危险化学品企业双重预防机制数据交换规范接口技术文档

V1.3

2024 年 5 月

修订记录

日 期	版本	修订章节	修订 内容	作者	
2022-06-29	1.0		创建	吕	
2022-11-17	1.1		修改	吕	
2023-06-15	1.2		修改	邬	
2024-04-30	1.3		修改，新增章节3.7、3.9、4.4、4.5，修改4.3章节，原3.7章节修改为3.8章节	邬	

目录

一、 概述

1.1 说明

依据《危险化学品企业双重预防机制数字化建设数据交换规范》，使用 RESTful 接口接收企业的风险单元、风险事件、管控措施、隐患信息、隐患排查任务、隐患排查记录等数据。

1.2 预期读者

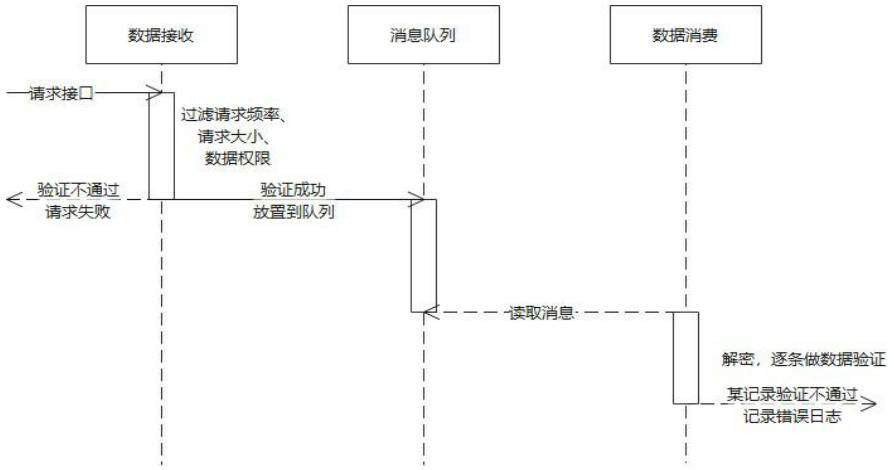
自建《危险化学品企业双重预防机制数字化系统》的危险化学品企业、政府及其开发单位。企业需要满足向《危险化学品企业双重预防机制数字化建设数据交换规范》上报数据的要求。

二、 设计概述

考虑到对接的单位较多，同时尽量将有效的数据存留下来，接收端的服务分为两个：数据接收与数据消费。数据接收主要负责提供接收数据的接口，并将通过初步过滤的数据放置到消息队列中；数据消费主要负责消费放置在消息队列里的数据，解析后进行二次逐条过滤。

