



中华人民共和国国家标准

GB/T 45393.5—2025

信息技术 建筑信息模型(BIM)软件 第5部分:数据接口

Information technology—Building information model(BIM) software—
Part 5:Data interface

2025-03-28 发布

2025-10-01 实施

国家市场监督管理总局
国家标准化管理委员会 发布

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	1
5 基本要求	2
5.1 接口	2
5.2 信息容器	2
5.3 安全控制	2
6 数据接口	3
6.1 接口概览	3
6.2 一般规定	4
6.3 项目管理接口	6
6.4 信息容器管理接口	8
附录 A (资料性) 接口请求示例(JSON)	17
A.1 获取项目列表接口	17
A.2 获取项目信息接口	18
A.3 创建项目接口	18
A.4 修改项目接口	19
A.5 删除项目接口	19
A.6 获取信息容器元数据列表接口	20
A.7 创建信息容器接口	21
A.8 获取信息容器详细信息接口	22
A.9 更新信息容器接口	23
A.10 追加信息容器引用链接接口	24
A.11 删除信息容器引用链接接口	25
A.12 删除信息容器接口	25
A.13 获取信息容器版本列表接口	26
A.14 存档信息容器接口	26
A.15 获取信息容器内容列表接口	26
A.16 更新信息容器内容接口	27
A.17 追加信息容器内容接口	28
A.18 删除信息容器内容接口	28
A.19 获取信息容器内容版本列表接口	28
A.20 获取信息容器内容接口	29

A.21 获取信息容器日志接口	30
A.22 获取预定义规则列表接口	30
A.23 信息容器检查接口	31
A.24 修改信息容器状态接口	31
附录 B (资料性) 检查结果文件内容示例(JSON)	32



前 言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》的规定起草。

本文件是 GB/T 45393《信息技术 建筑信息模型(BIM)软件》的第 5 部分。GB/T 45393 已经发布了以下部分：

- 第 2 部分：参数化模型；
- 第 4 部分：网格模型；
- 第 5 部分：数据接口。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本文件起草单位：中信数智(武汉)科技有限公司、清华大学、中国电子技术标准化研究院、山东智汇云建筑信息科技有限公司、中国测绘科学研究院、中国铁路设计集团有限公司、中国建筑第八工程局有限公司、中国建筑西南设计研究院有限公司、泰瑞数创科技(北京)股份有限公司、广联达科技股份有限公司、上海秉匠信息科技有限公司、中交第二公路勘察设计研究院有限公司、广州中望智城数字科技有限公司、上海建科数创智能科技有限公司、中建工程产业技术研究院有限公司、宝钢工程技术集团有限公司、核工业工程研究设计有限公司、太原市轨道交通智慧建造科技有限公司、北京城建智控科技股份有限公司。

本文件主要起草人：彭波、胡继强、王瑶瑶、郑州、顾明、高歌、万斌、连井龙、苏亚武、方长建、高颖、陶坤旺、赫英楠、赵飞飞、刘新宇、陈亚军、刘俊伟、孙中奎、高阳、王欣南、白宇、李士才、曾莎洁、邱奎宁、袁磊、杨远松、陈曦、刘喆。

引 言

BIM 软件以三维化、结构化、语义化模型的建立与运用为特征,通过提供 BIM 应用所需的建模、计算、表现、管理、交付、检查、协同功能,在建筑全生命周期不同阶段、不同专业、不同应用中发挥着支撑作用。

GB/T 45393《信息技术 建筑信息模型(BIM)软件》旨在指导从业人员对 BIM 软件形成一致认识,在 BIM 软件开发中,采用本系列标准,实现 BIM 软件的互联互通、BIM 模型的交换共享,提高 BIM 软件的可用性、适用性和成熟度,促进 BIM 软件生态的形成,助力 BIM 模型的资产化。

GB/T 45393 拟由 5 部分构成。

- 第 1 部分:通用要求。明确 BIM 软件的基本概念、功能和数据的要求,目的是协助 BIM 软件的需求定义、开发实现。
- 第 2 部分:参数化模型。给出采用标准化参数集表示几何对象的模型格式,接收方通过解析与修改几何对象参数来显示、修改 BIM 模型中的对象,目的是实现 BIM 软件间可编辑的模型交换。
- 第 3 部分:模型视图定义。提供对模型中的内容进行形式化定义的语言,目的是根据特定需求对提交的模型进行完备性检查或模型视图抽取。
- 第 4 部分:网格模型。给出基于三角形网格表示几何对象不可参数化编辑的模型格式,目的是用于 BIM 模型可视化的表现、沟通、管理。
- 第 5 部分:数据接口。给出不同 BIM 软件与平台间的数据调用方式、发送与返回信息,目的是实现不同的 BIM 软件以接口的方式共享 BIM 数据。

信息技术 建筑信息模型(BIM)软件

第5部分:数据接口

1 范围

本文件给出了 BIM 软件数据接口的类别及各类别的数据结构和相关参数。

本文件适用于支持开发和实现不同 BIM 协同平台、BIM 全过程管理平台等公共数据环境间的数据交换,也适用于支持开发相关应用软件。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中,注日期的引用文件,仅该日期对应的版本适用于本文件;不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 7408 日期和时间 信息交换表示法

GB 18030 信息技术 中文编码字符集

GB/T 36456.3 面向工程领域的共享信息模型 第3部分:测试方法

3 术语和定义

下列术语和定义适用于本文件。

3.1

数据接口 data interface

计算机软件系统之间为实现数据传输、信息交换所做出的一组约定。

3.2

信息容器 information container

实现 BIM 数据交换与共享操作的基本单元。

3.3

元数据 metadata

描述数据的数据。

注1:例如属性名称、单位等为元数据。

注2:信息容器中的信息资源由元数据描述,以便更轻松地获取、存储、组织、查找和使用此数据。

注3:通过元数据指定项目阶段、信息交换需求、信息详细程度等信息。

注4:统一组织信息资源的基础由定义属性和值范围的元数据模式提供。元数据模式根据项目需求进行扩展,例如指定模型格式。

4 缩略语

下列缩略语适用于本文件。

BIM:建筑信息模型(Building Information Model)

