

Table of Contents

总述	1.1
使用说明	1.2
图片	1.3
联系人	1.4
文件	1.5
认证	1.6
网页	1.7
地理位置	1.8
设备信息	1.9
轻应用	1.10
扫描二维码	1.11
应用市场	1.12
公有云	1.13
换肤主题	1.14
支付	1.15
网络连接信息	1.16
Umeeting 会议	1.17

WorkPlus Cordova API说明文档

WorkPlus Cordova SDK驻留在WorkPlus移动App上，为部署在WorkPlus应用中心的轻应用提供访问WorkPlus移动App功能的能力。

当前版本的Cordova SDK依赖的Cordova版本是3.8.0。本文只介绍WorkPlus Cordova API，如果您需要详细了解Cordova，请参考其官方网站<http://cordova.apache.org/>。

注入使用说明

cordova 所依赖的 js 并不需要调用方直接引入, 在原生代码里(android/ ios)已经含有对应的 js sdk 代码, 因此调用方只需要添加以下规则即可调用cordova:

```
1. <script src="applocal://www/cordova.min.js"></script>  
  
    // "xxx/xxx/cordova.min.js"为内容是空的 cordova.min.js 文件路径, 也即存在该文件,但不需要实际内容  
2. <script src="https://xxx/xxx/cordova.min.js"></script>
```

其中需要注意的是, 语句2 是为了兼容在 android 6.0 以下, https 环境里自定义协议头, 如 语句1 中的 applocal:// 会出现报错, 无法正常使用.

因此以上2个语句都是必须的, 否则在部分 android 机器会出现 cordova 无法调用的问题.

注入成功回调

当接收到以下回调时, 才说明 cordova 注入成功, 并且能开始调用:

```
document.addEventListener("deviceready", function () {  
    //可以开始调用 cordova 了  
  
}, false);
```

图片插件(WorkPlus_Image)

1. 拍照返回

拍照，压缩图片后直接返回图片

请求调用示例:

```
function takePhoto(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Image",
    "takePhoto",
    []);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "imageURL": "压缩后图像存在本地的地址",
    "key": "原图像存在本地的地址",
    "imageInfo": //相对于图像的信息
    {
        "height": "图像高",
        "width": "图像宽",
        "size": "图像大小"
    }
}
```

2. 拍照返回并且可编辑

拍照，截图返回

请求调用示例:

```
function takePhotoWithEdit(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Image",
    "takePhotoWithEdit",
    []);
}
```

```

    "WorkPlus_Image",
    "takePhotoWithEdit",
    []
);
}

说明:
请求输入数据: 无
请求返回数据:
{
    "imageURL": "压缩后图像存在本地的地址",
    "key": "原图像存在本地的地址",
    "imageInfo": //相对于图像的信息
    {
        "height": "图像高",
        "width": "图像宽",
        "size": "图像大小"
    }
}

```

3. 选择图片 (单张)

调用图片相册，选择图片并压缩返回

请求调用示例:

```

function selectImage(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Image",
    "selectImage",
    []
);
}

```

说明:

请求输入数据: 无

请求返回数据:

```

{
    "imageURL": "压缩后图像存在本地的地址",
    "key": "原图像存在本地的地址",
    "imageInfo": //相对于图像的信息
    {
        "height": "图像高",
        "width": "图像宽",
        "size": "图像大小"
    }
}

```

4. 选择图片并截图返回（单张）

调用图片相册，选择图片并截取返回

请求调用示例:

```
function selectImageWithEdit(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Image",
    "selectImageWithEdit",
    []);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "imageURL": "压缩后图像存在本地的地址",
    "key": "原图像存在本地的地址",
    "imageInfo": //相对于图像的信息
    {
        "height": "图像高",
        "width": "图像宽",
        "size": "图像大小"
    }
}
```

5. 选择多张图片

调用图片相册，选择多张图片并压缩返回，并且支持选过图片的传输

请求调用示例:

```
function selectImageWithEdit(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Image",
    "selectImages",
    [{"imageKeys": ["图片在本机的地址", "xxx", ...]}]
});
```

说明:

请求输入数据(可无): [{"imageKeys": ["图片在本机的地址", "xxx", ...]}]

请求返回数据:

```
[{
    "imageURL": "压缩后图像存在本地的地址",
    "key": "原图像存在本地的地址",
    "imageInfo": //相对于图像的信息
    {
        "height": "图像高",
        "width": "图像宽",
        "size": "图像大小"
    }
},
...
]
```

6. 清除压缩后的图片

拍照或选择照片后都会生成压缩图片，调用这个方法去清除

请求调用示例:

```
function cleanCompressImage(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Image",
    "cleanCompressImage",
    []
);
}
```

说明:

请求输入数据: 无

请求返回数据: 无

7. 批量预览图片(新增 position 请求参数，在workplus3.1.3版本后使用)

传输图片地址，预览图片

请求调用示例:

```
function showImages(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
}
```

```
function(error) {
    alert("调用失败");
},
"WorkPlus_Image",
"showImages",
[{
    "urls": ["图片在本机的地址", "xxxxx", ...],
    "medias": ["图片的mediaId", "xxxxx", ...],
    "position": "1"
}]
);
}
```

说明:

请求输入数据:

```
[{
    "urls": ["图片在本机的地址", "xxxxx", ...],
    "medias": ["图片的mediaId", "xxxxx", ...],
    "position": "1" (表示从第几张打开)
}]
```

请求返回数据: 无

8. 保存图片

传输图片地址，保存图片

请求调用示例:

```
function saveImages(){
cordova.exec(function(result) {
    alert(JSON.stringify(result, null, 4));
},
function(error) {
    alert("调用失败");
},
"WorkPlus_Image",
"saveImages",
[
    {
        "mediaId": "图片mediaId",
        "url": "图片url地址",
    }
]
);
}
```

说明:

