

E2MS 系统 WebApi 手册

目录

一、	基本约定	3
二、	WEB API 格式与组成	3
§2.1	格式	3
§2.2	常见参数缩写	4
§2.3	使用 API 的流程	4
§2.4	HTML 示例	4
§2.5	C# 示例	6
§2.6	JAVA 示例	8
三、	登陆与认证 API	8
§3.1	登陆或请求认证	8
§3.2	查询系统租户列表	9
§3.3	查询项目列表	10
四、	公共与基本数据操作 API	11
§4.1	标准代码(CODE)类别列表	11
§4.2	代码(TCODE)列表	12
§4.3	标准类别-查询	13
§4.4	标准类别-属性查询	14
§4.5	标准流程类型	15
§4.6	标准行为或场所、阶段的编码	16
§4.7	标准工作阶段编码表	16
§4.8	系统租户专用配置项	17
§4.9	区域-列表	19
§4.10	楼层	20
§4.11	单元	22
§4.12	设备	23
§4.13	联系单位-列表	25
§4.14	联系单位-详情	26
§4.15	联系人-列表	28
§4.16	联系人-详情	30
§4.17	维保规范	32
§4.18	服务规范	32
五、	工单与任务管理 API	32
§5.1	我的工单-列表(可指定条件)	33
§5.2	我的工单-新建	35
§5.3	我的工单-删除	36
§5.4	我的工单-主体变更	36
§5.5	我的工单-工单明细操作	37
§ 5.5.1	我的工单-工作组	37
§ 5.5.2	我的工单-工作成员(执行人)	38
§ 5.5.3	我的工单-设备明细	40
§ 5.5.4	我的工单-工单附件	44
§ 5.5.5	我的工单-巡检明细	46

§ 5.5.6	我的工单-执行步骤明细	46
§5.6	工作组	46
§ 5.6.1	工作组列表	46
§ 5.6.2	查询特定工作组	47
§ 5.6.3	新建工作组	47
§ 5.6.4	更新工作组	49
§ 5.6.5	删除工作组	49
§ 5.6.6	查询工作组成员	50
§ 5.6.7	更新工作组成员	50
§ 5.6.8	新建工作组成员	50
§ 5.6.9	删除工作组成员	50
六、	文档管理 API.....	51
§6.1	文档库列表	52
§6.2	MongoDb 数据库-列表	53
§6.3	目录-列目录	54
§6.4	目录-创建	56
§6.5	目录-修改(改名、移动等).....	57
§6.6	目录-删除	57
§6.7	文件-列表	58
§6.8	文件-修改(改名、移动等).....	59
§6.9	文件-删除	60
§6.10	文件-预览	62
§6.11	文件上传	63
§6.12	共享组-列表	64
§6.13	共享组-查询详情	65
§6.14	共享组-新建	65
§6.15	共享组-修改组信息	66
§6.16	共享组-删除组	67

一、基本约定

数据采用 JSON 格式传递。
其中日期类型采用 ISO8601 标准格式，北京+8 时区。

前端或移动端使用 JAVA、JavaScript 脚本或 HTML 的 FORM 向后端提交数据或查询数据；
API 以 RESTful 为目标，尽量贴近其标准操作：

- GET： 获取资源
- POST： 修改资源
- PUT： 创建资源
- DELETE： 删除资源

二、WEB API 格式与组成

§ 2.1 格式

为叙述方便，后面章节假设 WebApi 部署在 IP 地址为 api.losta.net 的机器上，且端口号使用 80：

WEB API 的 URL 调用格式					
http://api.losta.net/api	/子系统		控制器/	动作/	其它 (可选) 参数
固定部分	Auth	登陆与认证	参见 后面文档	List、	Schema： 系统租户 ID 或数据库名 ID： 含义见后面文档
	Common	公共信息查询		Read、	
	Basis	基础数据操作		Write、	
	Tasks	任务/工单管理		Update、	
	Doc	文档子系统		Delete	
	Report	统计与报告			

不同的 Api 一般分别支持 GET、POST、PUT、DELETE 等几个动词。考虑到部分浏览器的兼容性及使用便利性，PUT、DELETE 一般使用带动词参数的 POST 方法提供服务。
此后，登陆认证 api 之外的所有调用，都必须把 AccessKey 作为特殊的一行参数附加到 HTTP GET、或 HTTP POST、PUT、DELETE 的 HttpHeader 中去。具体方法参见后面示例代码。
AccessKey 的有效期默认 4 个小时，到期后需再次登陆认证。

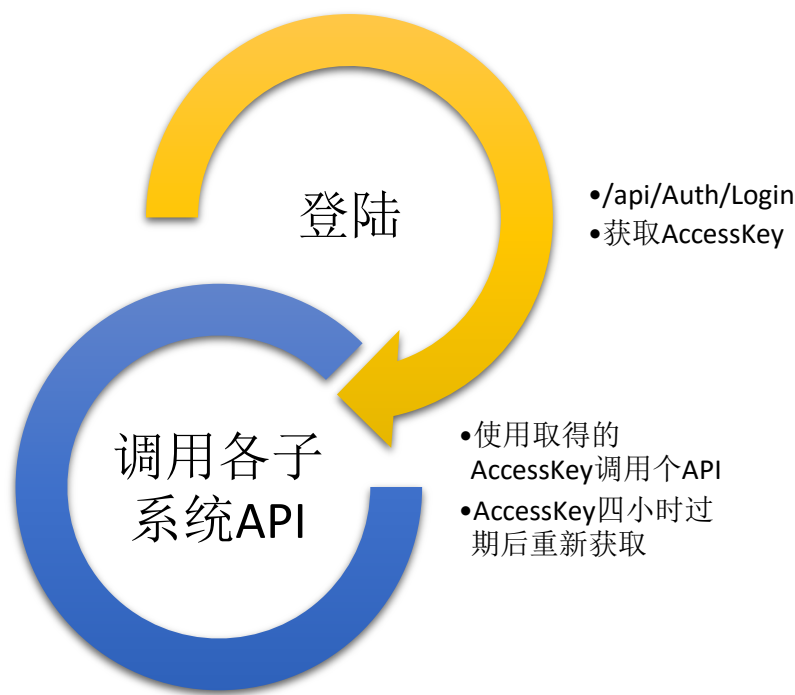
关于 Api 的 Url 参数部分：
针对文档系统，其中 Schema 为必选的 Mongodb 数据库名称，Id 为可选参数。
针对 Common、Basis、Tasks 等子系统，其中 Schema 就是必选的 SysTenantryId，Id 为可选参数。

Web-Api 的反馈一般有三种类别：
其一是按要求动作反馈的合法结果；
其二是因权限不足返回的 HTTP 401、403 错误；
其三是因参数类型不符合要求，而被 IIS 阻止或报错。这种情况下大多是 GUID 格式错误。

§ 2.2 常见参数缩写

序号	参 数 名 称	说 明
1	Id	这是一个最常用，也是含义最模糊的参数，请见个 api 章节
2	Schema	在文档系统下，Schema（或 db）表示必选文档数据库名称 在其它（如能耗监控、工单）等子系统，Schema 表示必选的 SysTenantryId
3	Pwd	密码
4	stid	系统租户 ID，SysTenantryId 的简称。有时候也使用 Schema。
5	StName	系统租户名称，SysTenantryName 的简称
	PrjId	项目 Id
	PrjName	项目名称
	BuildingId	区域 ID、楼宇 ID

§ 2.3 使用 API 的流程



§ 2.4 HTML 示例

```

function listFolderNames() {
    //var uri = 'http://api.losta.net/api/doc/folder/list/SHNHZHCH';
    var uri = 'http://localhost:1080//api/doc/folder/list/SHNHZHCH';

    $.ajax({
        url: uri,
        type: "GET",
        contentType: "application/json",
        beforeSend: function(jqXHR, settings) {
            // 在这里追加登陆时获取到的 accesskey
            jqXHR.setRequestHeader('accesskey', 'web_7e6d95f1e1af45c58f7a4c8ee9fd04cf');
        },
        success: function (data, textStatus, jqXHR) {
            // 成功....
            var itemsText = '';

            $.each(data, function (key, val) {
                var arr = JSON.stringify(val).split(',', 500);
                $.each(arr, function (key, val) {
                    itemsText = itemsText + val + "\r\n\t";
                });

                itemsText = itemsText + "\r\n";
            });

            $('#foldernames').text(itemsText);
        }
    });
}

```

JS 示例二、在档案库 SHNHZHCH 中创建一个文件夹（演示参数使用方法）

```

function createFolder() {
    var uri = 'http://localhost:1080/api/doc/folder/create/SHNHZHCH';

    $.ajax({
        url: uri,
        type: "POST",
        contentType: "application/json",
        beforeSend: function(jqXHR, settings) {
            // 在这里追加登陆时获取到的 accesskey
            jqXHR.setRequestHeader('accesskey', 'web_7e6d95f1e1af45c58f7a4c8ee9fd04cf');
        },
        data: {
            "parentFolderId": "955e533b-fd05-4a0c-901d-a281c81464f1",
            "folderId": "00000000-0000-0000-0000-000000000000",
            "FolderName": "新的子文件夹名称",
            "CreateTime": "2015-08-12T18:14:43.733+08:00",
            "Purview": []
        }
    });
}

```

```

    },
    dataType : "json",
    success: function (data, textStatus, jqXHR) {
        // 成功...
        var itemsText = '';

        $.each(data, function (key, val) {
            var arr = JSON.stringify(val).split(',', 500);
            $.each(arr, function (key, val) {
                itemsText = itemsText + val + "\r\n\t";
            });

            itemsText = itemsText + "\r\n";
        });

        $('#createdFolderInfo').text(itemsText);
    }
});
}

```

§ 2.5 C#示例

调用 WEB API 的框架性代码块（下面代码使用了第三方组件 RestSharp）

```

public class HttpApiClientException : Exception
{
    public HttpApiClientException(string message, Exception innerException)
        : base(message, innerException)
    {
    }
}

internal class HttpApiClient
{
    private static string BaseBinUrl = @"https://api.losta.net/";
    private const string quote = "\"";

    internal HttpApiClient()
    {
    }

    internal T Execute<T>(RestRequest request) where T : new()
    {
        var client = new RestClient();
        client.BaseUrl = new Uri(BaseBinUrl);

        var response = client.Execute<T>(request);
    }
}

```

```

        if (response.ErrorException != null) {
            const string message = "Error retrieving response. Check inner details for more info.";
            var apiException = new HttpApiClientException(message, response.ErrorException);
            throw apiException;
        }

        return response.Data;
    }

    internal ResponseResult Execute(RestRequest request)
    {
        return Execute(request, BaseBinUrl);
    }

    internal ResponseResult Execute(RestRequest request, string baseUrl)
    {
        var client = new RestClient();
        client.BaseUrl = new Uri(baseUrl);

        var response = client.Execute<ResponseResult>(request);

        response.ErrorMessage = response.ErrorMessage;
        response.ErrorException = response.ErrorException;

        response.Data = new ResponseResult();

        if (response.ContentLength > 0 && response.Content.StartsWith("{")) {
            response.Data.codes = parseCodeArray(response.Content);
        }

        return response.Data;
    }

    /// <summary>
    /// 读取指定文档库上的所有 GridFS 文件列表
    /// 返回的 ResponseResult 中包含 JSON 格式的文件列表
    /// </summary>
    internal ResponseResult GetGridFSFileList (string schema)
    {
        var request = new RestRequest();
        request.Method = Method.GET;
        request.AddHeader("accesskey", "web_XXXXXXX"); ➔ 这里提供 accessKey
        request.RequestFormat = DataFormat.Json;
        request.JsonSerializer = new StringAsJsonSerializer();
        request.Resource = string.Format("api/doc/GridFsList/{0}", schema);

        return Execute<ResponseResult>(request);
    }

```

```
}

```

§ 2.6 JAVA 示例

调用 WEB API 的框架性代码块

```
import web.rest.jsonClient;

// 创建 REST 客户端
var restClient = web.rest.jsonClient();

var restApi = restClient.declareApi("http://api.losta.net/api/x/x/x ")

restApi.ecranesoft.crane.post(
    email = "web@ecranesoft.com";
    password = "my password";
)

var restApi = restClient.declareApi("http:// api.losta.net/api/{...}")

/*
URL 模板中的{...} 占位符将会被资源名字替换, 多个资源名字自动使用斜杠连接
使用的实际请求 URL 为 http://服务器地址/api/forums/user
HTTP 请求动词为 "DELETE"
*/
restApi.forums.user.delete()
```

三、 登陆与认证 API

§ 3.1 登陆或请求认证

名称	登陆或请求认证
描述	
方法	GET、POST
请求	
URL	/Auth/Login/?id=xx&pwd=yyy&checkcode=zzz /Auth/Login/{id}/? pwd=yyy&checkcode=zzz
参数	Id = Id 用户登录名、或用户 email 地址 Pwd = 户登录密码 CheckCode = 登陆画面上的随机校验码, 一般是 4-6 个字符 TerminalType = 终端类型(web、mobile), 默认 web 类型

示例	http://api.losta.net/api/auth/login/?id=blueskit&pwd=12345678&Checkcode=zzzzz http://api.losta.net/api/auth/login/blueskit/?pwd=12345678&Checkcode=zzzzz
响应	
成功	成功则返回“登陆者”信息及 access_key <pre>{ "UserId": "9c08f187-c22b-4dc6-8008-e0fa68349a48", "UserName": "张三", "RoleId": "SYSMGR", "AccessKey": "web_cc0e2332da344628b77daeb801c22723", "TerminalType": "web", "SysTenantryId": "", "LoginTime": "2015-08-14T21:09:21.5200277+08:00", "OverTime": "2015-08-15T01:09:21.5200277+08:00", "Succesed": true }</pre>
失败	HTTP 403 “用户不存在，禁止访问”