HTTP:超文本传输协议(HyperText Transfer Protocol)

HTTPS:超文本传输安全协议(HyperText Transfer Protocol Secure)

JSON:JavaScript 对象简谱(JavaScript Object Notation)

MIME:多用途互联网邮件扩展类型(Multipurpose Internet Mail Extensions)

URL:统一资源定位符(Uniform Resource Locator)

5 基本要求

5.1 接口

接口要求如下:

- a) 接口应使用 HTTP/HTTPS 协议进行数据传输;
- b) 接口中的时间类型应符合 GB/T 7408 规定的相关要求;
- c) 接口请求与响应应采用 JSON 或可扩展标记语言(XML)数据格式,并在请求头中注明;
- d) 接口中数据编码格式应符合 GB 18030 规定的相关要求。

5.2 信息容器

信息容器由容器元数据和容器内容组成,其结构如图 1 所示。容器元数据描述信息容器中的数据概要,包括创建信息、描述信息、属性信息、状态等;容器内容为模型、图纸、文档、结构化数据以及对其他容器的引用信息。具体要求如下。

- a) 每个信息容器应归属于特定的项目。
- b) 信息容器应支持历史版本,当为信息容器生成一个新历史版本时,信息容器当前的元数据和内容信息被保存到新生成的历史版本中,历史版本中所有数据不能再被修改。
- c) 为确保严格的数据中立性和企业数据的保护,信息容器的版本管理仅限于信息所有者的范围。只有获得所有者的明确许可,才可获取版本历史记录及提供历史版本的访问权限。

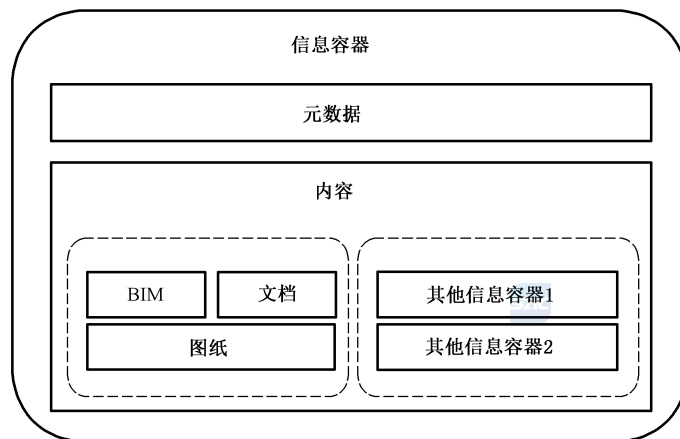


图 1 信息容器结构

5.3 安全控制

接口实现应通过开放授权 2.0(OAuth2.0)协议提供授权方式,宜采用授权码方式进行授权保障对外发布接口的安全性。应按以下授权流程实现对接口的安全控制:

- a) 请求方通过请求授权服务换取授权码;

- b) 请求方使用换取的授权码再次访问授权服务器换取令牌(token)；
- c) 请求方通过换取的令牌(token)访问接口资源。

6 数据接口

6.1 接口概览

数据接口包含项目管理和信息容器管理两个类别,分别描述在项目和信息容器管理两个层级应支持的操作,以及每种操作的传入数据结构与响应数据结构。本文件定义的数据接口列表见表1,各接口的请求示例见附录A。

表 1 接口列表

接口类别	接口名称	英文名称	说明
项目管理	获取项目列表	getProjectList	获取与当前用户相关的项目列表
	获取项目信息	getProjectDetail	获取指定项目的详细信息
	创建项目	createProject	通过请求创建一个新的项目
	修改项目	updateProject	通过请求修改一个已有的项目
	删除项目	deleteProject	请求删除指定的项目
信息容器管理	获取信息容器元数据列表	getContainerMetadataList	获取特定项目中的信息容器元数据列表
	创建信息容器	createContainer	创建一个新的信息容器
	获取信息容器详细信息	getContainerDetail	获取特定信息容器的详细信息
	更新信息容器	updateContainer	更新已有的信息容器元数据
	追加信息容器引用链接	appendRef	将其他容器追加到当前容器的引用字段
	删除信息容器引用链接	deleteRef	删除其他容器在当前容器的引用信息
	删除信息容器	deleteContainer	删除一个特定的信息容器
	获取信息容器版本列表	getContainerVersionList	获取特定信息容器的版本列表
	存档信息容器	saveContainer	将特定信息容器的状态进行存档
	获取信息容器内容列表	getContentList	获取特定容器中内容列表
	更新信息容器内容	updateContent	对容器内容特定项进行更新
	追加信息容器内容	appendContent	追加容器内容到信息容器中
	删除信息容器内容	deleteContent	将特定项从信息容器中删除
	获取信息容器内容版本列表	getContentVersionList	获取特定信息容器内容的版本列表
	获取信息容器内容	getContent	获取特定信息容器的特定内容
	获取信息容器日志	getContainerLog	获取信息容器的用户操作历史记录
	获取预定义规则列表	getSchemaList	根据用户在项目中的权限,获取相应规则列表
	信息容器检查	checkContainer	检查信息容器中的 BIM 是否满足规则要求
	修改信息容器状态	updateContainerStatus	修改特定信息容器的状态