请求输入数据:

```
[{
    "mediaId": "图片mediaId",
```

```

        "url": "图片url地址",
    }
]
请求返回数据: 无

```

9. 长按图片弹出框

弹出框包括"识别二维码", "保存图片"等选项

请求调用示例:

```

function actionForLongPressImage(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Image",
    "actionForLongPressImage",
    []
);
}

```

说明:

请求输入数据:

```

[
    {
        "imageData": "base64数据"
    }
]

```

请求返回数据: 无

10. 新增拍照返回接口(返回带mediald方式)(Workplus 3.1.3版本以上使用)

新增接口, 拍照返回, 根据传入参数决定是否可以进行裁剪编辑, 同时会将图片上传后台, 返回mediald

请求调用示例:

```

function takePicture() {
    cordova.exec(function(result) {
        alert(result)
    },
    function(error) {
        alert(error)
    },
    "WorkPlus_Image",

```

```

        "takePicture", [{"editable":true
    }]);
}

```

说明:

请求输入数据: [{"editable":true}]

editable 表示是否进行裁剪

true : 进行裁剪返回, false : 不裁剪直接返回

请求返回数据:

```

{
    "mediaId": "返回后台的mediaId"
    "imageURL": "压缩后图像存在本地的地址",
    "key": "原图像存在本地的地址",
    "imageInfo": //相对于图像的信息
    {
        "height": "图像高",
        "width": "图像宽",
        "size": "图像大小"
    }
}

```

11. 新增选择图片接口(返回带mediald方式)(Workplus 3.1.3版本以上使用)

调用图片相册, 根据用户参数决定是否选择多张图片或单张图片, 支持编辑剪裁(编辑剪裁功能仅限于单张图片功能), 并且支持选过图片的传输, 选择完后会进行后台上传, 返回值中带有上传后的mediald

请求调用示例:

```

cordova.exec(function(result) {
    alert(result)
},
function(error) {
    alert(error)
},
"WorkPlus_Image",
"chooseImages", [{"multiple":true,
"imageKeys": ["图片在本机的地址", "xxx", ...],
"editable":true,
"file_limit": {"max_select_count": 12, "single_select_size": -1, "total_select_size": -1}],
});

```

说明:

请求输入数据:

```

[{
    "multiple":true,
    "imageKeys": ["图片在本机的地址", "xxx", ...],
    "editable":true,
    "file_limit": {

```

```

    "max_select_count": 12,
    "single_select_size": -1,
    "total_select_size": -1
  },
}

multiple : 是否是多选, ture(多选)  fals(单选)
editable : 是否对图像进行剪裁 true(剪裁)  false(不剪裁)
注意: editable 为true剪裁方式仅对于单选图片情况下生效, 即multiple = false
imageKeys : 可以传送之前选过的图片

file_limit:
max_select_count 多选时数量的限制,

single_select_size 单个文件选择大小限制(byte), -1表示不限制

total_select_size, 所有文件选择大小限制(byte), -1表示不限制

```

请求返回数据:

```

[{
  "imageURL": "压缩后图像存在本地的地址",
  "key": "原图像存在本地的地址",
  "mediaId": "图像在后台的媒体mediaId"
  "imageInfo": //相对于图像的信息
  {
    "height": "图像高",
    "width": "图像宽",
    "size": "图像大小"
  }
},
...
]

```

12. 图片添加水印接口(Workplus 3.6.0版本以上使用)

拍照, 生成水印图片返回

请求调用示例:

```

function takePictureAddWatermark() {
  cordova.exec(function (result) {
    },
    function (error) {
      alert("调用失败");
    },
    "WorkPlus_Image",
    "takePhotoAndAddWaterMark",
    [{"content": "1234567890"},
```

```
    "font_size": "14",
    "color": "#FF5858"
  ],
)
}
```

说明:

请求输入数据:

```
[{
  "content": "水印内容",
  "font_size": "水印字号大小",
  "color": "水印颜色",
}]
```

请求返回数据:

无

获取联系人信息插件(WorkPlus_Contact)

1. 单选联系人(2.0接口, 3.0做了兼容)

打开当前组织的联系人列表, 单选一个联系人(兼容2.0数据)

请求调用示例:

```
function getContact(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "getContact",
    []);
}
```

说明:

请求输入数据 (也可以不输入数据, 默认显示选择自己) :

```
[{
    "hideMe": "true", // 表示在选择联系人的时候, 是否包括显示选择自己
    //true表示不显示自己, false表示显示自己

    "filterSenior": 1
    //1 表示过滤高管, 0 表示不过滤
}]
```

请求返回数据: 数据列表 (选择的雇员)

```
[{
    "identifier": "雇员id",
    "accountName": "雇员username",
    "tenantId": "雇员domainId",
    "post": "雇员职位信息",
    "job_title": "雇员职位信息",
    "org_name": "雇员所属机构",
    "com_name": "雇员所属公司",

    //---- 以上为2.0的字段, 接口做了兼容, 3.0用户可以忽略 ----

    "user_id": "雇员id",
    "domain_id": "域id",
    "username": "雇员username",
    "name": "雇员name",
    "userId": "雇员id",
```

```

"domainId", "雇员domainId",
"avatar":"雇员头像",
"mobile":"雇员电话号码",
"email":"雇员email",
"gender":"雇员性别",
"firstNameLetter":"雇员的简拼",
"status":"雇员激活状态",
"sort":"雇员排序号",
"post":"雇员职位信息",
"jobTitle":"雇员职位信息",
"orgName":"雇员所属机构",
"corpName":"雇员所属公司",
"positions":[{职位信息详情..}]

}]

```

2. 多选联系人列表(2.0接口, 3.0做了兼容)

打开当前组织的联系人列表，选择多个联系人(兼容2.0数据)

请求调用示例:

```

function getContacts(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "getContacts",
    []);
}

```

说明:

请求输入数据(可无):

```

[{
    "selectedContacts":[
        //用户回传过来的雇员列表信息
    {
        "identifier":"雇员id",
        "accountName":"雇员username",
        "tenantId":"雇员domainId",
        "post":"雇员职位信息",
        "job_title":"雇员职位信息",
        "org_name":"雇员所属机构",
        "com_name":"雇员所属公司",
    }
}]

```

---- 以上为2.0的字段，接口做了兼容，3.0用户可以忽略 ----

```

"user_id": "雇员id",
"domain_id": "域id",
"username": "雇员username",
"name": "雇员name",
"userId": "雇员id",
"domainId", "雇员domainId",
"avatar": "雇员头像",
"mobile": "雇员电话号码",
"email": "雇员email",
"gender": "雇员性别",
"firstNameLetter": "雇员的简拼",
"status": "雇员激活状态",
"sort": "雇员排序号",
"post": "雇员职位信息",
"jobTitle": "雇员职位信息",
"orgName": "雇员所属机构",
"corpName": "雇员所属公司",
"positions": [{职位信息详情..}]}

}

...
],

"hideMe" : "true", // 表示在选择联系人的时候，是否包括显示选择自己
"filterSenior":1
//1 表示过滤高管，0 表示不过滤
}]

```

请求返回数据：用户选择的雇员的数据列表

```
[{
  "identifier": "雇员id",
  "accountName": "雇员username",
  "tenantId": "雇员domainId",
  "post": "雇员职位信息",
  "job_title": "雇员职位信息",
  "org_name": "雇员所属机构",
  "com_name": "雇员所属公司",
}
```

---- 以上为2.0的字段，接口做了兼容，3.0用户可以忽略 ----

```

"user_id": "雇员id",
"domain_id": "域id",
"username": "雇员username",
"name": "雇员name",
"userId": "雇员id",
"domainId", "雇员domainId",
"avatar": "雇员头像",
"mobile": "雇员电话号码",
"email": "雇员email",
"gender": "雇员性别",
"firstNameLetter": "雇员的简拼",
"status": "雇员激活状态",

```

```

"sort":"雇员排序号",
"post":"雇员职位信息",
"jobTitle":"雇员职位信息",
"orgName":"雇员所属机构",
"corpName":"雇员所属公司",
"positions":[{职位信息详情..}]}

},
{
...
},
...
]

```

3. 多选当前组织的雇员列表

打开当前组织的联系人列表，选择多个联系人

请求调用示例:

```

function getEmployeesFromCurrentOrg(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "getEmployeesFromCurrentOrg",
    []);
}

```

说明:

请求输入数据(可无):

```
[{
    "selectedEmpList": [
        //用户回传过来的雇员列表信息
    {
        "username": "雇员username",
        "name": "雇员name",
        "domainId": "雇员domainId",
        "userId": "雇员id",
        "avatar": "雇员头像",
        "mobile": "雇员电话号码",
        "email": "雇员email",
        "gender": "雇员性别",
        "firstNameLetter": "雇员的简拼",
        "status": "雇员激活状态",
        "sort": "雇员排序号",
        "post": "雇员职位信息",
    }
}
```

```

    "jobTitle": "雇员职位信息",
    "orgName": "雇员所属机构"
}
...
],
"hideMe" : "true", // 表示在选择联系人的时候，是否包括显示选择自己
"filterSenior":1
//1 表示过滤高管, 0 表示不过滤
}]
请求返回数据：用户选择的雇员的数据列表
[{
    "username": "雇员username",
    "name": "雇员name",
    "domainId": "雇员domainId",
    "userId": "雇员id",
    "avatar": "雇员头像",
    "mobile": "雇员电话号码",
    "email": "雇员email",
    "gender": "雇员性别",
    "firstNameLetter": "雇员的简拼",
    "status": "雇员激活状态",
    "sort": "雇员排序号",
    "post": "雇员职位信息",
    "jobTitle": "雇员职位信息",
    "orgName": "雇员所属机构"
},
{
    ...
},
{
    ...
},
...
]

```

4. 获取当前登录用户详情

获取当前登录用户详细信息(包括某个组织下的雇员信息, 开发者传参决定)

请求调用示例:

```

function getCurrentUserInfo(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "getCurrentUserInfo",
    []);
}

```

说明:

请求输入数据(可无, 默认返回数据会带上雇员信息):

```
[  
  {  
    "needEmpInfo": true //返回用户资料是否带上当前组织的雇员信息  
  }  
]
```

请求返回数据:

```
{  
  "user_id": "用户id",  
  "domain_id": "域id",  
  "username": "用户名",  
  "name": "用户名字",  
  "nickname": "用户昵称",  
  "initial": "用户简拼",  
  "avatar": "用户头像",  
  "phone": "用户电话",  
  "email": "用户邮箱",  
  "gender": "用户性别",  
  "birthday": "用户生日",  
  "status": "用户状态",  
  "first_letter": "用户首字母",  
  "more_info": "更多信息",  
  "employee": {  
    "domain_id": "域id",  
    "org_code": "组织id",  
    "employee_type_id": "雇员类型id",  
    "employee_type": "雇员类型",  
    "id": "雇员id",  
    "name": "雇员名",  
    "avatar": "雇员头像",  
    "status": "雇员状态",  
    "type": "类型",  
    "sort_order": "排序",  
    "user_id": "用户id",  
    "senior": "是否高管",  
    "nickname": "昵称",  
    "username": "用户username",  
    "display_name": "显示名称",  
    "gender": "昵称",  
    "pinyin": "拼音",  
    "initial": "简拼",  
    "mobile": "电话",  
    "created": "创建日期",  
    "last_modified": "最后修改时间",  
    "expired": "过期时间",  
    "disabled": "是否不可用",  
    "positions": "职位",  
    "more_info": "更多信息",  
    "settings": "设置信息",  
  }  
}
```

```

    "properties": "properties属性列表",
    "data_schemas": "dataSchema对应properties值",
    "industry": "行业",
    "email": "邮箱",
    "sn": "工号",
    "label": "显示值",
    "region": "区 (比如广东省) ",
    "location": "地区 (比如广州市) ",
    "other_email": "其他邮箱",
    "other_phone": "其他电话",
    "work_phone": "工作电话",
    "fax": "传真",
    "tel": "电话",
    "birthday": "生日"
}
}
}