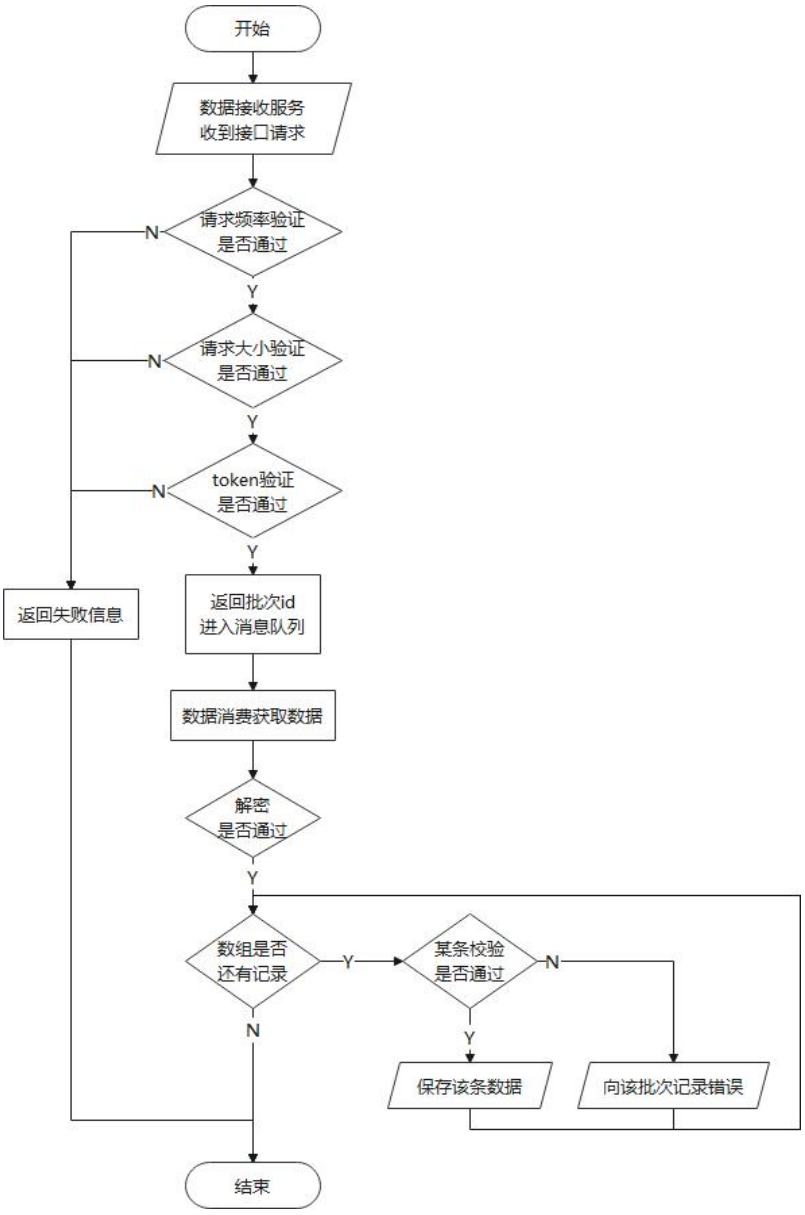
2.1 处理流程

下面是处理流程的时序图：



当收到接口请求后，数据接收先对请求的频率、请求的大小、数据权限进行验证，验证不通过的请求直接请求失败，不进一步解析请求体内的数据。当上述验证都通过之后，将会返回一个用于查询消费进度的批次 id，同时将请求内容放置到消息队列中，由订阅了消息队列的数据消费程序对内容进行解密并逐条做数据验证。

数据消费程序先解密请求体，解密失败等同于请求失败，不进一步解析数据。解密成功进行逐条数据验证时，某条验证通过将会保存，某条验证失败将会记录到错误日志中，不会影响验证成功的记录正常保存，仅仅是验证失败的那一条会失败。通过使用查询相关接口，可以查询到批次处理结果。用流程图的方式表达见下图：



2.2 请求规则

2.2.1 请求频率限制

对于上报接口的请求频率上，一个 token 在限定时间内请求次数过多将会返回 HTTP 状态码 403。

注：每分钟限定30次。

2.2.2 请求大小限制

对于上报接口，请求的大小要在15M 以内。超过将会返回HTTP 状态码413。

2.2.3 token 校验规则

token由appid和secret组成，采用AES加密方式加密，再用Base64对内容进行加密（加密 Java 示例代码见第七节），AES 加密的Key和IV请联系技术人员获取。示例如下：

明文示例：

{"appid":"u64dc7b32","secret":"s93261245c05ee3bc"}

加密后示例：

9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

接口的定义中，会看到 HTTP Header 中需要传 Authorization 作为身份认证信息。将加密后的字符串作为 HTTP Header 中的 Authorization 的值。

无法正常解析 token 的或经解析发现 token 无效的，返回 HTTP 状态码 401。

2.3 请求体规则

2.3.1 加密规则

上报接口中的请求体需采用 AES 加密方式，AES 加密的 Key 和 IV 请联系技术人员获取。请求时，先用 AES 加密，再用 Base64 对请求体的 data 内容进行加密（加密 Java 示例代码见第七节），最终效果为{"data":"加密字符串"}。对于无法正常解密的，记录错误日志。

2.3.2 参数校验规则

以下规则在参数校验时不通过，将会记录到错误记录中，可通过查询接口查看错误记录：

必填项为空时；

字典项不符合字典枚举时；

日期、时间不符合格式时；

字符超长时；

id 不符合 uuid 正则（xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx 或 32 位字符串）时；

companyCode 不在 token 限定的数据范围之内。

2.4 查看上报结果

第三章的上报接口中，返回结果里有批次 id，通过该 id 调用第四章的查询接口，可查询到当前批次 id 的处理结果。

三、 上报接口

注意：// xxx 表示对该字段的说明解释性文字，实际发送数据时不能加这些内容。

3.1 上报安全风险单元数据

接口路径	/taskSys/busi/syf/receiveUnitList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id": "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",// 唯一编码， 36 位 uuid，必填
	"companyCode": "440000000",// 企业编码，同“危险化学品登记信息

	管理系统”中的企业编码一致，必填 "hazardCode": "440000000008",//风险分析对象编码，同“危险化学品登记信息管理系统”中的危险源编码一致，必填 "riskUnitName": "反应器",// 风险分析单元名称，必填 "hazardDep": "分离加氢部",//责任部门，必填 "hazardLiablePerson": "李小强",//责任部门负责人姓名，必填 "deleted": "0",// 删除状态，字典含义见数据字典 deleted "createDate": "2020-11-28 11:44:58",// 创建时间，必填，时间格式 yyyy-MM-dd HH:mm:ss "createBy": "某化工企业管理员",// 创建人姓名，必填 "updateDate": "2020-11-28 11:44:58",// 修改时间，用于增量同步，必填，时间格式 yyyy-MM-dd HH:mm:ss "updateBy": "王红"// 修改人姓名，必填 }] }