6.2 一般规定

6.2.1 基本数据类型

本章节中使用到的基础数据类型说明见表 2。

表 2 基本数据类型

数据类型	名称	说明
string	字符串	字符串类型
int	整数	数据范围为-2147483648~2147483647
dateTime	日期	日期时间
list<T>	泛型列表	子数据项类型为 T 的列表
object	对象	任意对象
timestamp	时间戳	时间戳类型
stream	数据流	数据流类型

6.2.2 通用参数

接口采用授权码方式实现安全访问控制,按 5.3 所述流程获取访问令牌后,后续每次接口的访问,都应将令牌置于请求头中。授权及其他通用请求头参数见表 3。

请求的接口返回值为对象列表时,应进行分页查询,分页查询参数见表 4。

接口的返回基本对象数据结构应符合表 7 的要求。

表 3 请求头通用参数

参数名	中文名称	备注
authorization	授权	授权令牌,请求头参数,格式为: Bearer { token }。如: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhd2I6IjU6uvll0sInVzZXJfY29kZSI6ImVlNGRlMTc2M2Y0ZjIzN0aSI6ImJmZDQ2MjFkLW.M1ODAtNGIyYi04Yjg5LTE2ZDc2N
contentType	内容类型	请求或响应头的格式和编码格式,请求和响应头参数,如: application/json; charset=UTF-8
timestamp	时间戳	时间戳,请求头参数

表 4 分页查询参数

参数名称	中文名称	数据类型	必选	备注
page	页码	int	否	指定的页码,从 0 开始
pageSize	每页数量	int	否	指定的每页大小
sort	排序	string	否	以“,”作为分隔符的属性名列表,按属性名在列表中的顺序排序,默认升序,“-”表示降序。如按创建人升序,再按更新时间降序,可写为“createdBy,-updateAt”
filters	过滤	List<FilterItem>	否	过滤器对象列表,过滤器对象结构见表 5

表 5 FilterItem 数据结构

参数名称	中文名称	数据类型	必选	备注
parameterName	参数名	string	否	过滤器项的参数名
operator	操作符	int	否	操作符取值范围应符合表 6 的要求
value	值	string	否	过滤器项的值

表 6 操作符定义

值	名称
-4	不小于
-3	不大于
-2	不包含
-1	不等于
1	等于
2	包含
3	大于
4	小于

表 7 接口基本输出参数

参数名称	中文名称	数据类型	备注
code	状态码	int	接口的状态码应以 HTTP 状态码为最高优先级,在保证 HTTP 语义且有切实需求的前提下,不成功代码应符合表 8 的要求
message	信息	string	系统自定义信息
data	数据对象	object	响应中返回的实际的数据
pagination	分页	pagination	当接口返回对象为列表时,应使用 pagination 对象指示分页状态。pagination 对象结构见表 9

表 8 不成功代码定义

错误编码	备注
1000	参数错误
1001	指定页码不存在
1002	参数类型错误
1003	超出规定长度
1004	超出规定取值范围
1005	参数含有不支持的字符
1006	缺少必要的参数

表 8 不成功代码定义 (续)

错误编码	备注
1007	参数冲突
1100	对象不存在
1101	项目不存在
1102	容器不存在
1103	容器内容不存在
1104	版本不存在
1200	对象已存在
1201	项目名称已存在
1202	容器名称已存在
1300	权限不足
1301	无删除权限
1302	无修改权限
1303	无读取权限
1304	无创建权限

表 9 pagination 数据结构

参数名称	中文名称	数据类型	备注
page	页码	int	当前页码,从 0 开始
size	当前数量	int	当前返回列表中的对象实际个数
total	总数量	int	当前资源具有的全部对象个数

6.2.3 用户与权限

本文件描述数据的操作方式,不对用户与权限做出定义,接口实现方应向用户和接口调用方提供相关的用户与权限服务,例如项目中包含的人员、项目人员的角色、项目人员对特定操作的权限与授权等。

在部分涉及需要包含用户标识符数据的接口中,用户标识符由具体统一数据环境实现定义,调用方通过实现方提供的相关服务进行查询。

6.3 项目管理接口

6.3.1 获取项目列表

获取与当前用户相关的项目列表,其输出参数为对象列表,其中每个对象的参数应符合表 10 的要求。

表 10 项目信息数据结构

参数名称	中文名称	数据类型	备注
id	项目标识符	string	项目唯一标识符。本文件中“唯一标识符”应满足在公共数据环境中全局唯一的要求
name	项目名称	string	项目名称
description	描述	string	项目描述信息
createdAt	创建时间	dateTime	项目创建时间
createdBy	创建人	string	项目创建用户的唯一标识符
updatedAt	修改时间	dateTime	项目最新修改时间
updatedBy	修改人	string	项目最新修改用户的唯一标识符

6.3.2 获取项目信息

获取指定项目的详细信息,其输入参数应符合表 11 的要求,输出参数应符合表 10 的要求。

表 11 获取项目信息输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符

6.3.3 创建项目

通过请求创建一个新的项目,其输入参数应符合表 12 的要求,输出参数应符合表 13 的要求。

表 12 创建项目输入参数

参数名称	中文名称	数据类型	必选	备注
name	名称	string	是	项目名称。排序、过滤字段
description	描述	string	否	项目的详细描述信息



表 13 创建项目输出参数

参数名称	中文名称	数据类型	备注
id	项目标识符	string	项目的唯一标识符

6.3.4 修改项目

通过请求修改一个已有的项目,其输入参数应符合表 14 的要求。返回通用数据结构。

表 14 修改项目输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
name	名称	string	否	项目名称
description	描述	string	否	项目的详细描述信息