如果登陆认证成功，则服务器返回 **AccessKey**。

此后，所有 API 调用都必须附带这个 **AccessKey** 信息，作为身份及确权的手段。一般附加到 HTTP GET、或 HTTP PUT、POST、DELETE 的 **HttpHeader** 中去。

AccessKey 的默认有效期 4 个小时，一般用户没有任何操作超过 4 个小时，需再次登陆认证得到新的 **AccessKey** 才能继续。

后面的 API 说明或示例不一定严格包含 **AccessKey** 信息，请务必以这里的说明为准。

§ 3.2 查询系统租户列表

名称	查询系统租户列表
描述	查询当前用户有权访问的系统租户列表 当前用户就是拥有当前 AccessKey 的登陆用户
方法	GET POST
请求	
URL	/Auth/Inquire/SysTenantryList
参数	
调用示例	http://localhost:1080/api/auth/inquire/SysTenantryList?accesskey=web_7cd570a0fbae422a97e73be110f38c0f
响应	
成功	返回允许登陆的系统租户列表，例如： <pre>[{"SysTenantryId": "SHNHZHC", "SysTenantryName": "上海两港投资管理有限公司", "CreateDate": "0001-01-01T00:00:00", "SystemTitle": "上海智城",</pre>

	<pre>"SystemUri":null, "ContactAddr":""," "ContactName":""," "ContactPhone":""," "State":0}, {"SysTenantryId":"SHTPV", "SysTenantryName":"冠捷科技", "CreateDate":"0001-01-01T00:00:00", "SystemTitle":"冠捷科技","SystemUri":null, "ContactAddr":""," "ContactName":""," "ContactPhone":""," "State":0}]</pre>
失败	HTTP 403 无访问权限

§ 3.3 查询项目列表

名称	查询项目列表
描述	查询当前用户有权访问的特定系统租户下的项目列表
方法	GET
请求	
URL	/Auth/Inquire/ ProjectList
参数	Schema = 指定的系统租户 ID
调用示例	http://localhost:1080/api/auth/inquire/ProjectList?Schema=SHNHZHCH &accesskey=web_7cd570a0fbae422a97e73be110f38c0f
响应	
成功	<p>允许操作的项目列表</p> <pre>[{ "projectId":"264f0e7b-bac8-4ff1-ad83-68cae952b3bd", "projectName":"上海智城研发基地", "projectAcronym":"NHZHCH", "sysTenantryId":"SHNHZHCH", "regionId":"CN101", "exchangeGateId":null, "latitude":121.6934608623, "longitude":31.0406011105, "areaOfStructure":0.0, "utilizationArea":0.0, "introduction":"“智城”有租金收入、管理服务收入、清洁能源收入等多元化盈利模式\r\n 记者日前从上海南汇工业园区投资发展有限公司获悉，作为浦东新区直属的开发公司，该公司在国有开发公司经营开发模式、激发国企活力方面正积极探索，在产业结构调整、经济提质增效升级方面积极寻求新的路径方法。 \r\n “近阶段，园区以产业综合体‘智城’为引领，在加快推进园区产业转型、能级提升方面，迈</pre>

	<p>出了坚实的步伐。”上海南汇工业园区投资发展有限公司总经理奚志忠表示，“智城”有不同于传统意义上的厂房或办公楼租售模式，有多元化的盈利点，包括土地溢价及部分楼宇出售收入、租金收入、管理服务收入、清洁能源收入和咨询顾问收入等。</p> <p>“智城”成为运营成本洼地</p> <p>“‘智城’的土地是开发公司之前以较低的成本取得的。南汇工业园区前几年的土地成本还比较低。随着两区合并，南汇工业园区划入金桥经济技术开发区，加上上海自贸区的成立、迪士尼项目的开工建设，以及浦东加快推动中部城镇带崛起的发展战略，处于浦东中部的南汇工业园区有了一定的区位优势，土地溢价趋势明显。”奚志忠表示，随着轨交 16 号线的通车，以及“智城”配套齐全的生活办公，使其成了中小企业运营成本洼地，“这些企业选择‘智城’作为创业办公的地点，具有较高的性价比，也是城郊商务园主动承接城市中心中小型企业总部转移的必然趋势”。</p> <pre>"timestamp": "2015-08-07T13:12:51Z", "comment": "" }]</pre>
失败	<p>HTTP 403</p> <p>无访问权限</p>

四、公共与基本数据操作 API

基本信息分为①所有系统租户公用的全局性信息，以及②指定系统租户专有的局部性信息。

下面的 Common 下的是“全局性信息操作 API”；

下面的 Basis 下的是“特定系统租户的局部基本信息操作 API”；

§ 4.1 标准代码(CODE)类别列表

系统内部允许用户单选或多选的项目，均通过本代码类别及类别下的明细行提供。

每个类别下的编码明细在不同的系统租户下自行定义。

本系统也同时提供了部分的基本代码条目。

名称	查询标准代码 TCodeCategory 类别表
描述	TCodeCategory 代码类别定义。用于管理设备的各个分类项、或各种可选值的列表
方法	GET
请求	
URL	/Common/TCodeCategories
参数	
示例	http://localhost:1080/api/Common/TCodeCategories?accesskey=web_7cd570a0fb4e422a97e73be110f38c0f
响应	
成功	<p>HTTP 200/OK 返回所有标准类别</p> <pre>[{"tCodeCategoryId": "T01", "tCodeCategoryName": "使用状况", "tCodeCategoryCaption": "使用状况", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": true, "comment": ""}, ...</pre>

	<pre>{ "tCodeCategoryId": "T02", "tCodeCategoryName": "维修级别", "tCodeCategoryCaption": "维修级别", "flowCategoryId": 1, "allowMultiSelect": true, "enabled": true, "comment": ""}, { "tCodeCategoryId": "T03", "tCodeCategoryName": "保养级别", "tCodeCategoryCaption": "保养级别", "flowCategoryId": 2, "allowMultiSelect": true, "enabled": true, "comment": ""}, { "tCodeCategoryId": "T04", "tCodeCategoryName": "检验级别", "tCodeCategoryCaption": "检验级别", "flowCategoryId": 3, "allowMultiSelect": true, "enabled": true, "comment": ""}, { "tCodeCategoryId": "T05", "tCodeCategoryName": "故障类别", "tCodeCategoryCaption": "故障类别", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": true, "comment": ""}, { "tCodeCategoryId": "T06", "tCodeCategoryName": "缺陷等级", "tCodeCategoryCaption": "缺陷等级", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": true, "comment": ""}, { "tCodeCategoryId": "T07", "tCodeCategoryName": "计量单位", "tCodeCategoryCaption": "计量单位", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": true, "comment": ""}, { "tCodeCategoryId": "T08", "tCodeCategoryName": "评价缘由", "tCodeCategoryCaption": "评价缘由", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": true, "comment": ""} }, { "tCodeCategoryId": "T09", "tCodeCategoryName": "未定义", "tCodeCategoryCaption": "未定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T10", "tCodeCategoryName": "未定义", "tCodeCategoryCaption": "未定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T20", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T21", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T22", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T23", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T24", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T25", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T26", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T27", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T28", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""}, { "tCodeCategoryId": "T29", "tCodeCategoryName": "用户自定义", "tCodeCategoryCaption": "用户自定义", "flowCategoryId": 0, "allowMultiSelect": true, "enabled": false, "comment": ""} } } } } } } } } } } } } } } } } } }</pre>
失败	HTTP 403 无访问权限

§ 4.2 代码(TCODE)列表

名称	查询标准代码 TCODE 类别表
描述	这些具体编码由系统租户自行定义。

	为减轻用户负担，本系统发布时也同时提供了部分常用的基本代码条目。
方法	GET
请求	
URL	/Basis/TCodes/{Schema}/{id}
参数	Schema → 指定系统租户 ID Id → 代码类别 ID (TCodeCategoryId)
示例	<i>http://localhost:1080/api/basis/TCodes/SHNHZHC/101?accesskey=web_7cd570a0fbae422a97e73be110f38c0f</i> <i>http://localhost:1080/api/basis/TCodes/SHNHZHC?id=101?accesskey</i>
响应	
成功	<p>HTTP 200/OK 返回所有标准类别</p> <pre>[{"tCodeId":8,"tCodeName":"未启用","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":9,"tCodeName":"在用","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":10,"tCodeName":"检修","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":11,"tCodeName":"出租","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":12,"tCodeName":"报废","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":13,"tCodeName":"封存","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":14,"tCodeName":"停用","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":15,"tCodeName":"备用","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":true,"comment":""}, {"tCodeId":16,"tCodeName":"暂不使用","tCodeValue":0,"tCodeCategoryId":"T01","tCodeIndex":0,"readOnly":false,"comment":"2"}]</pre>
失败	<p>HTTP 403</p> <p>无访问权限</p>

§ 4.3 标准类别-查询

名称	查询设备分类列表
描述	查找并返回所有标准类别列表、或指定 Id 的单个类别
方法	GET
请求	
URL	/Common/Categories /Common/Categories?id=xxx
参数	
示例	<i>http://localhost:1080/api/Common/Categories?accesskey=web_7cd570a0fbae422a97e73be110f38c0f</i> <i>http://localhost:1080/api/Common/Categories/A1</i>
响应	
成功	<p>HTTP 200/OK 返回所有标准类别列表或单个类别</p> <pre>[{"categoryId":"A1", "categoryName":"智能化工程",</pre>

	<pre>"categoryNameEN":""," "categoryParentId":null, "categoryAcronym":null, "categoryIconUri":""," "isLocking":true, "comment":""," "isMeter":false}, { "categoryId": "A1-010", "categoryName": "通讯网络", "categoryNameEN": "", "categoryParentId": "A1", "categoryAcronym": null, "categoryIconUri": "", "isLocking": false, "comment": "", "isMeter": false},]</pre>
失败	HTTP 403 无访问权限

§ 4.4 标准类别-属性查询

名称	属性查询
描述	查询指定设备分类类别下的属性列表
方法	GET
请求	
URL	/Common/CategoriesAttrs/{id}
参数	Id ➔ 指定的设备分类 Id，全称 “CategoryId” AttrId ➔ (可选参数)，仅返回指定属性 Id 的内容
示例	<i>http://localhost:1080/api/Common/CategoriesAttrs/M1-010</i> <i>http://localhost:1080/api/Common/CategoriesAttrs/M1-010?attrid=M1-010-001</i>
响应	
成功	HTTP 200/OK 返回指定 CategoryId 下的标准类别属性 [{ "categoryAttrId": "M1-010-001", "categoryAttrName": "通讯地址", "categoryAttrNameEN": "Addr", "categoryId": "M1-010", "categoryAttrType": 0, "itemType": 0, "dataType": 4,

	<pre>"unitName":null, "unitCoef":1.0, "isUsingRange":false, "isDynamic":false, "isIndirect":false, "isAccumulation":false, "isLocking":true, "comment":"Modbus/MBus 的地址码" }, 其它属性...]</pre>
失败	HTTP 403 无访问权限

§ 4.5 标准流程类型

名称	查询系统预定义的流程类型表
描述	
方法	GET
请求	
URL	/Common/ WorkflowCategories
参数	
示例	http://localhost:1080/api/Common/WorkflowCategories
响应	
成功	<p>HTTP 200/OK 返回</p> <pre>[{"workFlowCategory":0,"workFlowCategoryName":"未定义","comment":""}, {"workFlowCategory":1,"workFlowCategoryName":"服务","comment":"应求性服务"}, {"workFlowCategory":2,"workFlowCategoryName":"维修","comment":""}, {"workFlowCategory":3,"workFlowCategoryName":"保养","comment":""}, {"workFlowCategory":4,"workFlowCategoryName":"点检","comment":""}, {"workFlowCategory":5,"workFlowCategoryName":"巡检","comment":""}, {"workFlowCategory":6,"workFlowCategoryName":"检定","comment":"检验/检定/认证"}, {"workFlowCategory":7,"workFlowCategoryName":"采购","comment":""}, {"workFlowCategory":8,"workFlowCategoryName":"领用","comment":"领用/领取/领料"}, {"workFlowCategory":9,"workFlowCategoryName":"通知","comment":"用于合同到期等的自动提醒"}]</pre>
失败	HTTP 403 无访问权限

§ 4.6 标准行为或场所、阶段的编码

名称	查询行为或场所、阶段的编码表
描述	查询并返回系统内建的行为或场所、阶段的编码表
方法	GET
请求	
URL	/Common/Workstages
参数	
示例	http://localhost:1080/api/Common/Workstages
响应	
成功	<p>HTTP 200/OK 返回指定 CategoryId 下的标准类别属性</p> <pre>[{"stageId":0,"stageName":"未定义","themeColor":0,"comment":null}, {"stageId":1,"stageName":"准备中","themeColor":0,"comment":"创建工单/准备执行"}, {"stageId":2,"stageName":"流程开始","themeColor":0,"comment":null}, {"stageId":3,"stageName":"自动派单","themeColor":0,"comment":"一般自动派单和手工派单二选一"}, {"stageId":4,"stageName":"派单","themeColor":0,"comment":"一般自动派单和手工派单二选一"}, {"stageId":5,"stageName":"执行人接单","themeColor":0,"comment":null}, {"stageId":6,"stageName":"执行人到达现场","themeColor":0,"comment":"准备开工"}, {"stageId":7,"stageName":"执行工单","themeColor":0,"comment":null}, {"stageId":8,"stageName":"执行反馈","themeColor":0,"comment":null}, {"stageId":9,"stageName":"客户回访","themeColor":0,"comment":"可选流程节点"}, {"stageId":10,"stageName":"评价","themeColor":0,"comment":"领导评价、执行人自我评价"}, {"stageId":81,"stageName":"事件升级一","themeColor":0,"comment":"工作响应时间超长则升级"}, {"stageId":82,"stageName":"事件升级二","themeColor":0,"comment":"工作完成超时则升级"}, {"stageId":99,"stageName":"流程结束","themeColor":0,"comment":null}]</pre>
失败	<p>HTTP 403</p> <p>无访问权限</p>