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次id
	}
	}

3.2 上报安全风险事件数据

接口路径	/taskSys/busi/syf/receiveEventList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6flVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPilZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id": "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",// 唯一编码， 36 位

	uuid，必填 "companyCode": "440000000",// 企业编码，同“危险化学品登记信息管理系统”中的企业编码一致，必填 "riskUnitId": "6a183cdd-d1de-a733-7c13-dfd2c26db26d",// 风险分析单元 ID，必填 "riskEventName": "储罐泄漏造成火灾爆炸",// 风险事件名称，必填 "deleted": "0",// 删除状态，字典含义见数据字典deleted "createDate": "2020-11-28 11:44:58",// 创建时间，必填，时间格式 yyyy-MM-dd HH:mm:ss "createBy": "某化工企业管理员",// 创建人姓名，必填 "updateDate": "2020-11-28 11:44:58",// 修改时间，用于增量同步，必填，时间格式 yyyy-MM-dd HH:mm:ss "updateBy": "王红"// 修改人姓名，必填 }] }

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次id
	}
	}

3.3 上报安全风险管控措施数据

接口路径	/taskSys/busi/syf/receiveControlMeasuresList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6flVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPilZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id": "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",// 唯一编码， 36 位

	uuid，必填 "companyCode": "440000000",// 企业编码，同“危险化学品登记信息管理系统”中的企业编码一致，必填 "riskEventId": "41505423-5dc1-d0f9-c2a9-9d95507e392c",// 风险事件ID，必填 "dataSrc": "2",// 管控方式，字典含义见数据字典 dataSrc "riskMeasureDesc": "定期进行设备检修、报检",// 管控措施描述 "troubleshootContent": "1、静设备是否定期检修、报检。",// 隐患排查内容，必填 "classify1": "2",// 管控措施分类 1，必填，字典含义见数据字典classify1 "classify2": "2-2",// 管控措施分类 2，必填，字典含义见数据字典classify2 "classify3": "环保设施",// 管控措施分类 3 "deleted": "0",// 删除状态，字典含义见数据字典deleted "createDate": "2020-11-28 11:44:58",// 创建时间，必填，时间格式yyyy-MM-dd HH:mm:ss "createBy": "某化工企业管理员",// 创建人姓名，必填 "updateDate": "2020-11-28 11:44:58",// 修改时间，用于增量同步，必填，时间格式 yyyy-MM-dd HH:mm:ss "updateBy": "王红"// 修改人姓名，必填 }] }

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次id
	}
	}

3.4 上报隐患排查任务数据

接口路径	/taskSys/busi/syf/receiveMeasuresTaskList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id": "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",// 唯一编码， 36 位 uuid，必填
	"companyCode": "440000000",// 企业编码，同“危险化学品登记信息管理系统”中的企业编码一致，必填
	"riskMeasureId": "3bcf96f3-9a80-0de0-b56b-9563a693455a",// 管控措施 ID，必填
	"troubleshootContent": "储罐泄漏造成火灾爆炸",// 隐患排查内容，必填
	"checkCycle": 4,// 巡检周期，必填，大于 0 的正整数
	"checkCycleUnit": "小时",// 巡检周期单位，字典含义见数据字典 checkCycleUnit
	"taskStartTime": "2020-11-28 11:44:58",// 任务开始时间，指该任务首 次开始执行的时间，必填，时间格式 yyyy-MM-dd HH:mm:ss
	"workStartTime": "8:00:00",// 工作开始时间，指每日该任务开始执行的时间，当巡检周期是小时的时候为必填，时间格式例如：8:00:00
	"workEndTime": "8:00:00",// 工作结束时间，指每日该任务结束执行的时间，当巡检周期是小时的时候为必填，时间格式例如：8:00:00
	"workDayType": "0",// 工作日类型（每天：0；法定工作日（我国职工的标准工作日，即国家机关、事业单位实行的统一工作时间）：1；非 法定工作日（法定节假日及非调休周末，即除法定工作日外的时间）：2）
	"workType": "0",// 任务类型（日常任务：0；主要负责人任务：1；技术负责人任务：2；操作负责人任务：3；）默认为日常任务：0，必填
	"taskNum": "1"// 包保任务对应项，当任务类型为主要负责人任务、技术负责人任务、操作负责人任务时，此项为必填。
	"deleted": "0",// 删除状态，字典含义见数据字典deleted

	"createDate": "2020-11-28 11:44:58",// 创建时间，必填，时间格式 yyyy-MM-dd HH:mm:ss "createBy": "某化工企业管理员",// 创建人姓名，必填 "updateDate": "2020-11-28 11:44:58",// 修改时间，用于增量同步，必填，时间格式 yyyy-MM-dd HH:mm:ss "updateBy": "王红",// 修改人姓名，必填 }] }

