContainsSubList)) {
                request.setAuthContainsSubList(new ArrayList<>(authContainsSubList));
            }
        }
    }
    if(!StringUtils.isEmpty(request.getPrecinctId())){
        List<String> precinctIdList = Arrays.asList(request.getPrecinctId().split(","));
        request.setPrecinctIdList(precinctIdList);
    }

    // 获取站点类型权限并组装为参数
    List<String> siteTypesPermission =
roleServiceClient.listUserSiteType(authCtx.getUser().getNamespace(),
authCtx.getUser().getUsername()).stream().map(UserSiteTypeDto::getSiteType).collect(Colle
ctors.toList());
    String siteType = request.getSiteType();
    if (StringUtils.isBlank(siteType)) {
        request.setSiteType(String.join(",", siteTypesPermission));
        request.setSiteTypeList(siteTypesPermission);
    } else {
        List<String> selectType = Arrays.asList(siteType.split(","));
        List<String> intersection =
siteTypesPermission.stream().filter(selectType::contains).collect(Collectors.toList());
        request.setSiteType(String.join(",", intersection));
        request.setSiteTypeList(intersection);
    }
}
```

2、Composite 层的 controller 方法中调用上面的方法初始化权限

```
@Override
@ResAction(resType = "energy_save_electricity_manager_estimate", action = "view")
public BaseResponse<EnergyComplianceDto> getAnalysisProgressRate(@RequestBody
EnergyComplianceRequest request) {
    //初始化权限
    setAuth(request);
    return energyComplianceClient.getAnalysisProgressRate(request);
}
```

3、Mapper 中 sql 加入数据权限查询条件，需要根据 sql 定义的表别名做相应调整

```
<!-- 数据权限 -->
<sql id="baseAuthWhereClause">
    <if test="authPrecinctList != null and authPrecinctList.size > 0" >
        and ( t1.site_id in (
        <foreach collection="authPrecinctList" item="item" index="index" separator=",">
```

```
            #{item, jdbcType=VARCHAR}
        </foreach>
        )
        <if test="authContainsSubList != null and authContainsSubList.size > 0" >
            or <foreach collection="authContainsSubList" item="item" separator=" or " open="(" close=")" >
            t1.site_id like concat(#{item},'%')
        </foreach>
        </if>
        )
    </if>
    <if test="authPrecinctList == null or authPrecinctList.size == 0" >
        <if test="authContainsSubList != null and authContainsSubList.size > 0" >
            and <foreach collection="authContainsSubList" item="item" separator=" or " open="(" close=")" >
            t1.site_id like concat(#{item},'%')
        </foreach>
        </if>
    </if>
    <if test="siteTypeList != null and siteTypeList.size > 0">
      and d.site_type in
      <foreach collection="siteTypeList" index="index" item="siteType" open="("
            separator="," close=")">
       #{siteType}
      </foreach>
      </if>
</sql>
```

2.9.4.4 敏感数据加密考虑及实现

敏感数据加密考虑及实现是否完成，如不涉及的写不涉及

动环敏感信息主要涉及到用户账号、姓名、手机号码、邮箱地址等，涉及到敏感信息，必须使用 base64 进行加密在传输到前端，前端做相应的解密。

1、java 后端 base64 加密，在涉及敏感信息的 DTO 增加加密方法，把对应字段进行加密

```
/**
* Base64 编码用户敏感信息脱敏
*
*/
public void encodeProperty() {
    //用户名加密
    if (StringUtils.isNotBlank(this.getName())){
        this.setName(Base64.encodeBase64String(this.getName().getBytes(StandardCharsets.UTF_8)));
    }
     //手机号码加密
    if (StringUtils.isNotBlank(this.getMobile())){
        this.setMobile(Base64.encodeBase64String(this.getMobile().getBytes()));
    }
```

```
    //办公号码加密
    if (StringUtils.isNotBlank(this.getPhone())){
        this.setPhone(Base64.encodeBase64String(this.getPhone().getBytes()));
    }
    //邮箱地址加密
    if (StringUtils.isNotBlank(this.getMail())){
        this.setMail(Base64.encodeBase64String(this.getMail().getBytes()));
    }
    //账号加密
    if (StringUtils.isNotBlank(this.getLdapId())){
        this.setLdapId(Base64.encodeBase64String(this.getLdapId().getBytes()));
    }
}
```

2.9.4.5 重放攻击考虑及实现

重放攻击考虑及实现是否完成

重复攻击就是防止重复提交，动环已经通过注解+AOP 的方案实现了防止重复提交的功能。只需要在 composite 层的 controller 方法中，添加注解@NonRepeatSubmit 即可。

```
@Override
@NonRepeatSubmit
@ResAction(action = "update", resType = "site_manage")
public BaseResponse modifySiteMappingName(@RequestBody SiteManageRequest request) {
    Map<String, String> userInfo = getUserInfo();
    request.setUserId(userInfo.get("userId"));
    request.setUserName(userInfo.get("userName"));
    return siteManageServiceClient.modifySiteMappingName(request);
}
```

2.9.4.6 应用、中间件配置及认证安全

应用、中间件配置及认证安全是否完成实现

开发中遵循以下原则：

1、Mysql、kafka、redis、elasticsearch、ftp 使用统一的配置文件，禁止在应用中添加这些配置，如需添加这些配置，只需要引入对应的公用配置文件即可。

application-mysql.yml：为 mysql 配置

application-redis.yml：为 redis 配置

application-elasticsearch.yml：为 es 配置

application-kafka.yml：为 kafka 配置

application-ftp.yml：为 ftp 配置

2、禁止在代码中出现 IP、端口、账号、密码等涉及安全的敏感信息，可通过配置项的方式实现。

2.9.4.7 文件上传下载安全

文件上传下载安全是否完成实现

1、关于上传，动环在 composite 层，通过统一的过滤器限制上传的文件类型。目前支持的后缀包含 doc,docx,xls,xlsx,ppt,pptx,pdf,txt,rar,zip,jpg,jpeg,gif,png,mp4,flv,avi,rm,rmvb,wmv。如需增加文件类型，只需要在 ms-composite-service.yml 配置文件中的 availableSuffixes 配置项增加即可。