§ 4.7 标准工作阶段编码表

名称	标准工作阶段编码表
描述	查询并返回维保与服务规范表所使用的工作阶段或流程的编码表
方法	GET
请求	
URL	/Common/JobStep
参数	
示例	http://localhost:1080/api/Common/JobStep/List?accesskey=web_7cd570a0fbae422a97e73be110f38c0f
响应	

成功	<p>HTTP 200/OK 返回指定 JobStep 下的标准工作阶段编码</p> <pre>[{"jobStepId":0,"jobStepName":"未定义","jobStepNameEN":"UnDefined","dataType":0,"themeColor":0,"comment":null}, {"jobStepId":10,"jobStepName":"方法","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":11,"jobStepName":"标准","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":12,"jobStepName":"工具","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":13,"jobStepName":"耗材","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null},{ "jobStepId":15,"jobStepName":"作业内容","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":20,"jobStepName":"作业流程","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":21,"jobStepName":"作业流程要点","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":22,"jobStepName":"安全措施","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":30,"jobStepName":"安全措施","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null},{ "jobStepId":40,"jobStepName":"故障代码","jobStepNameEN":"","dataType":3,"themeColor":0,"comment":null}, {"jobStepId":41,"jobStepName":"故障名称","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":42,"jobStepName":"现象描述","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":43,"jobStepName":"原因分析","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":50,"jobStepName":"责任部门","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":"从 50 到 59 为固定编号, 不允许修改"}, {"jobStepId":51,"jobStepName":"责任工作组","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":52,"jobStepName":"责任人","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":53,"jobStepName":"辅助部门","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":54,"jobStepName":"辅助工作组","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":55,"jobStepName":"辅助人","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":56,"jobStepName":"监督部门","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":57,"jobStepName":"监督工作组","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":58,"jobStepName":"监督人","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":59,"jobStepName":"执行位置","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}, {"jobStepId":999,"jobStepName":"备注","jobStepNameEN":null,"dataType":0,"themeColor":0,"comment":null}]</pre>
失败	<p>HTTP 403</p> <p>无访问权限</p>

§ 4.8 系统租户专用配置项

在每个系统租户内，拥有大量的配置项。这些配置项的 ID 在全球唯一，但允许用户修改配置值。

名称	查询系统租户专用配置
描述	查询全部或部分配置项
方法	GET
示例	<pre>http://localhost:1080//api/Basis/Configitems/SHNHZHCH http://localhost:1080//api/Basis/Configitems/SHNHZHCH?id=xxx http://localhost:1080//api/Basis/Configitems/SHNHZHCH?from=1000&to=10999</pre>
请求	
URL	/Basic/Configitems/{schema}?Id=xxx

参数	Schema ➔ 指定系统租户 ID Id ➔ （任选）配置项编号 From,To ➔ （任选）需要查询的配置项编号范围
响应	
成功	<p>HTTP 200/OK 返回所有配置项，或者指定 Id、或指定编号范围的配置项</p> <pre>[{ "ConfigItemId": 100, "ConfigItemName": "每周的第一天", "ConfigItemValue": "2", "NoGeneraterRoot": null, "Comment": "周一至日是 0..6" }, { "ConfigItemId": 101, "ConfigItemName": "每月的第一天", "ConfigItemValue": "26", "NoGeneraterRoot": null, "Comment": "取值范围 1..31" }, { "ConfigItemId": 200, "ConfigItemName": "维保提醒提前天数", "ConfigItemValue": "7", "NoGeneraterRoot": null, "Comment": "一般提前 1 周，允许 1..31 天" }, { "ConfigItemId": 301, "ConfigItemName": "维修计划单号", "ConfigItemValue": "WP {0} {1} {2} {3:yy-MM-dd} {4:0000}", "NoGeneraterRoot": null, "Comment": "{0} 项目; {1} 部门; {2} 工段; {3} 日期; {4} 流水号" }, { "ConfigItemId": 302, "ConfigItemName": "维修工单编号", "ConfigItemValue": "WX {0} {1} {2} {3:yyyyMMdd} {4:0000}", "NoGeneraterRoot": null, "Comment": "{0} 项目; {1} 部门; {2} 工段; {3} 日期; {4} 流水号" }, { "ConfigItemId": 303, "ConfigItemName": "保养计划单号", "ConfigItemValue": "BP {0} {1} {2} {3:yyyyMMdd} {4:0000}", "NoGeneraterRoot": null, "Comment": "{0} 项目; {1} 部门; {2} 工段; {3} 日期; {4} 流水号" },]</pre>

	<pre>{ "ConfigItemId": 304, "ConfigItemName": "保养工单编号", "ConfigItemValue": "BY{0} {1} {2} {3:yyyyMMdd} {4:0000}", "NoGeneratorRoot": null, "Comment": "{0}项目; {1}部门; {2}工段; {3}日期; {4}流水号" }, { "ConfigItemId": 305, "ConfigItemName": "检验计划单号", "ConfigItemValue": "JP{0} {1} {2} {3:yyyyMMdd} {4:0000}", "NoGeneratorRoot": null, "Comment": "{0}项目; {1}部门; {2}工段; {3}日期; {4}流水号" }, { "ConfigItemId": 306, "ConfigItemName": "检验工单编号", "ConfigItemValue": "JYZ{0} {2} {3:yyyyMMdd} {4:0000}", "NoGeneratorRoot": null, "Comment": "{0}项目; {1}部门; {2}工段; {3}日期; {4}流水号" }, { "ConfigItemId": 307, "ConfigItemName": "工程类工单编号", "ConfigItemValue": "GC{0} {1} {3:yy-MM-dd} {4:0000}", "NoGeneratorRoot": null, "Comment": "{0}项目; {1}部门; {2}工段; {3}日期; {4}流水号" }, { "ConfigItemId": 308, "ConfigItemName": "服务类工单编号", "ConfigItemValue": "FW{0} {1} {3:yyyyMMdd} {4:0000}", "NoGeneratorRoot": null, "Comment": "{0}项目; {1}部门; {2}工段; {3}日期; {4}流水号" } }</pre>
失败	HTTP 403 无访问权限

§ 4.9 区域-列表

名称	查询区域列表
描述	区域一般就是大楼，或大楼的部分区域
方法	GET

示例	<i>http://localhost:1080//api/Basis/Building/List/SHNHZHC?id=0</i> <i>http://localhost:1080//api/Basis/Building/List/SHNHZHC?id=264f0e7b-bac8-4ff1-ad83-68cae952b3bd</i>
请求	
URL	/Basic/Building/List/{schema}?Id=xxx
参数	Schema ➔ 指定系统租户 ID Id ➔ 项目 ID（如 ID(PrjId)为 0，则返回指定系统租户下的所有区域）
响应	
成功	<p>HTTP 200/OK 返回指定 ProjectId 下的所有区域</p> <pre>[{ "BuildingId": "018be6c3-2627-4ebf-af18-ab1ac764c6a5", "BuildingName": "1 号楼", "BuildingCode": null, "BuildingAcronym": null, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd", "ApportionCoefficient": 0.0, "Latitude": 0.0, "Longitude": 0.0, "Timestamp": "2015-07-15T12:58:58Z", "Comment": null, "Sort": 1 }, { "BuildingId": "026a96c6-2f00-494d-973b-4a39aeaf4bfe", "BuildingName": "17 号楼", "BuildingCode": null, "BuildingAcronym": null, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd", "ApportionCoefficient": 0.0, "Latitude": 0.0, "Longitude": 0.0, "Timestamp": "2015-07-15T12:59:45Z", "Comment": null, "Sort": 17 }]</pre>
失败	HTTP 403 无访问权限

§ 4.10 楼层

名称	查询指定区域的楼层列表
描述	
方法	GET

示例	<code>http://localhost:1080//api/Basis/Building/Floors/SHNHZHC?id=d62f4eec-b89e-46db-84da-6e758a64a740</code>
请求	
URL	<code>/Basic/Building/Floors/{schema}?Id=xxx</code>
参数	Schema ➔ 指定系统租户 ID Id ➔ 区域 ID
响应	
成功	<p>HTTP 200/OK 返回指定 区域 ID 下的所有楼层</p> <pre>[{ "BuildingFloorId": "2049336e-3067-4d3f-adaf-2bf7f696f9d2", "BuildingId": "d62f4eec-b89e-46db-84da-6e758a64a740", "BuildingFloorNo": 1, "BuildingFloorName": "1 楼", "Timestamp": "2015-07-28T12:19:10Z", "Comment": null }, { "BuildingFloorId": "2a82e2dc-e809-45d7-99c0-0239c08ff7c3", "BuildingId": "d62f4eec-b89e-46db-84da-6e758a64a740", "BuildingFloorNo": 4, "BuildingFloorName": "4 楼", "Timestamp": "2015-07-28T12:19:10Z", "Comment": null }, { "BuildingFloorId": "75e6ec31-7cd8-4b0e-b161-736e244eb110", "BuildingId": "d62f4eec-b89e-46db-84da-6e758a64a740", "BuildingFloorNo": 2, "BuildingFloorName": "2 楼", "Timestamp": "2015-07-28T12:19:10Z", "Comment": null }, { "BuildingFloorId": "a7fd32a8-89df-464a-b114-fbebd9cef031", "BuildingId": "d62f4eec-b89e-46db-84da-6e758a64a740", "BuildingFloorNo": 3, "BuildingFloorName": "3 楼", "Timestamp": "2015-07-28T12:19:10Z", "Comment": null }, { "BuildingFloorId": "ce71d70c-feb7-491c-b6df-b473b147dec5", "BuildingId": "d62f4eec-b89e-46db-84da-6e758a64a740", "BuildingFloorNo": 5, "BuildingFloorName": "5 楼", "Timestamp": "2015-07-28T12:19:10Z", "Comment": null }]</pre>

	<pre>}]</pre>
失败	HTTP 403 无访问权限

§ 4.11 单元

名称	查询指定范围内的单元列表
描述	根据提供的参数，可以分别查询并返回区域中的单元、楼层中的单元列表
方法	GET
示例	<code>http://localhost:1080//api/Basis/Building/Apartments/SHNHZCH?buildingid=2bd8b5eb-49f8-46e6-a722-ecf8cedf9b33</code> <code>http://localhost:1080//api/Basis/Building/Apartments/SHNHZCH?id=767e6f4e-0b09-420c-9bb4-663a692fcbea</code>
请求	
URL	<code>/Basis/Building/Apartments/{schema}?Id=xxx&BuildingId=xxx</code>
参数	Schema ➔ 指定系统租户 ID Id ➔ (可选参数) 楼层 ID BuildingId ➔ (可选参数) 区域或建筑 ID 注意：区域和楼层参数至少提供一个
响应	
成功	HTTP 200/OK 返回指定 区域 ID 或楼层下的所有单元 <pre>[{ "ApartmentId": "09f595dc-1f59-4adc-bb8f-29fec9ba6339", "ApartmentName": "UPS 室", "BuildingId": "2bd8b5eb-49f8-46e6-a722-ecf8cedf9b33", "FloorNo": 1, "ApportionCoefficient": 0.0, "TenantryId": "00000000-0000-0000-0000-000000000000", "Timestamp": "2015-07-11T21:18:44Z", "Comment": null }, { "ApartmentId": "36c21dc8-5246-4bb1-bcd1-ff8aa04d2b06", "ApartmentName": "监控设备间", "BuildingId": "2bd8b5eb-49f8-46e6-a722-ecf8cedf9b33", "FloorNo": 1, "ApportionCoefficient": 0.0, "TenantryId": "00000000-0000-0000-0000-000000000000", "Timestamp": "2015-07-11T21:18:44Z", "Comment": null }, { "ApartmentId": "38594b0b-bc96-4aba-8b0e-3d32294002eb",</pre>

	<pre>"ApartmentName": "男厕所", "BuildingId": "2bd8b5eb-49f8-46e6-a722-ecf8cedf9b33", "FloorNo": 1, "ApportionCoefficient": 0.0, "TenantryId": "00000000-0000-0000-0000-000000000000", "Timestamp": "2015-07-11T21:18:44Z", "Comment": null }, { "ApartmentId": "3890fab1-cc00-45c7-af09-3b89e5cc83ac", "ApartmentName": "前厅", "BuildingId": "2bd8b5eb-49f8-46e6-a722-ecf8cedf9b33", "FloorNo": 1, "ApportionCoefficient": 0.0, "TenantryId": "00000000-0000-0000-0000-000000000000", "Timestamp": "2015-07-11T21:18:44Z", "Comment": null }]</pre>
失败	HTTP 403 无访问权限