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次id
	}
	}

3.5 上报隐患排查记录

接口路径	/taskSys/busi/syf/receiveMeasuresTaskRecordList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6flVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPilZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id": "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",// 唯一编码， 36 位 uuid，必填
	"companyCode": "440000000",// 企业编码，同“危险化学品登记信息管理系统”中的企业编码一致，必填
	"checkTaskId": "4bae2b00-9e5e-3fea-7722-3c66917a8c82",// 隐患排查任务 ID
	"checkTime": "2020-11-20 10:13:35",// 排查时间，必填，时间格式 yyyy-MM-dd HH:mm:ss
	"mobileMe": "868993065120435",// 手机识别码 IMEI，必填，说明： 对于版本过高、技术架构老等无法自动获取 imei 码的情况，可采取人工 查询、手动绑定本机 imei 码的方式，且务必保证 imei 码的真实性。
	"isDefend": "1"// 是否包保责任人任务标识（是：1；否：0）， 必填
	"createBy": "某化工企业管理员",// 创建人姓名，必填
	"createByMobile": "13512341234",// 创建人手机号，必填

	}] }

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次id
	}
	}

3.6 上报隐患信息数据

接口路径	/taskSys/busi/syf/receiveDangerInvestigationList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id": "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",// 唯一编码， 36 位 uuid，必填
	"companyCode": "440000000",// 企业编码，同“危险化学品登记信息管理系统”中的企业编码一致，必填
	"hazardCode": "440000000008",// 危险源编码，同“危险化学品登记信息管理系统”中的危险源编码一致，必填
	"riskMeasureId": "3bcf96f3-9a80-0de0-b56b-9563a693455a",// 管控措施 ID
	"checkRecordId": "516d8193-f2e3-55e6-4b48-afce5da37677",// 隐患排查记录 ID
	"dangerName": "液位计红线模糊",// 隐患名称，必填
	"dangerLevel": "0",// 隐患等级，字典含义见数据字典 dangerLevel，

	必填 "dangerSrc": "1",// 隐患来源，字典含义见数据字典 dangerSrc，必填 "dangerManageType": "1",// 隐患治理类型，字典含义见数据字典 dangerManageType，必填 "dangerDesc": "液位计红线模糊",// 隐患描述，必填 "dangerReason": "油漆老化",// 隐患产生原因 "controlMeasures": "控制措施",// 控制措施 "cost": "10",// 资金，单位万元 "dangerManageDeadline": "2020-09-15 00:00:00",// 隐患治理期限，必填，时间格式 yyyy-MM-dd HH:mm:ss "dangerState": "1",// 隐患状态， 必填， 字典含义见数据字典 dangerState "hazardDangerType": "1",// 隐患类型，必填，字典含义见数据字典 hazardDangerType "hazardCategory": 0// 隐患类别（其他隐患：0 主要负责人隐患：1 技术负责人隐患：2 操作负责人隐患：3）默认为其他隐患：0 ，必填 "registrant": "张三",// 登记人姓名，必填 "registTime": "2020-08-27 00:00:00",// 登记时间，必填，时间格式 yyyy-MM-dd HH:mm:ss "liablePerson": "李小强",// 整改责任人姓名，必填 "checkAcceptPerson": "李小强",// 验收人，该项只在隐患状态为已验收时必填 "checkAcceptComment": "验收情况",// 验收情况 "checkAcceptTime": "2020-11-27 10:29:31",// 验收时间，该项只在隐患状态为已验收时必填，时间格式 yyyy-MM-dd HH:mm:ss "deleted": "0",// 删除状态，字典含义见数据字典 deleted "createDate": "2020-11-28 11:44:58",// 创建时间，必填，时间格式 yyyy-MM-dd HH:mm:ss "createBy": "某化工企业管理员",// 创建人姓名，必填 "updateDate": "2020-11-28 11:44:58",// 修改时间，用于增量同步，必填，时间格式 yyyy-MM-dd HH:mm:ss