```
availableSuffixes: doc,docx,xls,xlsx,ppt,pptx,pdf,txt,rar,zip,jpg,jpeg,gif,png,mp4,flv,avi,rm,rmvb,wmv
```

2、关于文件下载，动环通过 nginx 的 auth 模块功能，实现了下载限制，下载的 URL 必须遵守如下原则：^~/spider/web/download/。

2.9.4.8 未经授权访问 API 防护

未经授权访问 API 防护是否完成实现

1、无需授权的 API，必须通过白名单配置进行实现，只需要在 composite 服务的 ms-composite-service.yml 配置文件中添加白名单即可。

```
aspire.webbas.component.safeguarding:
  illegalityParam: <,>,</script,<script,script[\s],<img,%27,and[\s],select[\s],insert[\s],exec[\s],update[\s],drc
  whiteList:  #白名单，List集合
  - /v1/bulletin/saveSendNotice
  - /v1/scada/saveScadaCanvas/**
  - /v1/roles/alauda/getRoleByUserName/alauda
  - /v1/configManagement/update**
  - /v1/alerts/createColumnInfo**
  - /v1/alerts/getCurrVal
  - /xxl-static/**
  - /xxl/**
  availableSuffixes: doc,docx,xls,xlsx,ppt,pptx,pdf,txt,rar,zip,jpg,jpeg,gif,png,mp4,flv,avi,rm,rmvb,wmv
```

2.9.4.9 垂直越权

范围：新增接口

要求：对于新增的接口，需要做好垂直越权管理，规避“低权限身份可访问高权限数据以及功能”的风险

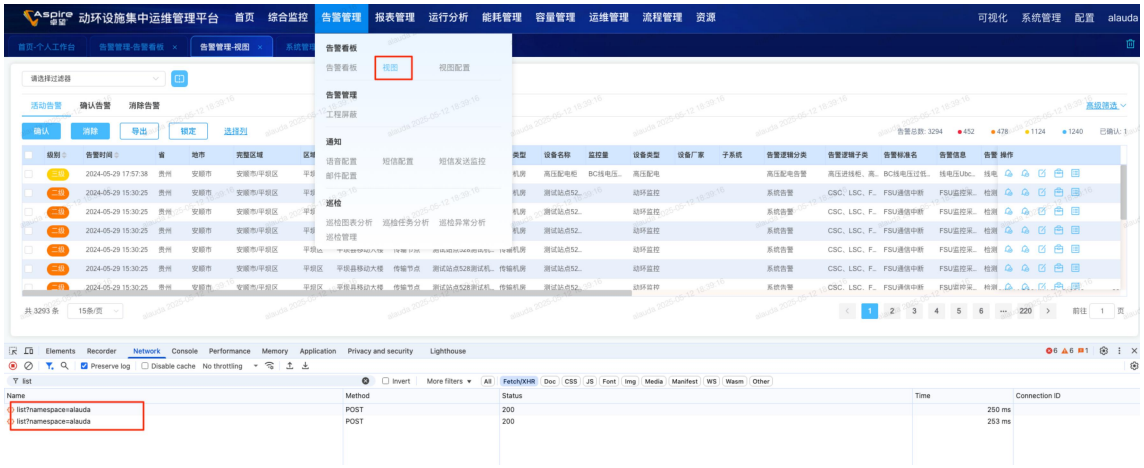
操作：

（1）保证“不鉴权”开关处于“禁用”状态。否则必定存在垂直越权风险。

（2）明确新增接口归属的资源，然后拿到对应的资源 ID

比如某新增“活动告警列表”接口，url 是 /v1/alerts/list，

这个接口需要绑定在 “告警管理-告警看板-视图”下面。那就得拿到“视图”这个资源的 ID（可能通过 wb_resources 表进行查询）



（3）将新增接口绑定到具体资源下：

（4）新增完“权限依赖”后，就可以验证了。
验证方式：使用低权限 cookie 访问这个接口，如果接口状态码返回 403，表示权限鉴权生效。

2.9.4.10 水平越权

新增接口时，要规避水平越权风险，做好相关鉴权。
水平越权：用户张三只有 A 市数据访问权限，但访问到了 B 市范围的数据。

2.9.5 扩展性

【检查点】程序是否具备未来功能、非功能场景的扩展能力，是否在程序中存在硬编码配置，是否在功能改造时需要“大动干戈”或“无从下手”

2.9.6 风险

如当前设计不包含、暂未考虑、产品未说明清楚/未规划等可能引发问题的内容

3 系统可部署性设计（可选）

［说明为了简化部署环节，缩短部署时间而在程序内部设计中作出的安排。］

4 系统可维护性设计

［说明为了系统维护的方便而在程序内部设计中作出的安排，包括在程序中专门安排用于系统的检查与维护的检测点、日志、屏幕输出、专用模块、工具等。］