§ 4.12 设备

名称	查询指定范围内的设备列表
描述	可以分别按设备类别、子类等筛选； 也可以按设备所安装的位置（区域、楼层、单元）筛选
方法	GET
示例	<pre>http://localhost:1080///api/Basis/Equipment/List/SHTPV ➔ 返回全部设备(仅调试期间允许) http://localhost:1080///api/Basis/Equipment/List/SHTPV?id=603 http://localhost:1080///api/Basis/Equipment/List/SHTPV?CategoryId=M1-010 http://localhost:1080///api/Basis/Equipment/List/SHTPV?ApartmentId=e560856a-01bc-4a13-bed4-bf35c1ef863a</pre>
请求	
URL	/Basis/ Equipment/ {schema}?Id=xxx&BuildingId=xxx
参数	<p>Schema ➔ 指定系统租户 ID</p> <p>Id ➔ 设备编号，用于返回一个特定设备的明细信息</p> <p>BuildingId ➔ (可选参数)区域或建筑 ID</p> <p>FloorId ➔ (可选参数)楼层 ID</p> <p>ApartmentId ➔ (可选参数)单元 ID</p> <p>CategoryId ➔ 设备所属类别编码</p> <p>注意：上述区域、楼层、单元、设备类别等参数至少提供一个</p>
响应	
成功	HTTP 200/OK 返回指定 区域或类别 的所有设备清单

```
[
  {
    "EquipmentNo": 602,
    "EquipmentName": "DM-1AA7-A",
    "EquipmentNum": "B3AP3",
    "Position": "1AA7-A",
    "ParentEquipmentNo": 0,
    "CategoryId": "M1-010",
    "DisplayIndex": 0,
    "Specification": "DM2300",
    "PermitNumber": "",
    "AssetNumber": "",
    "ManufacturerUnitId": "bbc71fa7-2a83-4fbb-b8ae-4cb0bf567c20",
    "ManufacturerUnitName": "冠捷科技-上海虹桥商务区总部",
    "ManufacturerModel": "",
    "MaintainUnitId": "bbc71fa7-2a83-4fbb-b8ae-4cb0bf567c20",
    "MaintainUnitName": "冠捷科技-上海虹桥商务区总部",
    "SupplyUnitId": "bbc71fa7-2a83-4fbb-b8ae-4cb0bf567c20",
    "SupplyUnitName": "冠捷科技-上海虹桥商务区总部",
    "ConstructionUnitId": "bbc71fa7-2a83-4fbb-b8ae-4cb0bf567c20",
    "ConstructionUnitName": "冠捷科技-上海虹桥商务区总部",
    "OperatingUnitId": "bbc71fa7-2a83-4fbb-b8ae-4cb0bf567c20",
    "OperatingUnitName": "冠捷科技-上海虹桥商务区总部",
    "AcceptanceCheckDate": "2015-01-01T00:00:00",
    "QualityAssuranceDays": 12,
    "ApartmentId": "e560856a-01bc-4a13-bed4-bf35c1ef863a",
    "OrgId": "bbc71fa7-2a83-4fbb-b8ae-4cb0bf567c20",
    "QRCode": null,
    "ParentMeterNo": 0,
    "BillingApportion": 0,
    "IsVirtual": false,
    "T01": null,
    "BlueprintNo": "",
    "BlueprintInnerNo": "",
    "DocumentCount": 0,
    "WikiUrl": null,
    "ImgUrl": null,
    "Timestamp": "2015-07-02T14:43:32Z",
    "Comment": "B3F 诱导风机"
  },
  {
    "EquipmentNo": 603,
    "EquipmentName": "DM-1AA7-B",
    "EquipmentNum": "B2AP2",
    "Position": "1AA7-B",
    "ParentEquipmentNo": 0,
    "CategoryId": "M1-010",
    "DisplayIndex": 0,
    "Specification": "DM2300",
```


	<pre>"PermitNumber": "", "AssetNumber": "", "ManufacturerUnitId": "00000000-0000-0000-0000-000000000000", "ManufacturerUnitName": null, "ManufacturerModel": null, "MaintainUnitId": "00000000-0000-0000-0000-000000000000", "MaintainUnitName": "施耐德", "SupplyUnitId": "00000000-0000-0000-0000-000000000000", "SupplyUnitName": null, "ConstructionUnitId": "00000000-0000-0000-0000-000000000000", "ConstructionUnitName": null, "OperatingUnitId": "00000000-0000-0000-0000-000000000000", "OperatingUnitName": null, "AcceptanceCheckDate": "0001-01-01T00:00:00", "QualityAssuranceDays": 0, "ApartmentId": "e560856a-01bc-4a13-bed4-bf35c1ef863a", "OrgId": "bbc71fa7-2a83-4fbb-b8ae-4cb0bf567c20", "QRCode": null, "ParentMeterNo": 605, "BillingApportion": 0, "IsVirtual": false, "T01": null, "BlueprintNo": "", "BlueprintInnerNo": "", "DocumentCount": 0, "WikiUrl": null, "ImgUrl": null, "Timestamp": "2015-05-19T13:58:55Z", "Comment": "B2F 空调通风" }]</pre>
失败	HTTP 403 无访问权限

§ 4.13 联系单位-列表

名称	联系人单位-列表
描述	获取联系人单位列表。 当提供 <code>Id</code> 参数时， <code>Id</code> 表示上级组织的编码。此时可列出下一级组织列表
方法	GET
示例	<code>http://api.losta.net/api/Basis/Org/List/SHNHZHC?id=00&accesskey=web_7cd570a0fbae422a97e73be110f38c0f</code>
请求	
URL	<code>/Basis/Org/List/{ Schema}?id=xxxx</code>
参数	Schema ➔ 指定系统租户 ID

	<div>Id → 上级单位 Id，如果为 0，表示顶级单位</div> <div>PageNo → 分页页号（默认 1）</div> <div>PageSize → 每页记录数(默认 50)</div>
响应	
成功	<div>HTTP 200/OK 返回指定 Schema / Schema 下的联系单位列表</div> <div>[{"orgId": "0fc4de35-05c0-40b0-8fdc-e5ddd01cafbb", "orgName": "上海两港投资发展有限公司", "orgAcronym": "两港发展", "parentOrgId": "00000000-0000-0000-0000-000000000000", "managerContactId": "fcd7bcfa-0999-4c6a-8985-f94cebc407b9", "phoneNumber": "", "faxNumber": "", "address": "上海市沪南公路 8666 号 19 号楼", "webSite": "", "documentCount": 0, "belongSysTenantry": false, "comment": ""}, {"orgId": "95a9f9cf-36b5-4222-b869-1298903ff873", "orgName": "测试保安服务商", "orgAcronym": "保安", "parentOrgId": "00000000-0000-0000-0000-000000000000", "managerContactId": "00000000-0000-0000-0000-000000000000", "phoneNumber": "", "faxNumber": "", "address": "", "webSite": "", "documentCount": 0, "belongSysTenantry": false, "comment": ""}]</div>
失败	<div>HTTP 403</div> <div>无访问权限</div>

§ 4.14 联系单位-详情

名称	针对一个联系单位(或部门)的操作
描述	查询、新增、修改、删除指定联系单位（往来单位、或下级单位）
方法	GET 查询 PUT 新建 POST 修改 DELETE 删除
示例	

请求	
URL	/Basis/Org/Units/{Schema}?id=xxxx
参数	Schema ➔ 指定系统租户 ID Id ➔ 往来单位 ID（一般就是 OrgId）
响应	
成功	<p>HTTP 200/OK 如果查询、新增或修改成功，则返回当前组织的明细信息</p> <pre>{ "OrgId": "95a9f9cf-36b5-4222-b869-1298903ff873", "OrgName": "测试保安服务商", "OrgAcronym": "保安", "ParentOrgId": "00000000-0000-0000-0000-000000000000", "ManagerContactId": "00000000-0000-0000-0000-000000000000", "PhoneNumber": "", "FaxNumber": "", "Address": "", "WebSite": "", "DocumentCount": 0, "BelongSysTenantry": false, "Comment": "" }</pre> <p>HTTP 200/OK 如果删除成功，则返回成功删除的记录数</p> <p>1</p>
失败	<p>HTTP 403</p> <p>无访问权限</p>
完整示例	
	<p>1、查询往来单位详情（GET 方法）</p> <p>http://localhost:1080/api/Basis/Org/Units/shnhzhch/95a9f9cf-36b5-4222-b869-1298903ff873</p> <p>http://localhost:1080/api/Basis/Org/Units/shnhzhch?id=95a9f9cf-36b5-4222-b869-1298903ff873</p> <p>返回往来单位详情（略）</p> <p>2、新增往来单位（PUT 方法）</p> <p>http://localhost:1080/api/Basis/Org/Units/shnhzhch</p> <p>PUT 参数(新增一个往来单位):</p> <pre>{ "OrgId": "95a9f9cf-36b5-4222-b869-1298903ff873", "OrgName": "测试保安服务商", "OrgAcronym": "保安", "ParentOrgId": "00000000-0000-0000-0000-000000000000", "ManagerContactId": "00000000-0000-0000-0000-000000000000", "PhoneNumber": "", "FaxNumber": "", "Address": "", "WebSite": "", "DocumentCount": 0, "BelongSysTenantry": false,</pre>

	<pre>"Comment": "" }</pre> <p>新增后返回所有结果（略）</p> <p>如果与现存记录冲突，则返回错误信息：</p> <pre>{ "ErrorCode": 409, "MessageDetail": "新增记录时发生错误，可能是有重复的设备号" }</pre> <p>3、修改网络单位信息（POST 方法），仅用于修改</p> <p>http://localhost:1080/api/Basis/Org/Units/shnhzhch?id=95a9f9cf-36b5-4222-b869-1298903ff873</p> <p>POST 参数(修改一行设备信息)：</p> <pre>{ "OrgId": "95a9f9cf-36b5-4222-b869-1298903ff873", ➔ 必须与Url 中的Id 一致 "OrgName": "测试保安服务商的新名字", ➔ 假设这次修改了新名字 "OrgAcronym": "保安", "ParentOrgId": "00000000-0000-0000-0000-000000000000", "ManagerContactId": "00000000-0000-0000-0000-000000000000", "PhoneNumber": "", "FaxNumber": "", "Address": "", "WebSite": "", "DocumentCount": 0, "BelongSysTenantry": false, "Comment": "" }</pre> <p>修改成功后返回结果（略）</p> <p>4、删除设备（DELETE 方法）</p> <p>http://localhost:1080/api/Basis/Org/Units/shnhzhch?id=95a9f9cf-36b5-4222-b869-1298903ff873</p>
--	--

§ 4.15 联系人-列表

名称	联系人列表
描述	获取指定往来单位下的联系人列表。
方法	GET
示例	http://localhost:1080/api/Basis/contact/List/SHTPV?id=00&accesskey=web_7cd570a0fbae422a97e73be110f38c0f
请求	
URL	/Basis/ Contact /List/{ Schema}?id=xxxx
参数	Schema ➔ 指定系统租户 ID Id ➔ 往来单位 OrgId，如果为 0，表示列出无单位的联系人 PageNo ➔ 分页页号（默认 1）

	PageSize ➡ 每页记录数(默认 50)
响应	
成功	<div>HTTP 200/OK 返回指定 Schema / OrgId 下的联系人列表</div> <div>[{"contactId": "29f1d414-c547-4170-b002-4cde132bfe60", "contactName": "tpvuser", "contactTitle": "先生", "orgId": "5603cc10-6f61-419f-9b8a-39333548b700", "orgName": "总经理办公室", "dutyTitle": "冠捷管理员", "userNo": "", "userId": "00000000-0000-0000-0000-000000000000", "skillList": "", "phoneNumber": "", "cellPhone": "13588996336", "email": "13588996336@qq.com", "wxOpenId": null, "qq": "", "address": "", "photoDocumentId": "c5ad98e9-0703-44e6-a108-a9b466403dce", "state": 0, "comment": "1"}], { "contactId": "e586f986-6bce-4758-8284-ccb7e21f8e4a", "contactName": "test111", "contactTitle": "女士", "orgId": "5603cc10-6f61-419f-9b8a-39333548b700", "orgName": "总经理办公室", "dutyTitle": "", "userNo": "", "userId": "00000000-0000-0000-0000-000000000000", "skillList": "", "phoneNumber": "", "cellPhone": "13588996336", "email": "", "wxOpenId": null, "qq": "", "address": "", "photoDocumentId": "00000000-0000-0000-0000-000000000000", "state": 0, "comment": ""}]</div>
失败	<div>HTTP 403</div> <div>无访问权限</div>

§ 4.16 联系人-详情

名称	针对一个联系人的操作
描述	查询、新增、修改、删除指定联系人
方法	GET 查询 PUT 新建 POST 修改 DELETE 删除
示例	
请求	
URL	/Basis/ Contact /persons/{ Schema}?id=xxxx
参数	Schema → 指定系统租户 ID Id → 联系人 Id（一般就是 ContactId）
响应	
成功	<p>HTTP 200/OK 如果查询、新增或修改成功，则返回指定联系人的明细信息</p> <pre>{ "ContactId": "f9f24575-f885-4ec7-a968-54dca5473056", "ContactName": "测试租户管理员", "ContactTitle": "先生", "OrgId": "6b726938-46c3-47cd-950e-62551c59b9b5", "OrgName": "测试租户", "DutyTitle": "", "UserNo": null, "UserId": "00000000-0000-0000-0000-000000000000", "SkillList": "", "PhoneNumber": "", "CellPhone": "", "Email": "", "WxOpenId": null, "QQ": "", "Address": "", "PhotoDocumentId": "00000000-0000-0000-0000-000000000000", "State": 0, "Comment": "" }</pre> <p>HTTP 200/OK 如果删除成功，则返回成功删除的记录数</p> <pre>{ "ErrorCode": 200, "Result": 1, "MessageDetail": "删除联系人成功" }</pre>
失败	HTTP 403 无访问权限
完整示例	
	1、查询联系人详情（GET 方法）

`http://localhost:1080/api/Basis/Contact/Persons/SHNHZCH/f9f24575-f885-4ec7-a968-54dca5473056`
`http://localhost:1080/api/Basis/Contact/person/shnhzhch?id=f9f24575-f885-4ec7-a968-54dca5473056`

返回联系人详情（略）

2、新增联系人（PUT 方法）

`http://localhost:1080/api/Basis/Contact/person/shnhzhch`

PUT 参数(新增一个联系人):

```
{
  "ContactId": "76E232F5-65F6-4C7D-815A-71256712258E", ➔ 新增时不提供或赋新唯一值
  "ContactName": "新增一个张三",
  "ContactTitle": "先生",
  "OrgId": "6b726938-46c3-47cd-950e-62551c59b9b5",
  "OrgName": "新增测试",
  "DutyTitle": "总经理",
  "UserNo": null,
  "UserId": "00000000-0000-0000-0000-000000000000",
  "SkillList": "",
  "PhoneNumber": "",
  "CellPhone": "138188888111",
  "Email": "",
  "WxOpenId": null,
  "QQ": "",
  "Address": "",
  "PhotoDocumentId": "00000000-0000-0000-0000-000000000000",
  "State": 0,
  "Comment": ""
}
```

新增后返回所有结果（略）

如果与现存记录冲突，则返回错误信息：

```
{
  "ErrorCode": 409,
  "MessageDetail": "新增记录时发生错误，可能是有重复的设备号"
}
```