6.3.5 删除项目

请求删除指定的项目,其输入参数应符合表 15 的要求。返回通用数据结构。

表 15 删除项目输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符

6.4 信息容器管理接口

6.4.1 获取信息容器元数据列表

获取特定项目中的信息容器元数据列表,其输入参数应符合表 16 的要求,输出参数为对象列表,其中每个对象的参数应符合表 18 的要求。

表 16 获取信息容器元数据列表输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerType	容器类型	int	否	容器的类型,取值范围应符合表 17 的要求

表 17 信息容器的类型

值	名称	说明
0	模型容器	内容仅包含模型的信息容器
1	规则容器	内容仅包含规则的信息容器。其中规则是指符合 GB/T 36456.3、模型视图定义标准的语法定义,用于模型检查的语句或语句文件
2	混合容器	包含文档、图纸或其他文件的容器

表 18 获取信息容器元数据列表输出参数

参数名称	中文名称	数据类型	备注
id	容器标识符	string	容器的唯一标识符
name	名称	string	容器名称。排序、过滤字段
description	描述	string	容器的详细描述信息

表 18 获取信息容器元数据列表输出参数（续）

参数名称	中文名称	数据类型	备注
status	状态	int	容器的状态,应至少包含表 19 中定义的 4 种状态
updatedAt	更新时间	dateTime	容器的更新时间。排序、过滤字段
createdBy	创建者	string	容器的创建者系统中的唯一标识符
containerVersion	版本	string	容器版本
createdAt	创建时间	dateTime	容器创建时间
updateBy	更新者	string	容器的最后更新者系统中的唯一标识符
containerType	容器类型	int	信息容器类型,取值应符合表 17 的要求

表 19 信息容器的状态

值	名称	说明
0	进行中	指示当前容器为私有状态,仅供创建者修改用
1	共享	指示当前容器为临时状态,仅供参考或协同设计过程中的交流使用
2	发布	指示当前容器为已认证状态,是可信任的具有法律效力的版本,是后续其他专业人员、协作方开展工作的基础
3	存档	指示当前容器为最终成果状态,仅供部分项目参与方访问,不能被修改,是法律证明的基础

6.4.2 创建信息容器

创建一个新的信息容器。其输入参数应符合表 20 的要求,输出参数应符合表 21 的要求。

表 20 创建信息容器输入参数

参数名称	中文名称	数据类型	必选	备注
name	容器名称	string	是	容器名称。排序、过滤字段
description	描述	string	否	容器的详细描述信息
containerType	容器类型	int	否	容器的类型,取值范围应符合表 17 的要求
contents	内容	List<content>	否	容器的内容列表。列表中对象的数据结构应符合表 22 的要求
refs	引用	List<ref>	否	对其他信息容器的引用列表。列表中对象的数据结构应符合表 23 的要求

表 21 创建信息容器输出参数

参数名称	中文名称	数据类型	备注
id	容器标识符	string	容器的唯一标识符

表 22 content 对象结构

参数名称	中文名称	数据类型	备注
id	容器内容标识符	string	容器内容的唯一标识符
name	容器内容名称	string	容器内容的名称
location	存储位置	string	内容的存储位置,即内容数据下载地址
type	类型	string	内容的类型,符合 IANA 的 MIME 规格定义
createdBy	创建者	string	容器内容的创建者系统中的唯一标识符
createdAt	创建时间	dateTime	容器内容的创建时间
updatedAt	更新时间	dateTime	容器内容的更新时间
schemaVersion	规则版本	string	规则版本。仅规则容器有效
schemaSubset	规则子集	string	规则的子集名称或者链接地址。仅规则容器有效
contentVersion	内容版本	string	内容的版本

表 23 ref 对象结构

参数名称	中文名称	数据类型	备注
id	容器标识符	string	同项目下被引用容器的唯一标识符
containerVersion	容器版本	string	被引用容器的版本

6.4.3 获取信息容器详细信息

获取特定信息容器的详细信息,其输入参数应符合表 24 的要求,输出参数应符合表 25 的要求。

表 24 获取信息容器详细信息输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
containerVersion	容器版本	string	否	容器的版本号。不传或传空时,返回最新版本

表 25 获取信息容器详细信息输出参数

参数名称	中文名称	数据类型	备注
id	容器标识符	string	容器的唯一标识符
name	名称	string	容器名称
description	描述	string	容器的详细描述信息
createdAt	创建时间	string	容器的创建时间
updatedAt	更新时间	string	容器的更新时间
createdBy	创建者	string	容器的创建者用户标识符

表 25 获取信息容器详细信息输出参数 (续)

参数名称	中文名称	数据类型	备注
containerType	容器类型	int	容器的类型,取值范围应符合表 17 的要求
containerVersion	容器版本	string	容器的版本
status	状态	int	容器的状态,应至少包含表 19 中定义的 4 种状态
projectId	项目标识符	string	容器所属项目的唯一标识符
contents	内容列表	List<content>	容器的内容列表,数据结构见表 22
refs	引用列表	List<ref>	对其他信息容器的引用列表,数据结构见表 23

6.4.4 更新信息容器

更新已有的信息容器元数据,其输入参数应符合表 26 的要求。返回通用数据结构。

表 26 更新信息容器输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
name	名称	string	否	容器名称
description	描述	string	否	容器的详细描述信息
status	状态	int	否	容器的状态,应至少包含表 19 中定义的 4 种状态
contents	内容	List<content>	否	容器的内容列表。列表中对象为 content,其数据结构见表 22
refs	引用	List<ref>	否	对其他信息容器的引用列表。列表中对象为 ref,其数据结构见表 23

6.4.5 追加信息容器引用链接

将其他容器追加到当前容器的引用字段,其输入参数应符合表 27 的要求。返回通用数据结构。

表 27 追加信息容器引用链接输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
refs	引用	List<ref>	否	对其他信息容器的引用列表。列表中对象为 ref,其数据结构见表 23