```

5. 返回当前用户的雇员信息

获取当前登录用户的某个组织里的雇员信息

请求调用示例:

```

function getCurrentEmployeeInfo(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "getCurrentEmployeeInfo",
    []);
}

```

说明:

请求输入数据(可无, 默认返回当前组织的雇员):

```
[
{
    "orgCode": "xxx" //表示返回该组织的雇员信息
}
]
```

请求返回数据:

```
{
    "username": "雇员username",
    "name": "雇员name",
    "tenantId": "雇员domainId",
    "userId": "雇员id",
    "avatar": "雇员头像",
    "mobile": "雇员电话号码",
}
```

```

    "email":"雇员email",
    "gender":"雇员性别",
    "firstNameLetter":"雇员的简拼",
    "status":"雇员激活状态",
    "sort":"雇员排序号",
    "post":"雇员职位信息",
    "jobTitle":"雇员职位信息",
    "orgName":"雇员所属机构"
}

```

6. 显示与该用户的聊天页面

轻应用输入userId, domainId, 跳转到workplus与该用户的聊天页面

请求调用示例:

```

function showUserChatViewByUser(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "showUserChatViewByUser",
    [
        {
            "userId":"xxx",
            "domainId":"xxx"
        }
    ]);
}

```

说明:

请求输入数据:

```

[
    {
        "userId":"xxx",
        "domainId":"xxx"
    },
]

```

请求返回数据: 无

7. 获取手机通讯录里面的联系人列表

获取手机通讯录联系人的列表

请求调用示例:

```

function getMobileContacts(){
}

```

```

cordova.exec(function(result) {
    alert(JSON.stringify(result, null, 4));
},
function(error) {
    alert("调用失败");
},
"WorkPlus_Contact",
"getMobileContacts",
[]);

}

```

说明:

请求输入数据: 无

请求返回数据:

```

{
    "a": {"name": "allin(联系人名称)", "tel": "123456(联系方式)"}
    "b": [{"name": "ball(联系人名称)", "tel": "135xxx(联系方式)"}, {"name": "boy(联系人名称)", "tel": "137xxx(联系方式)"}, ...],
    ...
}

```

8. 根据用户id获取用户信息

通过用户域id和用户id获取这个用户的详细信息

请求调用示例:

```

function getUserInfoByUserId(){
cordova.exec(function(result) {
    alert(JSON.stringify(result, null, 4));
},
function(error) {
    alert("调用失败");
},
"WorkPlus_Contact",
"getUserInfoByUserId",
[{
    "user_id": "用户id",
    "domain_id": "用户域id"
}]);
}

```

说明:

请求输入数据:

```

[{
    "user_id": "用户id",
    "domain_id": "用户域id"
}]

```

请求返回数据:

```
{
    "user_id": "用户id",
    "domain_id": "域id",
    "username": "用户名",
    "name": "用户名字",
    "nickname": "用户昵称",
    "initial": "用户简拼",
    "avatar": "用户头像",
    "phone": "用户电话",
    "email": "用户邮箱",
    "gender": "用户性别",
    "birthday": "用户生日",
    "status": "用户状态",
    "first_letter": "用户首字母",
    "more_info": "更多信息"
}
```

9. 根据用户名username展示用户详情页面

通过用户名username展示这个用户的详情页面

请求调用示例:

```
function showUserInfoByUsername(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "showUserInfoByUsername", [
        {
            "username": "15918723567"
        }]);
}
```

说明:

请求输入数据:

```
[{
    "username": "15918723567"
}]
```

请求返回数据:

无

10. 根据群组id打开群组聊天页面 (WorkPlus版本3.1.4以上版本使用)

通过群组id打开该群组的聊天页面，并且可以自定义发送应用所支持的消息

请求调用示例:

```
function openDiscussionById(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert(JSON.stringify(error, null, 4));
    },
    "WorkPlus_Contact",
    "openDiscussionById", [
        {
            "discussionId": "0629aad170784c9e810985c443fb0bd6",
            "body_type" : "SHARE",
            "body" : {
                "display_avatar":"***",
                "share_type":"LINK",
                "share_message": {
                    "avatar":"http://www.icon",
                    "digest":"这里显示摘要",
                    "url":"www.baidu.com"
                }
            }
        }
    ]);
}
```

说明:

请求输入数据:

```
[{
    "discussionId":"群组id"
}]
```

请求返回数据:

无

11. 调用组织选人页面，选人创建群组 (WorkPlus版本3.1.4以上版本使用)

通过调用原生选人界面，选择人员，创建群组，并且返回相关群组信息

请求调用示例:

```
function createDiscussionChat(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert(JSON.stringify(error, null, 4));
    },
    "WorkPlus_Contact",
    "createDiscussionChat", []);
```

```

function(error) {
    alert(JSON.stringify(error, null, 4));

},
"WorkPlus_Contact",
"createDiscussionChat", []);
}

```

说明:

请求输入数据: 无

请求返回数据:

12. 打开应用聊天界面 (WorkPlus版本3.4.1以上版本使用)

通过应用id和orgId的，打开改组织下某个应用聊天界面

请求调用示例:

```

function showAppChatViewById() {

    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));

    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "showAppChatView", [
        {
            "app_id": "eb66ba3d-1d83-4104-8119-c6fc3b12cdca",
            "domain_id": "atwork",
            "org_id": "37a2ef9f-8683-4369-ae7b-121cea149d05",
            "session_type": "service"
        }]);
}

}