3、修改联系人信息（POST 方法），仅用于修改

`http://localhost:1080/api/Basis/Contact/person/shnhzhch?id=76E232F5-65F6-4C7D-815A-71256712258E`

POST 参数(修改联系人信息):

```
{
  "ContactId": "76E232F5-65F6-4C7D-815A-71256712258E", ➔ 必须与 Url 中的值一致
  "ContactName": "张三新名字",
  "ContactTitle": "小姐",
  "OrgId": "6b726938-46c3-47cd-950e-62551c59b9b5",
  "OrgName": "新增测试",
  "DutyTitle": "总经理助理", ➔ 降级了 :- (
  "UserNo": null,
  "UserId": "00000000-0000-0000-0000-000000000000",
}
```

	<pre>"SkillList": "", "PhoneNumber": "", "CellPhone": ""138188888111, "Email": "", "WxOpenId": null, "QQ": "", "Address": "", "PhotoDocumentId": "00000000-0000-0000-0000-000000000000", "State": 0, "Comment": "" }</pre> <p>修改成功后返回结果（略）</p> <p>4、删除联系人（DELETE 方法）</p> <p>http://localhost:1080/api/Basis/Contact/Delete/shnhzhch?id=76E232F5-65F6-4C7D-815A-71256712258E</p>
--	---

§ 4.17 维保规范

§ 4.18 服务规范

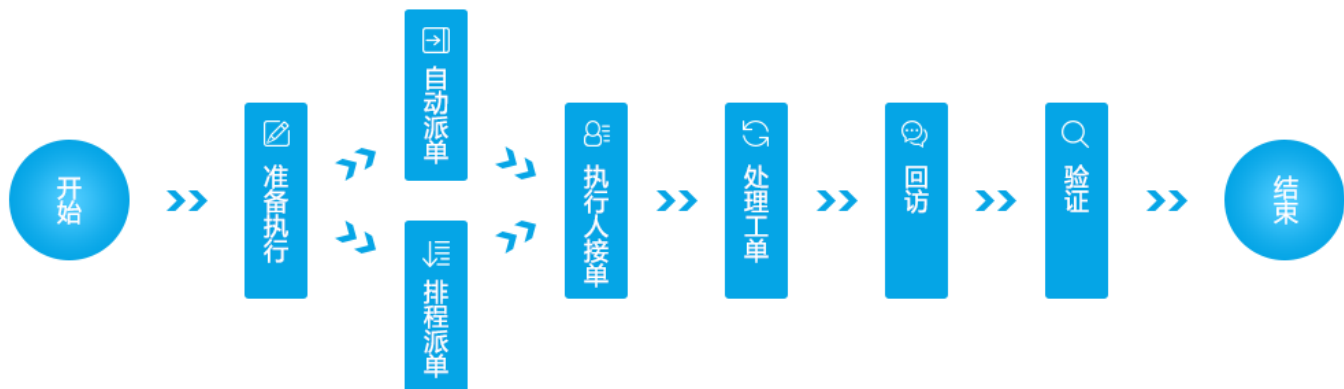
五、 工单与任务管理 API

任务管理总是局限于某个系统租户之下，不允许有跨越系统租户的工单存在。
工单的类型很多，系统内为了区分工单，从多个维度增加了类型编码。

- ✓ WorkflowCategory 工单的工作流类型定义（服务、维修、保养、点检、巡检、通知...）
- ✓ ServiceType 服务类型，二级编码，参见公共 API 部分
- ✓ OrderSource 工单来源（1 服务、2 巡检、3 应求工程、4 预防性工程、5 故障维修、6 强检）

工单的执行状态有两个

- ✓ CurrentStage 工单当前所处执行阶段或场景。一旦创建完成则处于 Ready 阶段（此时的 State 一般为 2、或 3）
- ✓ State 单据状态（0:创建中、1:创建完成、2:执行中、3:暂停、4:撤销、9:Finished）



§ 5.1 我的工单-列表(可指定条件)

名称	工单摘要列表
描述	查询当前用户作为管理者、或执行人的工单列表 可按照项目、日期范围等筛选 默认返回第一页最多 100 条记录，且以日期最新的为主
方法	GET
请求	
URL	~/Task/Sheets/{Schema}
参数	Schema → 指定系统租户 ID Begin → 开始时刻，例如'2015-07-30' End → 结束时刻，例如'2015-08-02' PageNo → 页号（默认 1） PageSize → 页大小（默认 100） State → 工单状态（可选参数，0:创建中、1:创建完成、2:执行中、3:暂停、4:撤销、9:Finished）
响应	
成功	HTTP 200/OK 返回符合要求的工单概要信息列表 <pre>[{ "SequenceNo": 11, "WorkorderNo": "WD000001", "OrderDate": "2015-07-30T14:38:00", "PlanDate": "2015-07-30T14:53:00", "CreateDate": "2015-07-30T14:38:56", "WorkFlowId": 0, "WorkFlowCategory": 0, "OrderSource": 1, "SourceWorkorderNo": null, "Priority": 1, "ServiceType": 100, "Period": "", "ResponseMinutes": 0.0, }]</pre>

```
"ExecuteMinutes": 0.0,
"ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd",
"BuildingId": "a1cd2357-a007-4bdf-9e19-fbca24clf387",
"BuildingFloorNo": 2,
"ApartmentId": "00000000-0000-0000-0000-000000000000",
"Subject": "11",
"Content": "22",
"Results": null,
"OrgId": "00000000-0000-0000-0000-000000000000",
"InputUserId": "675c5b30-330d-49b2-a32f-004e9fab79e",
"UserId": "00000000-0000-0000-0000-000000000000",
"UserName": "test",
"CurrentStage": 0,
"State": 4,
"Comment": null,
"ImgUrl": null
},
{
  "SequenceNo": 15,
  "WorkorderNo": "WD000005",
  "OrderDate": "2015-08-10T17:09:00",
  "PlanDate": "2015-08-10T17:24:00",
  "CreateDate": "2015-08-10T17:08:39",
  "WorkflowId": 0,
  "WorkflowCategory": 0,
  "OrderSource": 1,
  "SourceWorkorderNo": null,
  "Priority": 1,
  "ServiceType": 100,
  "Period": "",
  "ResponseMinutes": 0.0,
  "ExecuteMinutes": 0.0,
  "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd",
  "BuildingId": "54d19577-d08b-4a8d-a536-e51d0753d146",
  "BuildingFloorNo": 4,
  "ApartmentId": "00000000-0000-0000-0000-000000000000",
  "Subject": "11",
  "Content": "22",
  "Results": null,
  "OrgId": "00000000-0000-0000-0000-000000000000",
  "InputUserId": "9c08f187-c22b-4dc6-8008-e0fa68349a48",
  "UserId": "00000000-0000-0000-0000-000000000000",
  "UserName": "test",
  "CurrentStage": 0,
  "State": 1,
  "Comment": null,
  "ImgUrl": "6e6ac855-11de-214f-7ce6-e03305050505"
}
]
```

失败	HTTP 403 无访问权限

§ 5.2 我的工单-新建

名称	新建工单主体
描述	创建全新的工单
方法	PUT
请求	
URL	/Task/Sheets/{Schema}
参数	<pre>{ "SequenceNo": 0, "WorkorderNo": "WD000005", "OrderDate": "2015-08-10T17:09:00", "PlanDate": "2015-08-10T17:24:00", "CreateDate": "2015-08-10T17:08:39", "WorkFlowId": 0, "WorkFlowCategory": 0, "OrderSource": 1, "SourceWorkorderNo": null, "Priority": 1, "ServiceType": 100, "Period": "", "ResponseMinutes": 0.0, "ExecuteMinutes": 0.0, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd", "BuildingId": "54d19577-d08b-4a8d-a536-e51d0753d146", "BuildingFloorNo": 4, "ApartmentId": "00000000-0000-0000-0000-000000000000", "Subject": "11", "Content": "22", "Results": null, "OrgId": "00000000-0000-0000-0000-000000000000", "InputUserId": "9c08f187-c22b-4dc6-8008-e0fa68349a48", "UserId": "00000000-0000-0000-0000-000000000000", "UserName": "test", "CurrentStage": 0, "State": 1, "Comment": null, "ImgUrl": "6e6ac855-11de-214f-7ce6-e03305050505" }</pre> <div><p>→ 新的工单号，不允许重复。如留空，则由系统自行新建</p><p>→ 如果根据工作流创建，则需指定工作流 Id</p><p>→ 如果是执行其它工单过程中，因需创建本工单，则需记录前一个工单的号码</p><p>→ 服务类型，请按系统提供的内容设定</p><p>→ 如果是周期性执行的工单，则这里存放周期描述。格式参见后续章节</p></div>
响应	
成功	HTTP 200/OK 返回创建完成后的工单概要信息

失败	HTTP 203 表单已经存在 HTTP 403 无访问权限

§ 5.3 我的工单-删除

名称	删除工单
描述	仅当工单没有开始执行前，且当前用户作为管理者时允许删除工单
方法	DELETE
请求	
URL	/Task/Sheets/{ Schema }/Id
参数	Schema ➔ 指定系统租户 ID Id 工单编号 TradeDate (可选的)工单日期
响应	
成功	HTTP 200/OK 返回删除掉的工单概要信息
失败	HTTP 403 无访问权限

§ 5.4 我的工单-主体变更

名称	查询我的工单明细
描述	查询并返回指定编号工单的所有明细 当前用户作为管理者、或执行人的特定工单
方法	POST
请求	
URL	/Task/Sheets/{ Schema }
参数	参数类型通创建工单
响应	
成功	HTTP 200/OK 变更后到的工单主体信息
失败	HTTP 403 无访问权限

§ 5.5 我的工单-工单明细操作

§ 5.5.1 我的工单-工作组

名称	工单相关的工作组操作
描述	查询、新增、修改、删除指定工单的工作组条目
方法	GET 查询 PUT 新建 POST 修改 DELETE 删除
示例	
请求	
URL	/Task/SheetGroups/{Schema}?id=xxxx
参数	Schema ➔ 指定系统租户 ID Id ➔ 工单编号 workgroupId ➔ 工作组号，删除时需要提供本参数
响应	
成功	<p>HTTP 200/OK 如果查询、新增或修改成功，则返回当前工单内的所有工作组</p> <pre>[{ "WorkorderNo": "WD000001", "StageId": 1, "WorkgroupId": 1001 }, { "WorkorderNo": "WD000001", "StageId": 1, "WorkgroupId": 1002 }]</pre> <p>HTTP 200/OK 如果删除成功，则返回成功删除的记录数</p> <p>1</p>
失败	HTTP 403 无访问权限
完整示例	
	<p>1、查询工作组（GET 方法）</p> <p>http://localhost:1080/api/Task/SheetGroups/shnhzhch?id=WD000001</p> <p>http://localhost:1080/api/Task/SheetGroups/shnhzhch/WD000001</p> <p>2、新增工作组（PUT 方法）</p> <p>http://localhost:1080/api/Task/SheetGroups/shnhzhch/WD000001</p> <pre>[{ "WorkorderNo": "WD000001", "StageId": 1, "WorkgroupId": 1001 }, { "WorkorderNo": "WD000001", "StageId": 1, "WorkgroupId": 1002 }]</pre>

	<pre>] 3、修改工作组（POST 方法） http://localhost:1080/api/Task/SheetGroups/shnhzhch/WD000001 [{ "WorkorderNo": "WD000001", "StageId": 1, "WorkgroupId": 1001}, { "WorkorderNo": "WD000001", "StageId": 1, "WorkgroupId": 1002}] 4、删除工作组（DELETE 方法） http://localhost:1080/api/Task/SheetGroups/shnhzhch/WD000001?workgroupId=1001</pre>
--	---

§ 5.5.2 我的工单-工作成员(执行人)

名称	工单相关的执行人操作
描述	查询、新增、修改、删除指定工单的执行人条目
方法	GET 查询 PUT 新建 POST 修改 DELETE 删除
示例	
请求	
URL	/Task/SheetMembers/{Schema}?id=xxxx
参数	Schema ➔ 指定系统租户 ID Id ➔ 工单编号 userId ➔ 用户号，删除时需提供本参数
响应	
成功	<p>HTTP 200/OK 如果查询、新增或修改成功，则返回当前工单内的所有执行人信息</p> <pre>[{ "WorkorderNo": "WD000001", "WorkMemberRole": 1, "UserId": "15bc9d18-faa6-4e97-8a45-d4c76f52c0d7", "Grade": 0, "InformingMode": 0 }]</pre> <p>HTTP 200/OK 如果删除成功，则返回成功删除的记录数</p> <p>1</p>

失败	HTTP 403 无访问权限
完整示例	
	<p>1、查询执行人列表（GET 方法）</p> <p><i>http://localhost:1080/api/Task/SheetMembers/shnhzhch/WD000001</i> <i>http://localhost:1080/api/Task/SheetMembers/shnhzhch?id=WD000001</i></p> <p>返回执行人列表</p> <pre>[{ "WorkorderNo": "WD000001", "WorkMemberRole": 1, "UserId": "15bc9d18-faa6-4e97-8a45-d4c76f52c0d7", "Grade": 0, "InformingMode": 0 }, { "WorkorderNo": "WD000001", "WorkMemberRole": 1, "UserId": "81d51627-7d83-4449-8186-4ffba624655", "Grade": 0, "InformingMode": 0 }, { "WorkorderNo": "WD000001", "WorkMemberRole": 1, "UserId": "d77bce50-1bd7-4531-b31b-0990f26bd655", "Grade": 0, "InformingMode": 0 }]</pre> <p>2、新增执行人（PUT 方法）</p> <p><i>http://localhost:1080/api/Task/SheetMembers/shnhzhch/WD000001</i></p> <p>PUT 参数(新增一个执行人):</p> <pre>[{ "WorkorderNo": "WD000001", "WorkMemberRole": 1, "UserId": "d77bce50-1bd7-4531-b31b-0990f26bd655", "Grade": 1, "InformingMode": 2 }]</pre> <p>新增后返回所有现存执行人列表</p> <pre>[{ "WorkorderNo": "WD000001",</pre>