	"updateBy": "王红"// 修改人姓名，必填 }] }

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次id
	}
	}

3.7 上报专项检查隐患信息

接口路径	/taskSys/busi/syf/receiveCheckDangerList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPilZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id":"4e403ba4-53bd-4d6c-9d57-f83bb01bdddd",//唯一编码,36 位 uuid 必填
	"companyCode":"220180301",//企业编码，同“危险化学品登记综合服务系统”中的企业编码一致，必填
	"dangerld":"4e403ba4-53bd-4d6c-9d57-f83bb01bdddd",//隐患主键，必填
	"taskId":"bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b"，//检查任务主键ID，章节4.3 查询的检查任务 ID，必填
	"checkType":"1",//检查类型(企业自查:1:市级及以下检查:2;省级检查:3;部级检查:4)，必填
	"checkltemId":"da4612c8-0e73-4839-a0fb-bff79c23f29b".//检查项主键ID，章节4.4查询的检查项ID，必填
	"checkScoreld":"be942b6d-aff7-46d3-b3ce-ed6d13dfa044"，//检查项评分细则主键ID，章节4.5 查询的评分细则ID，必填
	"checkDate":"2023-05-15",//检查时间，yyyy-MM-dd，必填
	"checkPay":"2000",//罚款金额，单位:元
	"harmType":"1",//危害类型，事故隐患:1;火灾隐患:2;当任务类型为“危险化学品重大危险源企业安全专项检查督导”时此项为必填项
	"deleted":"0".// 删除状态，字典含义见数据字典deleted，必填
	"createBy":"sys"， //创建人，必填
	"createDate":"2023-05-15 15:00:00",//创建时间，必填
	"updateBy":"sys",//最后修改人，必填
	"updateDate":"2023-05-15 15:05:00"//最后修改时间，必填
	}]
	}

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次 id
	}
	}

3.8 上报停用/检修记录

接口路径	/taskSys/busi/syf/receiveUnitStopList
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id": "4e403ba4-53bd-4d6c-9d57-f83bb01bdddd", //主键 id，必填
	"companyCode": "220180301", //企业编码，,必填
	"hazardCode": "220180301001", // 风险分析对象编码，即危险化学品登记信息管理系统中的危险源编码，必填
	"riskUnitId": "4e403ba4-53bd-4d6c-9d57-f83bb01pnddd", // 所属风险单元 ID，必填
	"stopStartTime": "2023-05-15 10:35:00", //装置停用或者检维修开始时间，必填
	"stopEndTime": "2023-05-15 10:35:00", //装置停用或者检维修结束时间
	"stopReason": "停工检修", //描述停用原因，必填
	"createBy": "sys", //创建人，必填
	"createDate": "2023-05-15 15:00:00", //创建时间，必填
	"updateBy": "sys", //最后修改人，必填
	"updateDate": "2023-05-15 15:05:00" //最后修改时间，必填
	}]
	}

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次 id
	}
	}

3.9 上报检查情况记录

接口路径	/taskSys/busi/syf/receiveCheckRecordList

请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

明文格式举例与解释	{
	"data": [{
	"id":"4e403ba4-53bd-4d6c-9d57-f83bb01bdddd",//唯一编码,36 位 uuid.
	必填
	"companyCode":"220180301",//企业编码，同“危险化学品登记综合服务系统”中的企业编码一致，必填
	"hazardCode":"220180301001",// 危险化学品登记综合服务系统中的危险源编码;当任务类型为“危险化学品重大危险源企业安全专项检查督导”时此项为必填项
	"taskId":"bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b".//检查任务主键ID，章节 4.3 查询的检查任务 ID，必填
	"checkType":"1",//检查类型(企业自查:1;市级及以下检查:2;省级检查:3;部级检查:4)，必填
	"checkDate":"2023-05-15",//检查日期，yyyy-MM-dd，必填
	"resultStatus":"0",//是否停产整顿，正常:0，停产整顿:1
	"createBy":"sys"， //创建人，必填
	"createDate":"2023-05-15 15:00:00",//创建时间，必填
	"updateBy":"sys",//最后修改人，必填
	"updateDate":"2023-05-15 15:05:00"//最后修改时间，必填
	}]
	}

返回格式举例与解释	{
	"obj": {
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9"// 新增/修改的批次 id
	}
	}

四、 查询接口

注意：// xxx 表示对该字段的说明解释性文字，实际发送数据时不能加这些内容。

4.1 按批次 id 查询处理结果

接口路径	/taskSys/busi/syf/getLog
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6flVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

报文格式举例与解释	{
	"batchId": ["efdff4eb449f4a0c9f96ad1cd97cd0e9"]// 批次 ID 列表，必填
	}

返回格式举例与解释	{
	"obj": [{
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9",// 批 次 ID
	"status": "1",// 批次处理状态，字典含义见数据字典 batchStatus
	"statistics": {// 统计情况，若批次在处理中，数值会不断递增
	"totalCount": 100,// 批次总条数，和 data 数组大小一致
	"errorCount": 10// 该批次错误条数

	}, "errorIds": ["bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b", "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9c"]// 有错误的 ID 列表 }] }

4.2 按错误 id 查询错误详细信息

接口路径	/taskSys/busi/syf/getRequestLog
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

报文格式举例与解释	{
	"errorIds": [
	"bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",
	"bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9c"
	]// 有错误的 ID 列表
	}