```

说明:

domain_id为当前应用所在域的id

org_id为当前应用所在组织的id

app_id为当前应用的id

session_type为当前应用的类型: 主要类型有[service / lightApp] ,按其类型填入上述值即可
service为app是服务号, lightApp为当前应用是轻应用

请求输入数据:

```

"app_id": "eb66ba3d-1d83-4104-8119-c6fc3b12cdca",
"domain_id": "atwork",
"org_id": "37a2ef9f-8683-4369-ae7b-121cea149d05",

```

```
"session_type": "service"
```

请求返回数据:

13. 打开搜索页面 (WorkPlus版本3.4.1以上版本使用)

请求调用示例:

```
function searchInApp() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert(JSON.stringify(error, null, 4));
    },
    "WorkPlus_Contact",
    "searchInApp", []);
}
```

请求返回数据:

14. 打开App列表页面 (WorkPlus版本3.4.1以上版本使用)

根据组织id打开该组织下的应用列表页面

请求调用示例:

```
function showAppListById() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_AppStore",
    "showAppListById",
    [{"org_id": "37a2ef9f-8683-4369-ae7b-121cea149d05"}]
);
}
```

请求输入数据:

```
"org_id": "37a2ef9f-8683-4369-ae7b-121cea149d05",
```

org_id为需要显示应用列表的组织id, 如果用户不填, 我们将自动打开当前用户默认组织下的应用列表

15. 打开通用选择人员界面, 多选用户或者雇员 (WorkPlus版本3.6.0以上版本使用)

区别于接口: 1. getContact 2. getContacts 3. getEmployeesFromCurrentOrg, 该接口打开通用的选择界面,此界面能选择组织内的雇员信息,用户的星标联系人等

请求调用示例:

```
function selectContacts() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_Contact",
    "selectContacts",
    []
);
}
```

请求输入数据:

```
[
{
    "filterSenior": 1,
    //1 表示过滤高管, 0 表示不过滤,默认为 1,
    "maxCount" : 10
    //选人最大数量, 最小为1, 默认不限制数量
}
]
```

请求返回数据:

```
{
    "user": [{...}, {...}],
    "employee": [{...}, {...}]
}
//具体模型字段参照 1. getEmployeesFromCurrentOrg
```

文件插件(WorkPlus_Files)

1. 单选文件

打开WorkPlus文件选择管理器，选择单个文件返回

请求调用示例:

```
function selectFile (){  
    cordova.exec(function(result) {  
        alert(JSON.stringify(result, null, 4));  
    },  
    function(error) {  
        alert("调用失败");  
    },  
    "WorkPlus_Files",  
    "selectFile",  
    []  
);  
}
```

说明:

请求输入数据: 无

请求返回数据:

```
[  
    {"  
        "filePath": "文件在本机的路径"  
    }]
```

2. 多选文件

打开WorkPlus文件选择管理器，选择多个文件返回

请求调用示例:

```
function selectFiles(){  
    cordova.exec(function(result) {  
        alert(JSON.stringify(result, null, 4));  
    },  
    function(error) {  
        alert("调用失败");  
    },  
    "WorkPlus_Files",  
    "selectFiles",  
    []  
);  
}
```

```
说明:  
请求输入数据: 无  
请求返回数据:  
[  
  {  
    "filePath": "文件在本机的路径"  
  },  
  ...  
]
```

3. 打开邮件附件

打开指定路径的文件，并预览

请求调用示例:

```
function openEmailAttachment(){  
  cordova.exec(function(result) {  
    alert(JSON.stringify(result, null, 4));  
  },  
  function(error) {  
    alert("调用失败");  
  },  
  "WorkPlus_Files",  
  "openEmailAttachment",  
  [{"uri":"xx", "type":"xxx"}]  
);  
}
```

```
说明:  
请求输入数据:  
[  
  {  
    "uri": "xx",  
    "type": "xxx"  
  }  
]  
请求返回数据: 无
```

4. 获取邮件附件目录

获取当前登录帐号的邮件附件目录

请求调用示例:

```
function getEmailAttachmentDir(){
```

```

cordova.exec(function(result) {
    alert(JSON.stringify(result, null, 4));
},
function(error) {
    alert("调用失败");
},
"WorkPlus_Files",
"getEmailAttachmentDir",
[]
);
}

```

说明:

请求输入数据: 无

请求返回数据:

```

{
    "message": "xxxx",           //邮件附件路径
    "status": "xxx"
}

```

5. 新增文件选择接口(返回带mediald方式)(Workplus 3.1.3版本以上使用)

打开WorkPlus文件选择管理器，根据参数状态决定单选或多选文件，并上传该文件到后台，返回值带mediald 请求调用示例:

```

function chooseFiles() {
cordova.exec(function(result) {
    alert(result)
},
function(error) {
    alert(error)
},
"WorkPlus_Files",
"chooseFiles", [{"multiple":true, "file_limit": {"max_select_count": 12, "single_select_size": -1, "total_select_size": -1}, "file_keys":[]}]
);
}

说明:
请求输入数据:
{
    "multiple": true, //单选, 多选
    "file_limit": {
        "max_select_count": 12, //多选时数量的限制
        "single_select_size": -1, //单个文件选择大小限制(byte), -1表示不限制
        "total_select_size": -1 //所有文件选择大小限制(byte), -1表示不限制
    },
    "file_keys": [] //已选文件列表(file path)
}

```

请求返回数据:

```
[
  {
    "filePath": "文件在本机的路径"
    "mediaId": "该文件在后台的媒体mediaId"
    "name" : "文件名字"
  },
  ...
]
```

6. 获取 workplus 文件目录路径(Workplus 3.1.4版本以上使用)

```
function getUserFilePath() {
  cordova.exec(function(result) {
    alert(result)
  },
  function(error) {
    alert(error)
  },
  "WorkPlus_Files",
  "getUserFilePath", [{"system" : "file"}]
);
}
```

说明:

请求输入数据:

```
{
  "system" : "file",
  "custom" : "xxxx"
}
```

system: 表示workplus app 定义的路径

目前支持"file", "dropbox"类型, 分别获取下载文件目录, 跟网盘目录

custom: 轻应用自定义的用户目录, 如传入参数"custom" : "abc", app 将创建sdcard_root/app_file/use
rname/abc 的文件目录, 并返回路径给调用者

以上参数 system 优先级比较高, 即两个类型同时传入时, 以"system"为准

7. 打开文件详情

通过指定 mediald, fileName 等参数, 打开文件详情界面

```
function openFileDetail() {
  cordova.exec(function (result) {
    alert(JSON.stringify(result, null, 4));
  },
  function (error) {
    alert(error);
  },
  "WorkPlus_Files",
  "openFileDetail", [{"id" : "1234567890"}]
);
```

```

    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_Files",
    "showFile",
    [
    {
        "fileName": "xxx",
        "fileSize": xxx,
        "mediaId" : "Z3JvdXAxL00wMC8wOS82RS9yQkFDLUzzV1EwMkFJb1F5QUFDb293akxFY
jQ5NjIuanBn",
        "isImage": false
    }
];
}

```

说明:

请求输入数据:

```

[{
    "fileName": "xxx",
    "fileSize": xxx,
    "mediaId" : "xxx",
    "isImage": false //指定文件是否是图片, 若为 true, 客户端则强制打开图片预览界面, false 则通过
fileName后缀来判断文件类型, 非图片则使用文件详情界面
}]

```

8. 本地打开文件

原生打开文件(pdf, excel, word, ppt 等)

```

function readFile() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_Files",
    "readFile",
    [
    {
        "path": "xxxx"
    }
];
}

```

9. 判断文件是否存在

判断指定路径的文件是否存在

```
function isFileExist() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_Files",
    "isFileExist",
    [
        {
            "path": "xxxx"
        }
    ];
}
```

请求返回数据:

```
[
{
    "exist" : true //文件是否存在
}]
```

登录用户权限信息插件(WorkPlus_Auth)

1. 获取临时ticket接口

为当前登录用户获取一个临时性的ticket

请求调用示例:

```
function getUserTicket(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Auth",
    "getUserTicket",
    []);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "user_ticket":"申请到的用户ticket"
}
```

2. 获取当前登录用户的accessToken

当前登录用户的accessToken

请求调用示例:

```
function getAccessToken(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Auth",
    "getAccessToken",
    []);
}
```

说明:

请求输入数据: 无

```
请求返回数据:
{
    "access_token": "用户登陆的access_token"
}
```

3. 获取当前app请求后台api地址

获取当前app请求后台api地址

请求调用示例:

```
function getServerInfo(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Auth",
    "getServerInfo",
    []);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "api_url": "app后台地址"
}
```

4. 告诉WorkPlus当前的accessToken已过期

通过轻应用告知当前accessToken过期, workplus推出当前登录状态

请求调用示例:

```
function onAccessTokenOverdue(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Auth",
    "onAccessTokenOverdue",
    []);
}
```

说明:

请求输入数据: 无

请求返回数据: 无
app弹框请求退出重新登录

5. 获取当前的租户id

获取当前租户id, 即域id

请求调用示例:

```
function getTenantID(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Auth",
    "getTenantID",
    []);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "domain_id": "当前的域id"
    -----v2数据-----
    "tenant_id": "当前的域id"
}
```

6. 获取当前登录用户最基本详情

获取当前登录用户最基本详情, 包括token, 用户信息

请求调用示例:

```
function getLoginUserInfo(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Auth",
    "getLoginUserInfo",
    []);
}
```

说明:

```
请求输入数据: 无
请求返回数据:
{
  {
    "login_token": //当前登陆的accessToken信息
    {
      "access_token": "当前登陆的access_token",
      "refresh_token": "刷新token",
      "issued_time": "生效时间",
      "expire_time": "过期时间",
      "client_id": "所持有的用户的用户id"
    }
  },
  {
    "login_user":
    {
      "username": "用户注册username",
      "name": "用户名",
      "avatar": "用户头像"
    }
  }
}
```

网页的相关插件调用(WorkPlus_WebView)

1. 打开WorkPlus里已有的html

传入相对地址，打开WorkPlus里已有的网页

请求调用示例:

```
function openLocalURL(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "openLocalURL",
    [{"localURL":"打开本地所在的html文件地址"}]
);
}
```

说明:

请求输入数据: [{"localURL":"打开本地所在的html文件地址"}]

请求返回数据: 无

2. 打开一个网页(Workplus 3.1.3以上版本新增 use_android_webview参数)

传入地址，打开WorkPlus网页

请求调用示例:

```
function openWebView(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "openWebView",
    [
        {
            "url":"打开的网址url",
            "title":"打开网页的标题",
            "use_android_webview": true, //是否使用 android 原生 webview 打开，否则使用workplus 订制的 webview
            "display_mode": "FULL_SCREEN"//全屏打开 webview(即不包含原生标题栏)，默认非全屏
        }
    ]
};
```

```
        }
    ]
);
}

说明:
请求输入数据:
[
{
    "url": "打开的网址url",
    "title": "打开网页的标题"
}
]
```

请求返回数据: 无

3. 锁定网页顶部栏

锁定顶部栏，使得顶部栏按钮无效

请求调用示例:

```
function navigation(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "navigation",
    ["lock"/"unlock"]
);
}
```

说明:

请求输入数据: ["lock"]

请求返回数据: 无

4. 更换左侧按钮动作

更换左侧的按钮动作

请求调用示例:

```
function leftButton(){
    cordova.exec(function(result) {
```

```

        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "leftButton",
    ["back"]
);
}

function back(){
    alert("想返回，没门");
}

```

说明：

调用此cordova后，用户点击返回，会弹出一个【想返回，没门】的alert，用户无法返回

5. 更换右侧按钮动作(在workplus3.1.3版本之后新增disable参数)