	<pre>"WorkMemberRole": 1, "UserId": "a669c949-6850-4fd9-a05a-c56b41cc7c13", "Grade": 1, "InformingMode": 2 }, { "WorkorderNo": "WD000001", "WorkMemberRole": 1, "UserId": "d77bce50-1bd7-4531-b31b-0990f26bd655", "Grade": 0, "InformingMode": 0 }]</pre> <p>如果与现存记录冲突，则返回错误信息：</p> <pre>{ "ErrorCode": 409, "MessageDetail": "新增记录时发生错误，可能是有重复的工号或角色" }</pre> <p>3、修改执行人（POST 方法），仅用于修改</p> <p><i>http://localhost:1080/api/Task/SheetMembers/shnhzhch/WD000001</i></p> <p>4、删除执行人（DELETE 方法）</p> <p><i>http://localhost:1080/api/Task/SheetMembers/shnhzhch/WD000001</i></p> <p><i>?userid= d77bce50-1bd7-4531-b31b-0990f26bd655</i></p>
--	--

§ 5.5.3 我的工单-设备明细

名称	工单相关的设备明细操作
描述	查询、新增、修改、删除指定工单的设备明细条目
方法	GET 查询 PUT 新建 POST 修改 DELETE 删除
示例	
请求	
URL	/Task/SheetEquipments/{Schema}?id=xxxx
参数	Schema ➔ 指定系统租户 ID Id ➔ 工单编号 EquipmentNo ➔ 设备号，删除时需要提供本参数
响应	
成功	HTTP 200/OK 如果查询、新增或修改成功，则返回当前工单内的所有设备信息 [{

	<pre>"SequenceNo": 1, "WorkorderNo": "WD000001", "StageId": 1, "EquipmentNo": 605, "Level": 1, "Item": "项目 1", "Method": "目视", "Standard": "国标", "ResultCode": 1, "Result": "正常", "DocumentId": "00000000-0000-0000-0000-000000000000" }]</pre> <p>HTTP 200/OK 如果删除成功，则返回成功删除的记录数</p> <p>1</p>
失败	HTTP 403 无访问权限
完整示例	
	<p>1、查询设备明细表（GET 方法）</p> <p><i>http://localhost:1080/api/Task/SheetEquipments/shnhzhch/WD000001</i> <i>http://localhost:1080/api/Task/SheetEquipments/shnhzhch?id=WD000001</i></p> <p>返回设备列表</p> <pre>[{ "SequenceNo": 1, "WorkorderNo": "WD000001", "StageId": 1, "EquipmentNo": 605, "Level": 1, "Item": "项目 1", "Method": "目视", "Standard": "国标", "ResultCode": 1, "Result": "正常", "DocumentId": "00000000-0000-0000-0000-000000000000" }]</pre> <p>2、新增设备（PUT 方法）</p> <p><i>http://localhost:1080/api/Task/SheetEquipments/shnhzhch/WD000001</i></p> <p>PUT 参数(新增一个设备人):</p> <pre>[{ "SequenceNo": 0, "WorkorderNo": "WD000001", "StageId": 1, "EquipmentNo": 601,</pre> <p>➔ 新增时，这里的 SequenceNo 为 0</p>

```
    "Level": 1,
    "Item": "项目 XXX",
    "Method": "目视 ZZZ",
    "Standard": "国标 BBB",
    "ResultCode": 1,
    "Result": "正常",
    "DocumentId": "00000000-0000-0000-0000-000000000000"
  }
]
```

新增后返回所有现存设备列表

```
[
  {
    "SequenceNo": 1,
    "WorkorderNo": "WD000001",
    "StageId": 1,
    "EquipmentNo": 605,
    "Level": 1,
    "Item": "项目 1",
    "Method": "目视",
    "Standard": "国标",
    "ResultCode": 1,
    "Result": "正常",
    "DocumentId": "00000000-0000-0000-0000-000000000000"
  },
  {
    "SequenceNo": 2,
    "WorkorderNo": "WD000001",
    "StageId": 1,
    "EquipmentNo": 601,
    "Level": 1,
    "Item": "项目 XXX",
    "Method": "目视 ZZZ",
    "Standard": "国标 BBB",
    "ResultCode": 1,
    "Result": "正常",
    "DocumentId": "00000000-0000-0000-0000-000000000000"
  }
]
```

如果与现存记录冲突，则返回错误信息：

```
{
  "ErrorCode": 409,
  "MessageDetail": "新增记录时发生错误，可能是有重复的设备号"
}
```

3、修改设备（POST 方法），仅用于修改

<http://localhost:1080/api/Task/SheetEquipments/shnhzhch/WD000001>

POST 参数(修改一行设备信息):

```
[
  {
    "SequenceNo": 2,
    "WorkorderNo": "WD000001",
    "StageId": 1,
    "EquipmentNo": 601,
    "Level": 1,
    "Item": "项目项目项目项目项目",
    "Method": "目视目视目视目视目视",
    "Standard": "国标 BBB",
    "ResultCode": 1,
    "Result": "正常",
    "DocumentId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd"
  }
]
```

→ 这是修改时所依据的主键，是记录的唯一性标识
→ 工单号不允许修改

修改成功后返回当前工单的最新设备明细表

```
[
  {
    "SequenceNo": 1,
    "WorkorderNo": "WD000001",
    "StageId": 1,
    "EquipmentNo": 605,
    "Level": 1,
    "Item": "项目 1",
    "Method": "目视",
    "Standard": "国标",
    "ResultCode": 1,
    "Result": "正常",
    "DocumentId": "00000000-0000-0000-0000-000000000000"
  },
  {
    "SequenceNo": 2,
    "WorkorderNo": "WD000001",
    "StageId": 1,
    "EquipmentNo": 601,
    "Level": 1,
    "Item": "项目项目项目项目项目",
    "Method": "目视目视目视目视目视",
    "Standard": "国标 BBB",
    "ResultCode": 1,
    "Result": "正常",
    "DocumentId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd"
  }
]
```

4、删除设备 (DELETE 方法)

<http://localhost:1080/api/Task/SheetEquipments/shnhzhch/WD000001?EquipmentNo=601>

--	--

§ 5.5.4 我的工单-工单附件

名称	工单相关附件的明细操作
描述	查询、新增、修改、删除指定工单的附件明细
方法	GET 查询 PUT 新建 POST 修改 DELETE 删除
示例	
请求	
URL	<code>/Task/SheetAttachments/{Schema}?id=xxxx</code>
参数	Schema → 指定系统租户 ID Id → 工单编号 <i>AttachmentNo</i> → 附件号，删除时需要提供本参数
响应	
成功	<code>HTTP 200/OK</code> 如果查询、新增或修改成功，则返回当前工单内的所有附件信息 <code>HTTP 200/OK</code> 如果删除成功，则返回成功删除的记录数 0
失败	<code>HTTP 403</code> 无访问权限
完整示例	
	<p>1、查询附件明细表（GET 方法）</p> <p><code>http://localhost:1080/api/Task/SheetAttachments/shnhzhch/WD000002</code> <code>http://localhost:1080/api/Task/SheetAttachments/shnhzhch?id=WD000002</code></p> <p>返回附件列表</p> <pre>[{ "AttachmentNo": 1, "WorkorderNo": "WD000002", "StepNo": 1, "SubmitTime": "2015-08-13T00:47:33", "UserId": "aa54925e-9c4b-4ed8-b582-983d5989b6a9", "DocumentId": "00000000-0000-0000-0000-000000000000" }]</pre> <p>2、新增附件记录（PUT 方法）</p> <p><code>http://localhost:1080/api/Task/SheetAttachments/shnhzhch/WD000002</code></p> <p>PUT 参数(新增一个附件):</p> <pre>[</pre>

```
{
  "AttachmentNo": 0,
  "WorkorderNo": "WD000002",
  "StepNo": 2,
  "SubmitTime": "2015-08-13T00:47:33",
  "UserId": "aa54925e-9c4b-4ed8-b582-983d5989b6a9",
  "DocumentId": "47D8A630-B1DD-481F-9CFC-9EB9F8EC0A0E"
}
```

→ 新增时置零，系统自动用最新号+1

新增后返回所有现存附件列表

```
[
  {
    "AttachmentNo": 1,
    "WorkorderNo": "WD000002",
    "StepNo": 1,
    "SubmitTime": "2015-08-13T00:47:33",
    "UserId": "aa54925e-9c4b-4ed8-b582-983d5989b6a9",
    "DocumentId": "00000000-0000-0000-0000-000000000000"
  },
  {
    "AttachmentNo": 2,
    "WorkorderNo": "WD000002",
    "StepNo": 2,
    "SubmitTime": "2015-08-13T00:47:33",
    "UserId": "aa54925e-9c4b-4ed8-b582-983d5989b6a9",
    "DocumentId": "47d8a630-b1dd-481f-9cfc-9eb9f8ec0a0e"
  }
]
```

如果与现存记录冲突，则返回错误信息：

```
{
  "ErrorCode": 409,
  "MessageDetail": "新增记录时发生错误，可能是有重复的附件"
}
```

3、修改附件（POST 方法），仅用于修改

<http://localhost:1080/api/Task/SheetAttachments/shnhzhch/WD000002>

POST 参数 (修改一行附件信息)：

```
[
  {
    "AttachmentNo": 1,
    "WorkorderNo": "WD000002",
    "StepNo": 1,
    "SubmitTime": "2015-08-13T00:47:33",
    "UserId": "aa54925e-9c4b-4ed8-b582-983d5989b6a9",
    "DocumentId": "864D2DBD-D893-4F24-8C79-03970BF821D7"
  }
]
```

→ 这是修改时所依据的主键，是记录的唯一性标识

→ 工单号不允许修改

→ 修改的内容

	<p>修改成功后返回当前工单的最新附件明细表（略）</p> <p>4、删除附件（DELETE 方法）</p> <p>http://localhost:1080/api/Task/SheetAttachments/shnhzhch/WD000002? AttachmentNo=2</p>
--	---

§ 5.5.5 我的工单-巡检明细

§ 5.5.6 我的工单-执行步骤明细

§ 5.6 工作组

§ 5.6.1 工作组列表

名称	工作组列表
描述	返回所有工作组列表
方法	GET
请求	
URL	/Task/WorkGroups/{Schema}
参数	Schema ➔ 指定系统租户 ID
响应	
成功	<p>请求 http://localhost:1080/api/Task/workgroups/shnhzhch</p> <p>返回 HTTP 200/OK 所有工作组信息列表</p> <pre>[{ "WorkgroupId": 5, "WorkgroupName": "工程管理类", "WorkgroupDescription": "", "WorkgroupType": 9, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd" }, { "WorkgroupId": 6, "WorkgroupName": "电工一班", "WorkgroupDescription": "", "WorkgroupType": 1, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd" }, { "WorkgroupId": 7,</pre>

	<pre> "WorkgroupName": "电工二班", "WorkgroupDescription": "", "WorkgroupType": 1, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd" }] </pre>
失败	HTTP 403 无访问权限

§ 5.6.2 查询特定工作组

名称	工作组查询
描述	查询并返回指定 Id 的工作组信息
方法	GET
请求	
URL	/Task/WorkGroups/{Schema}/{id}
参数	Schema ➔ 指定系统租户 ID Id ➔ 工作组编号
响应	
成功	请求 http://localhost:1080/api/Task/workgroups/shnhzhch/5 返回 HTTP 200/OK 返回所有工作组信息列表 <pre> { "WorkgroupId": 5, "WorkgroupName": "工程管理类", "WorkgroupDescription": "", "WorkgroupType": 9, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd" } </pre>
失败	HTTP 403 无访问权限

§ 5.6.3 新建工作组

名称	新建一个工作组
描述	建立一个新工作组并存入数据库
方法	PUT
请求	
URL	/Task/WorkGroups/{Schema}

参数	Schema ➔ 指定系统租户 ID
响应	
成功	<p>请求 <code>http://localhost:1080/api/Task/workgroups/shnhzhch</code></p> <pre>{ "WorkgroupId": 0, "WorkgroupName": "工程管理类", "WorkgroupDescription": "", "WorkgroupType": 9, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd" }</pre> <p>➔ 工作组 ID 保持 0，或提供一个唯一值</p> <p>➔ 工作组类型参见后面介绍</p> <p>返回 HTTP 200/OK 返回成功新建的工作组信息（略）</p>
失败	<p>HTTP 403</p> <p>无访问权限</p>
	<pre>/// <summary> /// 工作组类型 /// </summary> [Flags] public enum WorkgroupTypeConst { /// <summary> /// 未定义 /// </summary> Undefined = 0x00, /// <summary> /// 工程类 /// </summary> Engineering = 0x01, /// <summary> /// 服务类 /// </summary> Services = 0x02, /// <summary> /// 管理类 /// </summary> Management = 0x08, /// <summary> /// 服务管理类 /// </summary> ManagementOfServices = Management Services, /// <summary></pre>

	<pre>/// 工程管理类 /// </summary> ManagementOfEngineering = Management Engineering, /// <summary> /// 综合管理类 /// </summary> ManagementOfComplex = Management Engineering Services, }</pre>
--	--

§ 5.6.4 更新工作组

名称	工作组更新
描述	更新某个特定的工作组信息
方法	POST
请求	
URL	/Task/WorkGroups/{Schema}/{id}
参数	Schema ➔ 指定系统租户 ID Id ➔ 工作组编号 POST 的数据部分格式如下： <pre>{ "WorkgroupId": 5, "WorkgroupName": "工程管理类", "WorkgroupDescription": "", "WorkgroupType": 9, "ProjectId": "264f0e7b-bac8-4ff1-ad83-68cae952b3bd" }</pre>
响应	
成功	请求 http://localhost:1080/api/Task/workgroups/shnhzhch/5 (略) 返回 HTTP 200/OK 返回所有工作组信息
失败	HTTP 403 无访问权限