返回格式举例与解释	{
	"obj": [{
	"id": "bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",// 有 错 误 的 ID
	"batchId": "efdff4eb449f4a0c9f96ad1cd97cd0e9",// 所属批次 ID
	"companyCode": "440000000",// 有错误的企业编码"requestType":
	"1",// 请求类型，字典含义见数据字典 requestType "receiveData":
	"xxx",// 接收的内容
	"failReason": "xxx 不能为空"// 失败原因，多个原因用换行符隔开
	"resultTime": "2020-11-28 11:44:58",// 接收结果产生时间，时间格式
	yyyy-MM-dd HH:mm:ss

	}] }

4.3 查询专项检查任务信息

接口路径	/taskSys/busi/syf/getCheckTask
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhElL1BOHU5F1ov2F6flVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPilZkPAzrpp

报文格式举例与解释	{
	"companyCodes": [
	"220180301",
	"220180302"
	]// 企业编码列表，必填，每次最多查询10家企业
	}

返回格式举例与解释	{
	"obj": [{
	"id":"bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b".//检查任务ID
	"companyCode":“440000000",// 企业编码，同“危险化学品登记综合服务系统”中的企业编码一致
	"taskName":"2024 危险化学品危险源企业安全专项检查督导",// 任务名称
	"taskStartTime":"2022-11-28"，//任务开始日期，日期格式yyyy-MM-dd
	"taskEndTime":"2023-11-28"，//任务结束日期，日期格式

	yyyy-MM-dd "taskTypeName":"危险化学品重大危险源企业安全专项检查督导",//区分检查任务 "taskContent":"危险化学品重大危险源企业2024年专项督查核查",//描述任务内容 "deleted":"0",//删除状态，字典含义见数据字典 deleted "createDate":"2020-11-2811:44:58",//创建时间，时间格式yyyy-MM-dd HH:mm:ss "updateDate":“2020-11-28 11:44:58",//修改时间，时间格式 yyyy-MM-dd HH:mm:ss }] }

注：查询结果为即将开始或正在进行的任务

4.4 查询检查项信息

接口路径	/taskSys/busi/syf/getCheckItem
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhEIL1BOHU5F1ov2F6fIVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPiIZkPAzrpp

报文格式举例与解释	{
	"taskId":"bfcc9012-3ele-440f-9e6c-e9d6b5cc5e9b",//检查任务ID，必填
	"page": {
	"current":1,//分页页码
	"size":100// 分页大小，最小10最大1000
	}
	}

返回格式举例与解释	{
	"total":100,// 总数
	"current":1,// 当前分页
	"size":100,// 分页大小
	"obj": [{
	"id":"da4612c8-0e73-4839-a0fb-bff79c23f29b",//检查项ID
	"taskId":"bfcc9012-3e1e-440f-9e6c-e9d6b5cc5e9b",//检查任务ID
	"checkName":"1.1",//检查序号
	"checkContent":"重大危险源的主要负责人依法经考核合格",//检查内容
	"checkBasis":"《危险化学品企业重大危险源安全包保责任制办法(试行)》(应急厅(2021)12号)",// 检查依据
	"applicablePlace":"储存单元、生产单元"// 适用场所
	"deleted":"0",//删除状态，字典含义见数据字典 deleted
	"createDate":"2020-11-2811:44:58",//创建时间，时间格式yyyy-MM-dd HH:mm:ss
	"updateDate":"2020-11-28 11:44:58"//修改时间，时间格式 yyyy-MM-dd HH:mm:ss
	}]
	}

4.5 查询检查项评分细则信息

接口路径	/taskSys/busi/syf/getCheckScore
请求方式	POST
请求头	Content-Type: application/json;charset=utf8 Authorization: 9nyQ8mhElL1BOHU5F1ov2F6flVIAu5YTNxJT7v5l/Ub/0eqnVcqeLzBE6EpRjw6vEOCTYCZBjrBewPilZkPAzrpp

报文格式举例与解释	{
	"taskId":"bfcc9012-3ele-440f-9e6c-e9d6b5cc5e9b",//检查任务ID，必填
	"checkItemId":"da4612c8-0e73-4839-a0fb-bff79c23f29b".//检查项ID
	"page": {
	"current":1,//分页页码
	"size":100// 分页大小，最小10最大1000
	}
	}

返回格式举例与解释	{
	"total":100,// 总数
	"current":1,// 当前分页
	"size":100,// 分页大小
	"obj": [{
	"id":"be942b6d-aff7-46d3-b3ce-ed6d13dfa044",//评分细则 ID
	"checkItemId":"da4612c8-0e73-4839-a0fb-bf79c23f29b",// 检查项 ID
	"isReject":"0",// 是否为否决项(是:1;否:0)
	"deductPoint":"5",//扣分分值
	"deductIllustrate":"xxx",//扣分说明
	"deleted":"0",// 删除状态，字典含义见数据字典 deleted
	"createDate":"2020-11-2811:44:58"//创建时间，时间格式yyyy-MM-dd HH:mm:ss
	'updateDate":"2020-11-2811:44:58",//修改时间，时间格式yyyy-MM-dd HH:mm:ss
	}]
	}