6.4.6 删除信息容器引用链接

删除其他容器在当前容器的引用信息,其输入参数应符合表 28 的要求。返回通用数据结构。

表 28 删除信息容器引用链接输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
ref	引用	ref	是	ref 对象,其数据结构见表 23

6.4.7 删除信息容器

删除一个特定的信息容器,其输入参数应符合表 29 的要求。返回通用数据结构。

表 29 删除信息容器输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符

6.4.8 获取信息容器版本列表

获取特定信息容器的版本列表,其输入参数应符合表 30 的要求,输出参数应符合表 31 的要求。

表 30 获取信息容器版本信息输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符

表 31 获取信息容器版本信息输出参数

参数名称	中文名称	数据类型	必选	备注
versions	版本列表	List<string>	是	容器版本号列表

6.4.9 存档信息容器

将特定信息容器的状态进行存档,存档后的信息容器通过历史版本访问,不支持修改,同时当前的信息容器版本号增加。如当前状态与上一版本无变化,则版本号不进行增加。输入参数应符合表 32 的要求。返回通用数据结构。

表 32 存档信息容器输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符

6.4.10 获取信息容器内容列表

获取特定容器中内容列表。其输入参数应符合表 33 的要求,输出参数应符合表 34 的要求。

表 33 获取信息容器内容输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符

表 34 获取信息容器内容输出参数

参数名称	中文名称	数据类型	备注
contents	内容	List<content>	容器的内容列表。列表中对象为 content,其数据结构见表 22

6.4.11 更新信息容器内容

对容器内容特定项进行更新,其输入参数应符合表 35 的要求。返回通用数据结构。

表 35 更新信息容器内容输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
contentId	内容标识符	string	是	内容的唯一标识符
name	名称	string	否	内容名称
location	存储位置	string	否	内容的存储位置,即内容数据下载地址
type	类型	string	否	内容的类型,符合 IANA 的 MIME 规格定义
schema	规则	string	否	内容的规则名称或者规则链接地址
schemaVersion	规则版本	string	否	规则版本。仅规则容器有效
schemaSubset	规则子集	string	否	规则的子集名称或者链接地址。仅规则容器有效

6.4.12 追加信息容器内容

追加容器内容到信息容器中,其输入参数应符合表 36 的要求。输出参数应符合表 37 的要求。

表 36 追加信息容器内容输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
name	名称	string	是	内容名称
location	存储位置	string	否	内容的存储位置,即内容数据下载地址

表 36 追加信息容器内容输入参数 (续)

参数名称	中文名称	数据类型	必选	备注
type	类型	string	否	内容的类型,符合 IANA 的 MIME 规格定义
schema	规则	string	否	内容的规则名称或者规则链接地址
schemaVersion	规则版本	string	否	规则版本。仅规则容器有效
schemaSubset	规则子集	string	否	规则的子集名称或者链接地址。仅规则容器有效

表 37 追加信息容器内容输出参数

参数名称	中文名称	数据类型	备注
contentId	内容标识符	string	追加成功的内容唯一标识符

6.4.13 删除信息容器内容

将特定项从信息容器中删除,其输入参数应符合表 38 的要求。返回通用数据结构。

表 38 删除信息容器内容输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
contentId	内容标识符	string	是	内容的唯一标识符

6.4.14 获取信息容器内容版本列表

获取特定信息容器内容的版本列表,其输入参数应符合表 39 的要求,输出参数应符合表 40 的要求。

表 39 获取信息容器内容版本列表输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
contentId	内容标识符	string	是	内容的唯一标识符

表 40 获取信息容器内容版本列表输出参数

参数名称	中文名称	数据类型	备注
versions	版本列表	List<string>	容器内容版本号列表

6.4.15 获取信息容器内容

获取特定信息容器的特定内容,其输入参数应符合表 41 的要求,输出参数应符合表 22 的要求。

表 41 获取信息容器内容输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
contentId	内容标识符	string	是	内容的唯一标识符
contentVersion	内容版本	string	否	内容的版本。不传或传空时返回最新版本

6.4.16 获取信息容器日志

获取信息容器的用户操作历史记录,其输入参数应符合表 42 的要求,输出参数为 log 对象列表,log 对象结构应符合表 43 的要求。

表 42 获取信息容器日志输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符

表 43 log 对象结构

参数	中文名称	数据类型	备注
userId	用户 id	string	用户唯一标识符
containerId	容器标识符	string	信息容器唯一标识符
date	记录时间	dateTime	日志记录的时间
action	操作类型	string	用户操作类型,如增加、删除等

6.4.17 获取预定义规则列表

根据当前用户在当前项目中的权限职责,获取相应规则列表,其输入参数应符合表 44 的要求,输出参数应符合表 45 的要求。

表 44 获取预定义规则列表输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
status	容器状态	int	否	容器的状态,其定义见表 19。不传值时,查询项目下全部规则列表

表 45 获取预定义规则列表输出参数

参数	中文名称	数据类型	备注
schemas	规则容器列表	List<Container>	适用于当前用户的规则容器列表,其中每个对象的参数应符合表 25 的要求

6.4.18 信息容器检查

容器及其引用容器中若存在模型,应进行容器检查。检查容器时,应将需检查的规则容器引用追加到该容器或其引用容器中。检查时,针对该容器及其引用容器中的模型和规则进行检查。容器及其所有引用容器通过检查时,该容器检查结果为通过。其输入参数应符合表 46 的要求,输出参数应符合表 47 的要求。

表 46 容器检查输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	待校验容器的唯一标识符

表 47 容器检查输出参数

参数	中文名称	数据类型	备注
checkResult	检查结果	int	校验结果,0:不通过;1:通过
checkCode	检查码	string	校验码。当校验结果为“通过”时不为空,否则为空
location	检查结果文件位置	string	检查结果文件的位置(url),文件内容示例见附录 B
schema	结果文件定义	string	结果定义文件位置(url)
schemaVersion	结果文件定义版本	string	1.0.0