§ 5.6.5 删除工作组

名称	工作组更新
描述	删除某个特定的工作组及其组员信息
方法	DELETE
请求	
URL	/Task/WorkGroups/{Schema}/{id}
参数	Schema ➔ 指定系统租户 ID

	Id ➔ 工作组编号
响应	
成功	请求 http://localhost:1080/api/Task/workgroups/shnhzhch/5 (略) 返回 HTTP 200/OK 返回删除掉的工作组信息
失败	HTTP 403 无访问权限

§ 5.6.6 查询工作组成员

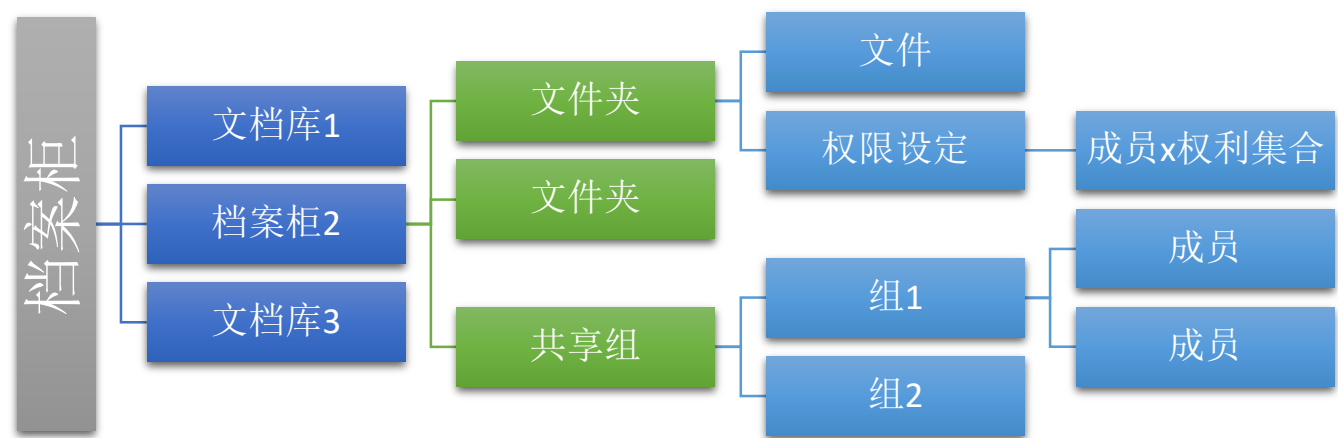
§ 5.6.7 更新工作组成员

§ 5.6.8 新建工作组成员

§ 5.6.9 删除工作组成员

六、 文档管理 API

文档库采用 Mongoddb 保存。因此文档库隐含着某些与 mongoddb 相关的概念或模式。



目录是一个虚拟的概念，目录系统模拟了文件系统的树形结构，目录项保存在 **mongoddb** 的集合中。

文件有时候又称文档，它们存储在 **Mongoddb** 的 **GridFS** 中，每个文件拥有一个唯一的 ID。每个文件除了在 **GridFS** 拥有一个信息项之外，我们的文档管理库另外再建一个集合用于存储关于文档的更多元信息。例如档案号、文件编号、单据号码之类。

此外，当某个文件拥有多个版本时，在文档入口项中登记着前几个版本的文档 ID。

为了区分两种文档（常规/**GridFS**）信息入口，后面 API 中把 **GridFS** 下的文档 Entry 称为“**GridFSFileInfo**”，而本系统使用自定义的入口简称为“**FileInfo**”。即只要名称中有“**GridFS**”，便意味着调用 **Mongoddb** 底层的操作。

权限仅针对目录进行控制。采用 资源 + 用户/工作组 + 读写权 的组合来管理。

每个目录项拥有下述属性：

FolderId	文件及 Id，GUID 类型
FolderName	文件夹名称
CreateTime	创建时刻
ParentFolderId	上级文件夹 ID, Guid.Empty 表示顶级目录
Purview	权限列表，指示了多个成员（或组）在本文件夹上的权限

每个文件拥有下述几个重要的元信息

DocumentId	=	文档 Id，每个文档拥有一个独立唯一的 ID
ParentFolderId	=	文档所属的文件夹 ID。可通过 folder/list 得到文件夹列表
Filename	=	文件名
ReferenceNo	=	档案编号/文档编号（
UploadTime	=	上传时刻-自动修改为当前时刻

Length = 文件大小，文件在服务器上占据的空间大小
ContentType = 文件类型，一般维持原样。例如 image/jpeg、image/png

共享组 MemberGroup

MemberGroupId 共享组 Id
MemberGroupName 名称
List<MemberInfo> 组内成员

用户/成员 Member

MemberId 用户 ID，这个 Id 来自无限能系统内的联系人
MemberName 用户名

权限列表(Purview)

List<FolderPurviewEntity> 权限列表，指示了多个成员在本文件夹上的权限

权限基本项(FolderPurviewEntity)

MemberOrGroupList 成员 ID、或组 ID 列表
Permission 操作权限组合（读、写、删）

权限枚举

```
public enum PermissionConst
{
    Undefined = 0x0000,

    FileUpload = 0x0001,
    FileDownload = 0x0002,
    FileDelete = 0x0004,
    FileEdit = 0x0008,
    FileRename = 0x0010,

    FolderCreat = 0x0100,
    FolderDelete = 0x0200,
    FolderRename = 0x0400,

    AllFileRights = FileUpload | FileDownload | FileDelete | FileEdit | FileRename,
    AllFolderRights = FolderCreat | FolderDelete | FolderRename,
    AllRights = AllFileRights | AllFolderRights,
}
```

§ 6.1 文档库列表

每个系统租户拥有一个主文档库，这个库一般与系统租户 ID 相同。
此外，还可以拥有多个私有文档库，每套文档库均独立存储。

名称	查询文档库列表
描述	根据当前登录用户信息，获取并返回其有权操作的文档库列表
方法	GET

请求	
URL	Common/ArchiveRepo?accesskey=xxxx
参数	
示例	<code>http://localhost:1080/api/Common/ArchiveRepo ? accesskey=web_7cd570a0fbae422a97e73be110f38c0f</code>
响应	
成功	<p>HTTP 200/OK 返回文档库列表</p> <pre>[{ "ArchiveRepoId": "24f10f54-3a38-4842-bcba-9d22ab7b657d", "ArchiveRepoTitle": "ABC 公司主档案库", "SysTenantryId": "SHTPV", "ArchiveRepoName": "SHTPV", ➔ 文档库名称，此后访问时的必带参数 "AppKey": null, "MasterKey": null, "MasterRepository": true, "TimeStamp": "2015-06-26T16:38:16Z", "Comment": null }, { "ArchiveRepoId": "863461f4-6c6e-4471-a9a5-6113c0759d81", "ArchiveRepoTitle": "ABC 公司加密档案库", "SysTenantryId": "SHTPV", "ArchiveRepoName": "SHTPV_S1", ➔ 文档库名称，此后访问时的必带参数 "AppKey": null, "MasterKey": null, "MasterRepository": false, "TimeStamp": "2015-06-26T16:38:16Z", "Comment": null }]</pre>
失败	<p>HTTP 403</p> <p>无访问权限</p>

§ 6.2 MongoDB 数据库-列表

每个文档库（又称档案柜）存储在一套独立的 MongoDB 中，拥有一个唯一的名称。
后续的操作均以某个特定的 MongoDB 为操作对象

名称	查询 MongoDB 数据库列表
描述	根据当前登录用户信息，获取并返回其有权操作的 MongoDB 库列表
方法	GET

请求	
URL	/doc/databases?accesskey=xxxx
参数	
示例	http://localhost:1080/api/doc/Databases?accesskey=web_7cd570a0fbae422a97e73be110f38c0f
响应	
成功	<p>HTTP 200/OK 返回文档库列表</p> <pre>["SHNHZHCH", "SHTPV", "TPV"]</pre>
失败	<p>HTTP 403</p> <p>无访问权限</p>

§ 6.3 目录-列目录

名称	列目录
描述	列出指定文档数据 schema 中的特定目录{ id }下的子目录列表
方法	GET
请求	
URL	<p>或 /Doc/Folders/{schema} → 列出顶级目录列表</p> <p>或 /Doc/Folders/{schema}/{id} → 列出给定父目录 Id 的下级目录列表</p> <p>或 /Doc/Folders/{schema}?{id}=xxxx → 同上</p>
参数	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部
	TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	<p>列出所有顶级目录</p> <p>http://localhost:1080/api/doc/folders/SHNHZHCH ?accesskey=xxxxxxxxxx</p> <p>列出“工单”下的子目录</p> <p>http://localhost:1080/api/doc/folders/SHNHZHCH?id=955e533b-fd05-4a0c-901d-a281c81464f1&accesskey=xxxxxxxxxx</p>
响应	
成功	http://localhost:1080/api/doc/folders/shtpv?accesskey=web_6142db855142437d86d144f48c97716f

	<div>成功则返回目录列表</div> <div>[{"FolderId": "9850c5f8-fd03-405c-b541-88ba11efa62", "FolderName": "图纸", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}, {"FolderId": "3b72be51-fd02-4edf-81b8-c56d9c205ec5", "FolderName": "预算", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}, {"FolderId": "6fc81e5c-fd06-4e8c-95f7-c8e308b61c97", "FolderName": "联系人", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}, {"FolderId": "1b2e3f16-fd01-4c18-8f72-201cced67771", "FolderName": "合同", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}, {"FolderId": "5b983a46-fd04-4d66-9d06-4725b462cde8", "FolderName": "规范", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}, {"FolderId": "955e533b-fd05-4a0c-901d-a281c81464f1", "FolderName": "工单及附件", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}, {"FolderId": "458db469-fd07-4dd9-b3f2-aa43cb219f4a", "FolderName": "其它", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}, {"FolderId": "6EC10E7A-FD08-48B2-97DD-05E280A0915D", "FolderName": "回收站", "CreateTime": "2015-07-24T18:14:43.733+08:00", "ParentFolderId": "00000000-0000-0000-0000-000000000000", "Purview": []}]</div>	左边的目录是系统内建的默认目录。其ID值全局唯一。
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"	

§ 6.4 目录-创建

名称	创建目录
描述	在指定目录下创建子目录 如果 ParentFolderId 为空白，则创建顶级目录
方法	PUT
请求	
URL	/Doc/ Folders/{schema} 创建新目录
参数	{ "ParentFolderId": "955e533b-fd05-4a0c-901d-a281c81464f1", ➔ 父目录 ID。这里表示在“工单”下创建子目录 "FolderName": "这是工单下的子文件夹", }
	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	http://localhost:1080/api/doc/folders/SHTPV?accesskey=web_6142db855142437d86d144f48c97716f
响应	
成功	<p>成功则返回创建完成的目录信息</p> <pre>{ "FolderId": "a4955237-d9eb-4916-8a95-facb12cf1d3f", "FolderName": "这是工单下的子文件夹", "CreateTime": "2015-08-14T23:10:18.8089396+08:00", "ParentFolderId": "955e533b-fd05-4a0c-901d-a281c81464f1", "Purview": [] }</pre> <p>查询工单目录下的子目录</p> <p>http://localhost:1080/api/doc/folders/shtpv/955e533b-fd05-4a0c-901d-a281c81464f1?accesskey=web_6142db855142437d86d144f48c97716f</p> <pre>[{ "FolderId": "3fc73b5d-7efa-4cf4-92cd-06549305a950", "FolderName": "这是子文件夹名称", "CreateTime": "2015-08-09T21:44:27.929+08:00", "ParentFolderId": "955e533b-fd05-4a0c-901d-a281c81464f1", "Purview": [] }, { "FolderId": "3bb9eaf9-cble-4df7-86fe-839eed87b41d", "FolderName": "新的子文件夹名称", "CreateTime": "2015-08-11T02:17:27.715+08:00", "ParentFolderId": "955e533b-fd05-4a0c-901d-a281c81464f1", "Purview": [] }]</pre>
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"

--	--

§ 6.5 目录-修改(改名、移动等)

名称	修改目录项信息
描述	允许修改文件名、父文件夹、权限设置等 只要修改特定目录实体的 ParentFolderId 即可实现移动的目的。但是务必保证新的父目录 Id 存在
方法	POST
请求	
URL	/Doc/Folders/{schema}
参数	<pre>{ "folderId": "3fc73b5d-7efa-4cf4-92cd-06549305a950", "folderName": "新文件夹名", "createTime": "2015-08-20T21:44:27.9291028+08:00", "parentFolderId": "458DB469-FD07-4DD9-B3F2-AA43CB219F4A", "purview": [] }</pre> <p> → 这个 Id 永远不允许改动 → 可以修改文件夹名 → 时刻也变成当前时刻 → 也可顺便移动到“其它”目录下 </p> <p> AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile), 默认 web 类型 </p>
调用示例	http://api.losta.net/api/doc/folder/create/shnhzhch?accesskey=xxxxxxxxxx
响应	
成功	<p>成功则返回创建完成的目录信息</p> <pre>{ "folderId": "3fc73b5d-7efa-4cf4-92cd-06549305a950", "folderName": "新文件夹名", "createTime": "2015-08-20T21:44:27.9291028+08:00", "parentFolderId": "458DB469-FD07-4DD9-B3F2-AA43CB219F4A", "purview": [] }</pre>
失败	<p>HTTP 403</p> <p>“用户未登陆或认证失败，禁止访问”</p>

§ 6.6 目录-删除

名称	删除目录项
描述	删除目前项之前，应确保该目录下没有文件、没有子目录
方法	DELETE
请求	
URL	或 /Doc/Folders/{schema}/{id} 或 /Doc/Folder/{schema}?id=文件夹 ID
参数	Id = 文件夹 ID