更换右侧的按钮动作

请求调用示例：

```

function rightButtons(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "rightButtons",
    [
        [
            [
                {
                    "disable":"false",
                    "icon":"图标",
                    "title":"标题",
                    "action":"动作 list等",
                    "value":""
                }
            ...
        ]
        ...
    ]
);
}

```

说明:

请求输入数据:

```
[  
[  
{  
"disable":"按钮是否可用 true (不可用) / false (可用)"  
,"icon":"图标",  
"title":"标题",  
"action":"动作 list等",  
"value":""  
}  
...  
]  
...  
]
```

请求返回数据: 无

6. 更换头部title

更换头部title

请求调用示例:

```
function title(){  
cordova.exec(function(result) {  
    alert(JSON.stringify(result, null, 4));  
},  
function(error) {  
    alert("调用失败");  
},  
"WorkPlus_WebView",  
"title",  
["头部显示的标题"]  
);  
}
```

说明:

请求输入数据: ["头部显示的标题"]

求返回数据: 无

.7 清除左侧按钮

清除左侧按钮事件和显示

请求调用示例:

```
function clearLeftButton(){  
cordova.exec(function(result) {  
    alert(JSON.stringify(result, null, 4));  
},
```

```
        },
        function(error) {
            alert("调用失败");
        },
        "WorkPlus_WebView",
        "clearLeftButton",
        []
    );
}
```

说明：

请求输入数据：无

求返回数据：无

8. 清除右侧按钮

清除右侧按钮事件和显示

请求调用示例:

```
function clearRightButtons(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "clearRightButtons",
    []
);
}
```

说明：

请求输入数据：无

求返回数据：无

9. 退出webview

关闭整个webview

请求调用示例:

```
function exit(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
}
```

```

    "WorkPlus_WebView",
    "exit",
    []
);
}

```

说明:

请求输入数据: 无

求返回数据: 无

10. 弹出分享对话框

弹出分享对话框, 跳转分享

请求调用示例:

```

function share(){
cordova.exec(function(result) {
    alert(JSON.stringify(result, null, 4));
},
function(error) {
    alert("调用失败");
},
"WorkPlus_WebView",
"share",
[{"url":"http://www.baidu.com","title":"百度一下","cover_media_id": "分享图标mediaId"
, "scope" : 0, "summary": "abc"}]
);
}

```

说明:

请求输入数据:

"url": "http://www.baidu.com"

"title": "百度一下"

"cover_media_id": "分享图标mediaId"

"scope": 0 // 该字段是用于显示范围, 0 只显示内部分享的操作, 包括"发送给", "分享到圈子", 1则显示全部

"summary" : "abc" 分享链接的摘要

求返回数据: 无

11. 跳转到workkplus特定页面

跳转到workkplus特定页面

请求调用示例:

```

function toActivity(){
cordova.exec(function(result) {
    alert(JSON.stringify(result, null, 4));
}

```

```

    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "toActivity",
    [
    {
        "activity":"toMain",
        "client_id":"用户id",
        "next_url":"www.baidu.com",
        "access_token":"123456789",
        "username":"用户注册username",
        "name":"用户名"
    }]
);
}

```

说明:

请求输入数据:

```

[{
    "activity":"toMain",
    "client_id":"用户id",
    "next_url":"www.baidu.com",
    "access_token":"123456789",
    "username":"用户注册username",
    "name":"用户名"
}]

```

请求返回数据: 无

12. 是否隐藏或显示左侧的按钮和关闭字样

是否隐藏或显示左侧的按钮和关闭字样

请求调用示例:

```

function visibleLeftButton(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "visibleLeftButton",
    [
    {
        "showBack":"是否显示返回图标 true/false",
        "showClose":"是否显示关闭图标 true/false",
    }]
);
}

```

}

说明:

请求输入数据:

```
[{
    "showBack": "是否显示返回图标 true/false",
    "showClose": "是否显示关闭图标 true/false",
}]
```

求返回数据: 无

13. 更换左侧侧按钮和定义动作(在workplus3.1.3版本之后新增disable参数)

更换左侧的按钮动作

请求调用示例:

```
function changeLeftButton(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "changeLeftButton",
    [
        [
            {
                "disable": "false",
                "icon": "图标",
                "title": "标题",
                "action": "动作 list等",
                "value": ""
            }
        ...
    ]
    ...
]);
}
```

说明:

请求输入数据:

```
[
    [
        {
```

```

    "disable":"按钮是否可用 true (不可用) / false (可用) "
    "icon":"图标",
    "title":"标题",
    "action":"动作 list等",
    "value":""}
}
...
]
...
]
请求返回数据: 无

```

14. 控制屏幕旋转(WorkPlus版本3.1.3以上版本使用)

根据参数控制屏幕横屏显示或者竖屏显示

请求调用示例:

```

function changeOrientation(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "changeOrientation",
    [{"landscape": true, "lock" : false}]
);
}

```

说明:

请求输入数据:

```

[
{
    "landscape": true, //表示横屏或者竖屏(可不传)
    "lock": true
    //true 表示锁定当前横屏或者竖屏的状态, false 表示 app 随用户系统设置来控制旋转, **该参数默认为 false**
}
]
请求返回数据: 无

```

15. 水印cordova

给页面添加水印

```

cordova.exec(function(winParam) {
},

```

```

        function(error) {
            alert("调用失败");
        },
        "WorkPlus_WebView",
        "addWaterMask",
        [
        {
            "textColor" : "#000000",
            "orgId": "XXX",
            "alpha": 1.0,
            "verticalPadding": 40,
            "fontSize": 16
        }]);
    
```

说明：

请求输入数据：

```

[
{
    "textColor" : "#000000", // 字体颜色
    "orgId": "XXX", // 组织id, 传值则以这个组织下的雇员名显示内容, 可不传, 默认当前组织
    "alpha": 1.0, // 水印文字透明度 0-1.0之间, 可不传, 默认1.0
    "verticalPadding": 40, // 水印文字上下的垂直高度(密度), 可不传, 默认40
    "fontSize": 16 // 字体大小, 可不传, 默认16
}
]
    
```

请求返回数据：无

移除水印

```

cordova.exec(function(winParam) {
},
function(error) {
    alert("调用失败");
},
"WorkPlus_WebView",
"removeWaterMask", []);
    
```

16. 注册/注销摇一摇监听

注册摇一摇监听

```

function registerShakeListener() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "registerShakeListener", []);
    
```

```
}
```

前端需要实现 `onWorkplusShake()` 方法, 在该处执行自己的业务, 摆一揺后将在该方法回调.