6.4.19 修改信息容器状态

修改特定信息容器的状态,信息容器状态的取值应符合表 19 中“容器状态”参数的要求。修改容器状态接口输入参数应符合表 48 的要求。返回通用数据结构。

表 48 修改信息容器状态输入参数

参数名称	中文名称	数据类型	必选	备注
projectId	项目标识符	string	是	项目的唯一标识符
containerId	容器标识符	string	是	容器的唯一标识符
status	容器状态	int	是	要修改的目标状态,定义见表 19
checkCode	校验码	string	是	校验码,由容器校验接口取得

附 录 A
(资料性)
接口请求示例(JSON)

A.1 获取项目列表接口

Url: https: // {domain} /project/list

Method: POST

Content-Type: application/json; charset=utf8

Authorization: bearer {accessToken}

Header: -

Body: {

```

    "page": 200,
    "pageSize": 10,
    "sort": "createdBy,-updateAt",
    "filters": [
      {
        "filter": "name",
        "operator": 2,
        "filterValue": "{关键字}"
      },
      {
        "filter": "createdAt",
        "operator": 3,
        "filterValue": "2021-12-21"
      }
    ]
  }
```

}

返回:

```

{
  "code": 200,
  "message": "成功",
  "data": [
    {
      "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",
      "name": "项目名称 1",
      "description": "{项目描述}",
      "createdAt": "2022-01-01 09:10:10",
      "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
      "updatedAt": "2021-02-01 09:10:10",
      "updatedBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541"
    },
  ]
}
```

```
{
  "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",
  "name": "项目名称 2",
  "description": "{项目描述}",
  "createdAt": "2022-03-01 09:10:10",
  "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
  "updatedAt": "2021-02-01 09:10:10",
  "updatedBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541"
},
.....
],
"paging": {
  "page": 0,
  "size": 10,
  "total": 100
}
}
```

A.2 获取项目信息接口

Url: https: // {domain} /project / {id}

Method: GET

Content-Type: application /json ; charset = utf8

Authorization: bearer {accessToken}

Header: -

Body: -

返回:

```
{
  "code": 200,
  "message": "获取项目信息成功",
  "data": {
    "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",
    "name": "项目名称 1",
    "description": "{项目描述}",
    "createdAt": "2021-01-01 09:10:10",
    "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
    "updatedAt": "2021-02-01 09:10:10",
    "updatedBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541"
  }
}
```

A.3 创建项目接口

Url: https: // {domain} /project /create

Method: POST

```

Content-Type:application/json;charset=utf8
Authorization:bearer {accessToken}
Header:-
Body: {
    "name": "项目名称 3",
    "description": "{项目描述}"
}
返回:
{
    "code": 200,
    "message": "成功",
    "data": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02"
}

```

A.4 修改项目接口



```

Url:https://{domain}/project/update? id={id}
Method:POST
Content-Type:application/json;charset=utf8
Authorization:bearer {accessToken}
Header:-
Body: {
    "name": "项目名称 4",
    "description": "{项目描述}"
}
返回:
{
    "code": 200,
    "message": "成功",
    "data": null
}

```

A.5 删除项目接口

```

Url:https://{domain}/project/delete/{id}
Method:DELETE
Content-Type:application/json;charset=utf8
Authorization:bearer {accessToken}
Header:-
Body:-
返回:
{
    "code": 200,
    "message": "成功",
    "data": null
}

```

}

A.6 获取信息容器元数据列表接口

Url: https://{{domain}}/container/list? projectId={{projectId}}

Method:POST

Content-Type:application/json;charset=utf8

Authorization:bearer {{accessToken}}

Header:-

Body:{

```

    "page": 0,
    "pageSize": 10,
    "sort": "createdBy,-updateAt",
    "filters": [
      {
        "filter": "name",
        "operator": 2,
        "filterValue": "{关键字}"
      },
      {
        "filter": "createdAt",
        "operator": 3,
        "filterValue": "2021-12-21"
      },
      .....
    ]

```

}

返回:

```

{
  "code": 200,
  "message": "成功",
  "data": [
    {
      "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",
      "name": "容器 1",
      "description": "{容器描述}",
      "status": "0",
      "containerVersion": "1",
      "containerType": "1",
      "createdAt": "2021-1-1 09:10:10",
      "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541"
      "updatedBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541"
      "updatedAt": "2021-1-2 09:00:00"
    }
  ]
}

```

```

    },
    {
      "id": "F82781C5-1363-4681-8271-006B4AC7E092",
      "name": "容器 2",
      "description": "{容器描述}",
      "status": "0",
      "containerVersion": "1",
      "containerType": "1",
      "createdAt": "2021-1-2 09:00:00",
      "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
      "updatedAt": "2021-1-2 09:00:00",
      "updatedBy": "2ED52BB9"
    }
  ]
  "paging": {
    "page": 0,
    "size": 10,
    "total": 100
  }
}

```

A.7 创建信息容器接口

Url: `https://{domain}/container/create? projectId={projectId}`

Method: POST

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header:-

Body: {

```

  "id": "678B4D10-D59D-4C05-97D1-8A2942EEDD47",
  "name": "容器 3",
  "description": "{容器描述}",
  "containerType": "1",
  "contents": [
    {
      "id": "458CD8E7-D7B3-4464-B946-B237F24B06C0",
      "name": "内容 1",
      "location": "{文件下载链接}",
      "type": "application/octet-stream",
      "createdAt": "2021-1-1 09:10:10",
      "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
      "updateAt": "2021-2-1 09:10:10",
      "schemaVersion": "v1",
      "schemaSubset": "",