	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	<code>http://api.losta.net/api/doc/folders/shnhzhch/3fc73b5d-7efa-4cf4-92cd-06549305a950?accesskey=xxxxxxxxxx</code>
响应	
成功	成功则返回已经删除掉的目录的信息 <pre>{ "folderId": "3fc73b5d-7efa-4cf4-92cd-06549305a950", "folderName": "新文件夹名", "createTime": "2015-08-20T21:44:27.9291028+08:00", "parentFolderId": "458DB469-FD07-4DD9-B3F2-AA43CB219F4A", "purview": [] }</pre>
失败	HTTP 403 “用户未登陆或认证失败，禁止访问”

§ 6.7 文件-列表

名称	文件列表
描述	返回指定文件夹下的所有文件列表
方法	GET
请求	
URL	<code>/Doc/Files/{schema}?mode={mode}&folderId={folderId}&pageNo={no}&pageSize={size}</code>
参数	Mode → 列表方式，“Normal” 常规列表、“GridFS” 按 MongoDB 内部格式列表 FolderId → (选项) 文件夹 ID，默认为“00000000-0000-0000-0000-000000000000”，表示列出根目录下文件 PageNo、PageSize → (选项) 页号、页大小。当记录太多需要分页时提供。默认第 1 页，每页 100 条记录
	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	<code>http://localhost:1080/api/doc/files/SHTPV?mode=normal&accesskey=web_6142db855142437d86d144f48c97716f</code> <code>http://localhost:1080/api/doc/files/SHTPV?mode=normal&folderId=3fc73b5d-7efa-4cf4-92cd-06549305a950&accesskey=web_6142db855142437d86d144f48c97716f</code> <code>http://localhost:1080/api/doc/files/SHTPV?mode=gridfs&accesskey=web_6142db855142437d86d144f48c97716f</code>
响应	
成功	成功指定目录下文件列表 <pre>[{ "DocumentId": "b6c5f1ef-508b-4b87-b5ea-5b964bfd334d", "Filename": "b6c5f1ef-508b-4b87-b5ea-5b964bfd334d.jpg", "ReferenceNo": "", "UploadTime": "2015-07-27T11:13:57.11+08:00", "Length": 90958, }]</pre>

	<pre>"ContentType": null, "ParentFolderId": "ff0ff050-26c6-412b-a03c-ad0f849966c0", "VersionHistory": [{ "Version": 1, "Time": "2015-07-27T03:13:57.647Z", "GridFileId": "b6c5f1ef-508b-4b87-b5ea-5b964bfd334d" }], }, { "DocumentId": "e34f9207-5031-4b5f-8490-97091609b22d", "Filename": "e34f9207-5031-4b5f-8490-97091609b22d.jpg", "ReferenceNo": "", "UploadTime": "2015-07-27T11:14:01.152+08:00", "Length": 135774, "ContentType": null, "ParentFolderId": "ff0ff050-26c6-412b-a03c-ad0f849966c0", "VersionHistory": [{ "Version": 1, "Time": "2015-07-27T03:14:01.451Z", "GridFileId": "e34f9207-5031-4b5f-8490-97091609b22d" }] }]</pre>
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"

§ 6.8 文件-修改(改名、移动等)

名称	文件修改
描述	改名、移动到其它文件夹、更改文档号等操作均在本 API 下完成 务必首先查询获取文件的全部信息，然后变更其中需要变更的部分，再调用本更新操作
方法	POST
请求	
URL	或 /Doc/Files/{schema}
参数	DocumentId ➔ 文件 ID, 必选参数 Filename ➔ 文件名称 ReferenceNo ➔ 文档号 UploadTime ➔ 上传时刻 Length ➔ 长度（应保持与查询到的一致） ParentFolderId ➔ 父文件 ID VersionHistory ➔ 版本历史（应保持与查询到的一致）

	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型 (web、mobile), 默认 web 类型
调用示例	POST 调用 http://localhost:1080/api/doc/files/shtpv/895b5e2a-53f4-4446-9f18-a5896060ff6b 内容 : <pre> { "DocumentId": "895b5e2a-53f4-4446-9f18-a5896060ff6b", ➔ 与 Url 中的 Id 必须一致 "Filename": "改个新名字 C.txt", "ReferenceNo": "加个文档号", "UploadTime": "2015-08-14T00:00:00+08:00", "Length": 1998, "ContentType": null, "ParentFolderId": "00000000-0000-0000-0000-000000000000", "VersionHistory": [{ "Version": 1, "Time": "2015-07-10T12:40:48.08Z", "GridFileId": "2c3ddc25-5357-4dfd-acf5-c815abf19a1b" }] }</pre>
响应	
成功	成功则返回更新后的文件信息 (略)
失败	HTTP 401 “用户未登陆或认证失败，禁止访问”

§ 6.9 文件-删除

名称	文件删除
描述	返回指定 Id 的文件或文件列表
方法	DELETE
请求	
URL	Doc/Files/{schema}/{id} ➔ 删除一个文件 Doc/Files/{schema} ➔ 删除多个文件，文件 Id 列表在 BODY 中
参数	Id ➔ 文件 ID，待删除的文档 ID AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型 (web、mobile), 默认 web 类型
调用示例	删除一个文件 DELETE 调用 http://localhost:1080/api/doc/files?id=a0e667fb-7f21-4a03-8759-138e3deede67 删除多个文件 DELETE 调用 http://localhost:1080/api/doc/files <pre>[{"DocumentId": "a0e667fb-7f21-4a03-8759-138e3deede67"}, </pre>

	<pre>{ "DocumentId": "bc2862e9-324d-4e0b-8863-7a745304fc79", "DocumentId": "d99b4aa1-127e-4bbe-af13-a577204f64d0", "DocumentId": "be345a2d-b1e5-4596-8bce-8e6955f7b532" }</pre>
响应	
成功	<p>返回成功删除掉的文件信息列表</p> <pre>[{ "Key": "Fault", "Value": { "DocumentId": "a0e667fb-7f21-4a03-8759-138e3deede67", "Filename": "文件不存在!", "ReferenceNo": "", "UploadTime": "2015-08-15T00:00:00+08:00", "Length": 0, "ContentType": null, "ParentFolderId": "00000000-0000-0000-0000-000000000000", "VersionHistory": [] } }, { "Key": "Succeeded", "Value": { "DocumentId": "895b5e2a-53f4-4446-9f18-a5896060ff6b", "Filename": "ABC.txt", "ReferenceNo": "", "UploadTime": "2015-08-14T00:00:00+08:00", "Length": 1998, "ContentType": null, "ParentFolderId": "00000000-0000-0000-0000-000000000000", "VersionHistory": [{ "Version": 1, "Time": "2015-07-10T12:40:48.08Z", "GridFileId": "2c3ddc25-5357-4dfd-acf5-c815abf19a1b" }] } }, { "Key": "Fault", "Value": { "DocumentId": "bc2862e9-324d-4e0b-8863-7a745304fc79", "Filename": "文件不存在!", "ReferenceNo": "", "UploadTime": "2015-08-15T00:00:00+08:00", "Length": 0, "ContentType": null, </pre>

	<pre> "ParentFolderId": "00000000-0000-0000-0000-000000000000", "VersionHistory": [] } }, { "Key": "Fault", "Value": { "DocumentId": "d99b4aa1-127e-4bbe-af13-a577204f64d0", "Filename": "文件不存在!", "ReferenceNo": "", "UploadTime": "2015-08-15T00:00:00+08:00", "Length": 0, "ContentType": null, "ParentFolderId": "00000000-0000-0000-0000-000000000000", "VersionHistory": [] } }, { "Key": "Fault", "Value": { "DocumentId": "be345a2d-b1e5-4596-8bce-8e6955f7b532", "Filename": "文件不存在!", "ReferenceNo": "", "UploadTime": "2015-08-15T00:00:00+08:00", "Length": 0, "ContentType": null, "ParentFolderId": "00000000-0000-0000-0000-000000000000", "VersionHistory": [] } }]</pre>
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"

§ 6.10 文件-预览

名称	文件下载或预览
描述	返回指定 Id 的文件，可在浏览器中观看或另存为本地文件
方法	GET
请求	
URL	或 /Doc/Files/{schema}/{id}
参数	Id = 文档 ID，待预览的文档 ID AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile),默认 web 类型
调用示例	http://localhost:1080/api/doc/files/shtpv/21778b87-9ede-47ef-83fe-2a23984f586d

响应	
成功	成功则返回文件供预览
失败	<pre>{ "ErrorCode": 403, "MessageDetail": "用户未登陆或认证失败，禁止访问" }</pre> <pre>{ "ErrorCode": 404, "MessageDetail": "文件不存在" }</pre>

§ 6.11 文件上传

名称	文件上传
描述	上传一个全新文档到指定目录中。 必须使用 multipart/form-data 格式上传文件。否则将出错
方法	PUT
请求	
URL	或 /Doc/Files/{schema}
参数	DocumentId = 文件 ID, 留空 Filename = 文件名称 ReferenceNo = 文档号 (可留空) UploadTime = 上传时刻 (系统自动改为当前时刻) ParentFolderId = 父文件 ID AccessKey = 这个值应放置在 HTTP HEADER 中, 开发期间可附加在 URL 尾部 TerminalType = 终端类型 (web、mobile), 默认 web 类型
调用示例	http://localhost:1080/api/doc/files/shtpv <pre>{ "Filename": "bc2862e9-324d-4e0b-8863-7a745304fc79.jpg", "ReferenceNo": "", "ParentFolderId": "ff0ff050-26c6-412b-a03c-ad0f849966c0" }</pre>
响应	
成功	成功则返回上传文件的完整信息 <pre>{ "DocumentId": "bc2862e9-324d-4e0b-8863-7a745304fc79", "Filename": "bc2862e9-324d-4e0b-8863-7a745304fc79.jpg", "ReferenceNo": "", "UploadTime": "2015-07-27T11:32:51.219+08:00", </pre>

	<pre>"Length": 176836, "ContentType": null, "ParentFolderId": "ff0ff050-26c6-412b-a03c-ad0f849966c0", "VersionHistory": [{ "Version": 1, "Time": "2015-07-27T03:32:51.513Z", "GridFileId": "bc2862e9-324d-4e0b-8863-7a745304fc79" }]</pre>
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"

§ 6.12 共享组-列表

名称	共享组查询
描述	列出所有共享组的全部信息
方法	GET
请求	
URL	或 /Doc/ Groups/{schema}
参数	
	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	http://api.losta.net/api/doc/groups/shnhzhch?accesskey=xxxxxxxxxx
响应	
成功	成功则返回档案库内所有已有的共享组信息 <pre>[{ "GroupId": "c034d9e2-cb60-4ae5-ae07-71972748ac0a", "GroupName": "共享一组", "Members": [] }]</pre>
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"

§ 6.13 共享组-查询详情

名称	查询某个特定 ID 的共享组信息
描述	
方法	GET
请求	
URL	或 /Doc/ Groups/{schema}/{id}
参数	{id} → 共享组 ID
	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部
	TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	http://api.losta.net/api/doc/groups/shnhzhch/ c034d9e2-cb60-4ae5-ae07-71972748ac0a?accesskey=xxxxxxxxxx
响应	
成功	成功则返回档案库内所有已有的共享组信息 <pre>[{ "GroupId": "c034d9e2-cb60-4ae5-ae07-71972748ac0a", "GroupName": "共享一组", "Members": [] }]</pre>
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"

§ 6.14 共享组-新建

名称	新建共享组
描述	新建一个共享组
方法	PUT
请求	
URL	或 /Doc/ Groups/{schema}
参数	GroupId: 共享组 ID，留空系统自动分配一个
	GroupName: 共享组名称
	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	http://api.losta.net/api/doc/groups/shnhzhch?accesskey=xxxxxxxxxx <pre>{ "GroupName": "共享三组" }</pre>

响应	
成功	成功则返回档案库内所有已有的共享组信息 [{ "GroupId": "c034d9e2-cb60-4ae5-ae07-71972748ac0a", "GroupName": "共享三组", "Members": [] }]
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"

§ 6.15 共享组-修改组信息

名称	修改共享组
描述	可以一次性的修改共享组名称、增删组员等 务必首先查询共享组信息，然后再返回信息的基础上变更后修改
方法	POST
请求	
URL	或 /Doc/ Groups/{schema}
JSON 参数	{ "GroupId": "c034d9e2-cb60-4ae5-ae07-71972748ac0a", "GroupName": "共享三组", "Members": [] }
	AccessKey = 这个值应放置在 HTTP HEADER 中，开发期间可附加在 URL 尾部 TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	http://api.losta.net/api/doc/groups/shnhzhch
响应	
成功	成功则返回已经修改后的组信息 { "GroupId": "c034d9e2-cb60-4ae5-ae07-71972748ac0a", "GroupName": "共享三组", "Members": [] }
失败	HTTP 403 "用户未登陆或认证失败，禁止访问"
说明	这里的共享组成员信息可从联系人 API 中查询得到。 文档系统不存储联系人的信息，也不主动与联系人信息同步。

§ 6.16 共享组-删除组

名称	删除共享组
描述	删除某个指定的一个共享组。 组信息、组内成员将被删除。 如果该组被某些文件夹引用，则暂不删除那些引用
方法	DELETE
请求	
URL	DELETE 方法的 /Doc/ Groups/{schema}/{id}
参数	Id: 共享组 ID, 这个组及组内成员信息将被删除
	AccessKey = 这个值应放置在 HTTP HEADER 中, 开发期间可附加在 URL 尾部
	TerminalType = 终端类型(web、mobile), 默认 web 类型
调用示例	http://api.losta.net/api/doc/groups /shnhzhch?id= c034d9e2-cb60-4ae5-ae07-71972748ac0a
响应	
成功	成功则返回已经删除的组信息 <pre>[{ "GroupId": "c034d9e2-cb60-4ae5-ae07-71972748ac0a", "GroupName": "共享三组", "Members": [] }]</pre>
失败	HTTP 403 "用户未登陆或认证失败, 禁止访问"