五、 返回值含义

以下返回的 HTTP 状态码为数据接收立即返回的，由数据接收返回的错误是整个请求完全失败的。

HTTP 状态码	返回消息	说明
200	正确	请求成功，obj 会返回批次 id
400	无法解析的 token	无法正常解析的 token，请求失败
401	无效的 token	token 是无效的，请求失败
403	请求过于频繁	一个 token 在限定时间内请求次数过多，请求失败
413	请求的大小过大	请求的大小限制在 15M 以内，请求失败
425	过时的 token	时间戳在当前时间之前 1 小时开外，请求失败
500	系统错误	系统内部发生错误，请联系系统管理员，请求失败

504	请求超时	请求超时，请求失败

以下错误日志信息为数据消费返回的，由数据消费返回的错误请求是成功的，解析为明文后，逐条进行数据验证的，仅验证不通过的记录是失败的，验证通过 的记录是成功写入的。数据消费返回的错误需要通过调用查询接口得到。

错误日志信息	说明
系统错误	系统内部发生错误，请联系系统管理员
请求超时	请求超时
无法解密	无法根据 Key 和 IV 对请求体进行解密，请求失败
权限错误	请求中的数据权限范围不在 token 限制的数据权限范围之内，请求失败
危险源不存在	危险源不存在，仅该条记录失败
xxx 为空	违反非空约束，xxx 为具体字段名称，仅该条记录失败
xxx 字典项不符合字典枚举	字典值无效，xxx 为具体字段名称，仅该条记录失败
xxx 时间不符合格式	时间格式无效，xxx 为具体字段名称，仅该条记录失败
xxx 长度过长	字段超长，xxx 为具体字段名称，仅该条记录失败
xxx 不符合正则	不符合正则，xxx 为具体字段名称，仅该条记录失败
无效的批次id	查询的批次 id 不存在
查询的时间范围超过 24 小时	查询的时间范围超过 24 小时

六、 数据字典

字典键	字典值	字典类型
0	未删除	deleted
1	已删除	deleted
1	自动化监控	dataSrc
2	隐患排查	dataSrc
3	其他	dataSrc
1	工程技术	classify1
2	维护保养	classify1
3	操作行为	classify1
4	应急设施	classify1
1-1	工艺控制	classify2
1-2	关键设备/部件	classify2
1-3	安全附件	classify2
1-4	安全仪表	classify2
1-5	其它	classify2
2-1	动设备	classify2
2-2	静设备	classify2

2-3	其它	classify2
3-1	人员资质	classify2
3-2	操作记录	classify2
3-3	交接班	classify2
3-4	其它	classify2
4-1	应急设施	classify2
4-2	个体防护	classify2
4-3	消防设施	classify2
4-4	应急预案	classify2
4-5	其它	classify2
0	一般隐患	dangerLevel
1	重大隐患	dangerLevel
1	日常排查	dangerSrc
2	综合性排查	dangerSrc
3	专业性排查	dangerSrc
4	季节性排查	dangerSrc
5	重点时段及节假日前排查	dangerSrc
6	事故类比排查	dangerSrc
7	复产复工前排查	dangerSrc
8	外聘专家诊断式排查	dangerSrc
9	管控措施实效	dangerSrc
10	其他	dangerSrc
0	即查即改	dangerManageType
1	限期整改	dangerManageType
0	整改中	dangerState
1	待验收	dangerState
9	已验收	dangerState
1	安全	hazardDangerType
2	工艺	hazardDangerType
3	电气	hazardDangerType
4	仪表	hazardDangerType

5	消防	hazardDangerType
6	总图	hazardDangerType
7	设备	hazardDangerType
8	其他	hazardDangerType
小时	小时	checkCycleUnit
天	天	checkCycleUnit
月	月	checkCycleUnit
年	年	checkCycleUnit
0	正常	checkStatus
1	风险单元	requestType
2	风险事件	requestType
3	管控措施	requestType
4	隐患信息	requestType
5	隐患排查任务	requestType
6	隐患排查记录	requestType
7	装置停工检修记录	requestType
8	专项检查隐患	requestType
9	检查情况记录	requestType
1	处理中	batchStatus
2	处理完成且成功	batchStatus
3	处理完成且失败	batchStatus