```

```

        "contentVersion": "v1"
    },
    .....
],
"refs": [
    {
        "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",
        "containerVersion": "v1"
    },
    {
        "id": "F82781C5-1363-4681-8271-006B4AC7E092",
        "containerVersion": "v1"
    },
    .....
]
}

```

返回:

```

{
    "code": 200,
    "message": "成功",
    "data": "81FCD465-F4B3-4408-A7EA-6DD150A681E8"
}

```

A.8 获取信息容器详细信息接口

Url: `https:// {domain} /container/info? projectId = {projectId} &containerId = {containerId} &containerVersion=v1`

Method: GET

Content-Type: application/json; charset=utf8

Authorization: bearer {accessToken}

Header: -

Body: -

返回:

```

{
    "code": 200,
    "message": "成功",
    "data": {
        "id": "458CD8E7-D7B3-4464-B946-B237F24B06C0",
        "name": "容器 1",
        "description": "{描述}",
        "createdAt": "2021-1-1 09:10:10",
        "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
        "updateAt": "2021-2-1 09:10:10",
        "containerVersion": "v1",
    }
}

```

```

        "status": "0",
"containerType": "1",
        "projectId": "9FE5526A-21F5-4BB6-8236-5405F3D394C9",
"contents": [
    {
        "id": "458CD8E7-D7B3-4464-B946-B237F24B06C0",
        "name": "内容 1",
        "location": "{文件下载链接}",
        "type": "application/octet-stream",
        "createdAt": "2021-1-1 09:10:10",
        "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
        "updatedAt": "2021-2-1 09:10:10",
        "schemaVersion": "v1",
        "schemaSubset": "",
        "contentVersion": "v1"
    },
    .....
],
"refs": [
    {
        "id": "F82781C5-1363-4681-8271-006B4AC7E092",
        "contentVersion": "v1"
    },
    {
        "id": "F82781C5-1363-4681-8271-006B4AC7E092",
        "containerVersion": "v1"
    },
    .....
]
}
}

```

A.9 更新信息容器接口

Url: `https://{domain}/container/update? projectId={projectId}&containerId={containerId}`

Method: POST

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header: -

Body: {

 " name": "容器 4",

 " description": "{容器描述}",

 " status": 1,

 " contents": [

```

    {
      "id": "458CD8E7-D7B3-4464-B946-B237F24B06C0",
      "name": "内容 1",
      "location": "{文件下载链接}",
      "type": "application/octet-stream",
      "createdAt": "2021-1-1 09:10:10",
      "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
      "updatedAt": "2021-2-1 09:10:10",
      "schemaVersion": "v1",
      "schemaSubset": "",
      "contentVersion": "v1"
    },
    .....
  ],
  "refs": [
    {
      "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",
      "containerVersion": "v1"
    },
    {
      "id": "F82781C5-1363-4681-8271-006B4AC7E092",
      "containerVersion": "v1"
    },
    .....
  ]
}
返回:
{
  "code": 200,
  "message": "更新成功",
  "data": null
}

```

A.10 追加信息容器引用链接接口

Url: `https://{domain}/container/appendRef? projectId = {projectId} &containerId = {containerId}`

Method: POST

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header: -

Body: [

```

  {
    "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",

```

```

        "contentVersion": "v1"
    },
    {
        "id": "F82781C5-1363-4681-8271-006B4AC7E092",
        "contentVersion": "v1"
    },
    .....
]

```

返回:

```

{
    "code": 200,
    "message": "更新成功",
    "data": null
}

```

A.11 删除信息容器引用链接接口

Url: `https:// {domain}/container/deleteRef? projectId = {projectId} & containerId = {containerId}`

Method: DELETE

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header: -

Body: {

```

    "id": "3CB02F92-2B8C-42A7-B9E9-DF468398DD02",
    "containerVersion": "v1"
}

```

}

返回:

```

{
    "code": 200,
    "message": "成功",
    "data": null
}

```



A.12 删除信息容器接口

Url: `https:// {domain}/container/delete? projectId={projectId} & containerId={containerId}`

Method: DELETE

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header: -

Body: -

返回:

```

{
    "code": 200,

```

```
"message": "删除成功",  
"data": null  
}
```

A.13 获取信息容器版本列表接口

Url: `https://{{domain}}/container/versions? projectId={projectId}&containerId={containerId}`

Method: GET
Content-Type: `application/json; charset=utf8`
Authorization: `bearer {accessToken}`
Header: -
Body: -
返回:

```
{  
  "code": 200,  
  "message": "成功",  
  "data": {  
    "versions": ["v1", "v2", "v3", .....]  
  }  
}
```

A.14 存档信息容器接口

Url: `https://{{domain}}/container/save? projectId={projectId}&containerId={containerId}`

Method: POST
Content-Type: `application/json; charset=utf8`
Authorization: `bearer {accessToken}`
Header: -
Body: -
返回:

```
{  
  "code": 200,  
  "message": "成功",  
  "data": null  
}
```



A.15 获取信息容器内容列表接口

Url: `https://{{domain}}/content/list? projectId={projectId}&containerId={containerId}`

Method: GET
Content-Type: `application/json; charset=utf8`
Authorization: `bearer {accessToken}`
Header: -
Body: -
返回:

```

{
  "code": 200,
  "message": "成功",
  "data": [
    {
      "id": "458CD8E7-D7B3-4464-B946-B237F24B06C0",
      "name": "内容 1",
      "location": "{文件下载链接}",
      "type": "application/octet-stream",
      "createdAt": "2021-1-1 09:10:10",
      "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
      "updateAt": "2021-2-1 09:10:10",
      "schemaVersion": "v1",
      "schemaSubset": "",
      "contentVersion": "v1"
    },
    .....
  ]
}