七、 加解密示例与工具

现将 AES-GCM-256 加解密工具类的 Java 代码展示如下：

```
import org.bouncycastle.crypto.DataLengthException;
import org.bouncycastle.crypto.InvalidCipherTextException;
import org.bouncycastle.crypto.engines.AESFastEngine;
import org.bouncycastle.crypto.modes.GCMBlockCipher;
import org.bouncycastle.crypto.params.AEADParameters;
import org.bouncycastle.crypto.params.KeyParameter;

import java.nio.charset.StandardCharsets;
import java.security.SecureRandom;
import java.util.Base64;

/**
 * AES-GCM-256 工具类
 *  加解密方法中已调用 Base64 方法
 */
public class AesGcm256Util {

    private static final SecureRandom SECURE_RANDOM = new
SecureRandom();

    public static final int NONCE_BIT_SIZE = 128;
    public static final int MAC_BIT_SIZE = 128;
    public static final int KEY_BIT_SIZE = 256;

    private AesGcm256Util() {
    }

    /**
     *  创建密钥
     *
     * @return  密钥
     */
    public static byte[] key() {
        byte[] key = new byte[KEY_BIT_SIZE / 8];
        SECURE_RANDOM.nextBytes(key);
        return key;
    }

    /**
     *  创建向量
     *
     * @return  向量
     */
    public static byte[] iv() {
```

```
        byte[] iv = new byte[NONCE_BIT_SIZE / 8];
        SECURE_RANDOM.nextBytes(iv);
        return iv;
    }

    /**
     * 编码
     *
     * @param hexStr  文本
     * @return  字节数组
     */
    public static byte[] hexToByte(String hexStr)
        { int len = hexStr.length();
        byte[] data = new byte[len / 2];
        for (int i = 0; i < len; i += 2) {
            data[i / 2] = (byte) ((Character.digit(hexStr.charAt(i), 16) << 4)
                    + Character.digit(hexStr.charAt(i + 1), 16));
        }
        return data;
    }

    /**
     * 转为十六进制
     *
     * @param data 字节数组
     * @return  转换结果
     */
    public static String toHex(byte[] data)
        { StringBuilder ret = new
        StringBuilder(); for (byte datum : data)
        {
            String hex = Integer.toHexString(datum & 0xFF);
            if (hex.length() == 1) {
                hex = '0' + hex;
            }
            ret.append(hex.toUpperCase());
        }
        return ret.toString();
    }

    /**
     * 加密
     *
     * @param plainText 明文文本
     * @param key           密钥
```

```
     * @param iv             向量
     * @return  加密字符串
     */
    public static String encrypt(String plainText, byte[] key, byte[] iv)
        { String sr;
        try {
            byte[] plainBytes = plainText.getBytes(StandardCharsets.UTF_8);

            GCMBlockCipher cipher = new GCMBlockCipher(new
AESFastEngine());
            AEADParameters parameters =
                    new AEADParameters(new KeyParameter(key),
MAC_BIT_SIZE, iv, null);

            cipher.init(true, parameters);

            byte[] encryptedBytes = new
byte[cipher.getOutputSize(plainBytes.length)];
            int retLen = cipher.processBytes(plainBytes, 0, plainBytes.length,
encryptedBytes, 0);
            cipher.doFinal(encryptedBytes, retLen);
            sr = Base64.getEncoder().encodeToString(encryptedBytes);
        } catch (Exception ex) {
            throw new RuntimeException(ex.getMessage());
        }
        return sr;
    }

    /**
     *  解密
     *
     * @param encryptedText 已加密文本
     * @param key                密钥
     * @param iv                 向量
     * @return  已解密文本
     */
    public static String decrypt(String encryptedText, byte[] key, byte[] iv)
        { String sr;
        try {
            byte[] encryptedBytes = Base64.getDecoder().decode(encryptedText);

            GCMBlockCipher cipher = new GCMBlockCipher(new
AESFastEngine());
            AEADParameters parameters =
```

```
                    new AEADParameters(new KeyParameter(key),
MAC_BIT_SIZE, iv, null);

            cipher.init(false, parameters);
            byte[] plainBytes = new
byte[cipher.getOutputSize(encryptedBytes.length)];
            int retLen = cipher.processBytes
                    (encryptedBytes, 0, encryptedBytes.length, plainBytes, 0);
            cipher.doFinal(plainBytes, retLen);

            sr = new String(plainBytes, StandardCharsets.UTF_8);
        } catch (IllegalArgumentException | IllegalStateException |
                DataLengthException | InvalidCipherTextException ex) {
            throw new RuntimeException(ex.getMessage());
        }
        return sr;
    }
}
```

maven 的引用见下：

```
<dependency>
    <groupId>org.bouncycastle</groupId>
    <artifactId>bcprov-jdk15on</artifactId>
    <version>1.56</version>
</dependency>
```

加密解密 Java 示例代码如下，加解密方法中已调用 Base64 加解密方法：

```
String encryptedStr = AesGcm256Util.encrypt(originData,
                    AesGcm256Util.hexToByte(AES_KEY),
                    AesGcm256Util.hexToByte(AES_IV));
String resultStr = AesGcm256Util.decrypt(encryptedStr,
                    AesGcm256Util.hexToByte(AES_KEY),
                    AesGcm256Util.hexToByte(AES_IV));
```