```

A.16 更新信息容器内容接口

Url: `https://{domain}/content/update? projectId={projectId} &containerId={containerId} &contentId={contentId}`

Method: POST

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header: -

Body: {

 "name": "内容 2",

 "location": "{文件下载链接}",

 "type": "application/octet-stream",

 "schemaVersion": "v2",

"schemaSubset": "",

"schema": ""

}

返回:

```
{
```

 "code": 200,

 "message": "成功",

 "data": null

```
}
```

A.17 追加信息容器内容接口

Url: `https://{{domain}}/container/appendContent? projectId={projectId}&containerId={containerId}&contentId={contentId}`

Method: POST

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header: -

Body: {

 " name " : " 内容 2 " ,

 " location " : " { 文件 下载 链接 } " ,

 " type " : " application/octet-stream " ,

 " schemaVersion " : " v1 " ,

 " schemaSubset " : " " ,

 " schema " : " " ,

}

返回:

{

 " code " : 200 ,

 " message " : " 更新 成功 " ,

 " data " : " 458CD8E7-D7B3-4464-B946-B237F24B06C0 " ,

}

A.18 删除信息容器内容接口

Url: `https://{{domain}}/container/deleteContent? projectId={projectId}&containerId={containerId}&contentId={contentId}`

Method: DELETE

Content-Type: `application/json; charset=utf8`

Authorization: `bearer {accessToken}`

Header: -

Body: -

返回:

{

 " code " : 200 ,

 " message " : " 删除 成功 " ,

 " data " : null ,

}

A.19 获取信息容器内容版本列表接口

Url: `https://{{domain}}/content/versions? projectId={projectId}&containerId={containerId}&contentId={contentId}`

Method: GET

Content-Type: `application/json; charset=utf8`

Authorization: bearer {accessToken}

Header:-

Body:-

返回:

```
{
  "code": 200,
  "message": "成功",
  "data": {
    "versions": [
      "v1",
      "v2",
      "v3",
      .....
    ]
  }
}
```

A.20 获取信息容器内容接口

Url: https://{{domain}}/content/info? projectId = {projectId} &containerId = {containerId}
&contentId={contentId}? contentVersion={contentVersion}

Method: GET

Content-Type: application/json; charset=utf8

Authorization: bearer {accessToken}

Header:-

Body:-

返回:

```
{
  "code": 200,
  "message": "成功",
  "data": {
    "contents": [
      {
        "id": "458CD8E7-D7B3-4464-B946-B237F24B06C0",
        "name": "内容 1",
        "location": "{文件下载链接}",
        "type": "application/octet-stream",
        "createdAt": "2021-1-1 09:10:10",
        "createdBy": "2ED52BB9-8EF8-4F83-AECD-FB386D47B541",
        "updateAt": "2021-2-1 09:10:10",
        "schemaVersion": "v1",
        "schemaSubset": "",
        "contentVersion": "v1"
      },

```

```
        .....  
    ]  
  }  
}
```

A.21 获取信息容器日志接口

Url: https://{{domain}}/container/log? projectId={projectId}

Method:GET

Content-Type:application/json;charset=utf8

Authorization:bearer {accessToken}

Header:-

Body:-

返回:

```
{  
  "code": 200,  
  "message": "成功",  
  "data": [  
    {  
      "userId": "458CD8E7-D7B3-4464-B946-B237F24B06C0",  
      "containerId": "2110DD14-F220-4A69-B99E-DB146BF172A0",  
      "date": "2021-1-1 09:10:10",  
      "action": "add"  
    },  
    .....  
  ]  
}
```

A.22 获取预定义规则列表接口

Url: https://{{domain}}/schema/list? projectId={projectId}

Method:GET

Content-Type:application/json;charset=utf8

Authorization:bearer {accessToken}

Header:-

Body:-

返回:

```
{  
  "code": 200,  
  "message": "成功",  
  "data": [  
    {schema1},{schema2}.....  
  ]  
}
```

A.23 信息容器检查接口

Url: https: // {domain} /container/check

Method: POST

Content-Type: application/json; charset = utf8

Authorization: bearer {accessToken}

Header: -

Body:

```
{
  "projectId": "id",
  "containerId": "id",
}
```

返回:

```
{
  "code": 200,
  "message": "成功",
  "data": {
    "checkResult": 1,
    "checkCode": "ZDc2NmJkOzzTNSIsImNsaWVudF9pZCI6ImNpaXBfZGVza3RvcCJ",
    "location": "xxxxxx",
    "schema": "dxxxx",
    "schemaVersion": "1.0"
  }
}
```

A.24 修改信息容器状态接口

Url: https: // {domain} /container/updateStatus? projectId = {projectId} & containerId = {containerId} & checkCode = {checkCode} & status = {status}

Method: GET

Content-Type: application/json; charset = utf8

Authorization: bearer {accessToken}

Header: -

Body: -

返回:

```
{
  "code": 200,
  "message": "成功",
  "data": null
}
```

附 录 B
(资料性)
检查结果文件内容示例(JSON)

```
{
  "CheckRuleResults": [
    {
      "SentenceResults": [
        {
          "SentenceIndex": "6.4.7-1",
          "Description": "电梯不应与卧室紧邻布置。",
          "Text": "所有 电梯 不紧邻 卧室",
          "EntityNames": ["卧室", "电梯"],
          "PassedIDS": ["35422"],
          "FailedIDS": [],
          "IsCheckComplete": true,
          "IdAndValue": {},
          "NotApplicableResult": {}
        }
      ],
      "Description": "电梯不应与卧室紧邻布置。",
      "Num": "6.4.7"
    }
  ],
  "SummaryText": "规则总数 1;通过规则 1/1;未通过规则 0/1;不适用规则 0/1"
}
```