注销揆一揺监听

```
function unregisterShakeListener() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_WebView",
    "unregisterShakeListener", []);
}
```

17. 微信分享(会话/朋友圈)

根据接口直接调起微信分享页面

请求调用示例:

```
function wxShare() {
    cordova.exec(function (result) {
        alert("微信分享成功");
    },
    function (error) {
        alert("微信分享失败");
    },
    "WorkPlus_WebView",
    "wxShare", [
        {
            "app_id": "wx63f4a539bf345965",
            "title": "这是一个标题",
            "type" : "webpage",
            "description": "这是概要, 你信不信",
            "thumb": "base64://xxxxx==",
            "scene": 0,
            "data" : { "url" : "https://www.baidu.com" }
        }]);
}
```

说明：

请求输入数据：

```
app_id : 微信的AppId  
title: 分享的标题  
type: 分享的消息类型, webpage(网页链接), image(图片)  
description: 描述  
thumb: 缩略图数据  
scene: 0 为聊天界面 1为朋友圈  
data: {"url": "www.baidu.com"} or {"image": "base64://AAAAAA=="}  
type为webpage时 url字段为分享的链接  
type为image时 image为分享的图片链接或者是base64
```

地理位置信息插件(WorkPlus_Location)

1. 获取定位信息

返回地理位置信息

请求调用示例:

```
function getLocation(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Location",
    "getLocation",
    []
);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "latitude": "xxxx",
    "longitude": "xxx",
    "address" : "广州市天河区 xxx 街道 xxx" //具体地址信息,
    "city" : "xxx", //广州市
    "district" : "xxx", //天河区
    "street" : "xxx", //翰景路
    "aoiName" : "xxx" //金星大厦
}
```

开启自动签到

```
function OpenSignIn(){
    cordova.exec(function(result) {
        alert("开启成功");
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Location",
    "enableOrgSignIn",[]);
}
```

```
}
```

调用此接口，开启的组织为当前用户选择的组织

关闭自动签到

```
function CloseSignIn(){
    cordova.exec(function(result) {
        alert(result);
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Location",
    "disableOrgSignIn",[]);
}
```

调用此接口，关闭的组织为当前用户选择的组织

设备插件(WorkPlus_DeviceInfo)

1. 获取IP地址

获取当前连接状态下的IP地址

请求调用示例:

```
function getIpAddress(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_DeviceInfo",
    "getIpAddress",
    []
);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "result": "ok",
    "ipAddress": "手机连接网络的ip地址"
}
```

2. 获取设备信息

获取当前设备和WorkPlus的一些信息

请求调用示例:

```
function getDeviceInfo(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_DeviceInfo",
    "getDeviceInfo",
    []
);
```

```
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "device_id": "设备id",
    "platform": "平台 (android/ios) ",
    "domain_id": "域id",
    "product_version": "workplus当前应用的versionCode",
    "system_version": "使用的系统版本 (ios9 / android6.0 等。。) ",
    "system_model": "使用的手机型号 (iphone7 / android sumsang等。。) ",
    "channel_id": "应用的包名",
    "channel_vendor": "Apple or ",
}
```

3. 获取设备的运动数据

获取设备当天凌晨0点至当前时间的运动数据 (从版本3.12.1起提供)

请求调用示例:

```
function getPedometerData(){
    cordova.exec(function(result){
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
    },
    "WorkPlus_DeviceInfo",
    "getPedometerData",
    [{}]);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "steps": "步数",
    "distances": "距离, 单位米",
}
```

如果当前设备计步器不可用, 返回code = -10001

获取App配置插件(WorkPlus_LightApp)

1. 获取app请求的api地址

获取app请求的api地址

请求调用示例:

```
function getConfig(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_LightApp",
    "getConfig",
    ["shakeUrl", "colleagueCircleUrl",...]
);
}
```

说明:

请求输入数据:

根据需要的配置传入如下参数: shakeUrl (摇一摇地址) , colleagueCircleUrl (同事圈地址) , _ApiUrl (轻应用管理后台地址) , _AdminUrl (管理平台地址) , scheduleUrl (日程管理地址) ;

请求返回数据:

根据传入的参数, 决定返回的数据

```
{
    "shakeUrl": "xxx",
    "colleagueCircleUrl": "xxx",
    ....,
    "status": 0
}
```

扫描二维码插件(WorkPlus_BarcodScanner)

调用相机扫描二维码

请求调用示例:

```
function scanner(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_BarcodScanner",
    "scanner",
    []
);
}
```

说明:

请求输入数据: 无

请求返回数据: 无

应用市场插件(WorkPlus_AppStore)

1. 添加应用

添加应用

请求调用示例:

```
function installApp(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_AppStore",
    "installApp",
    [{"app_id":"xxx"}]
);
}
```

说明:

请求输入数据: [{"app_id": "xxx"}]

请求返回数据:

```
{
    "message": "xxxx",
    "status": "xxx"
}
```

2. 原生app 路由跳转(WorkPlus版本3.1.3以上版本使用)

根据参数跳转指定app

请求调用示例:

```
function routeApp(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_AppStore",
    "route",
    [
        {
            "scheme_url": ""
        }
    ]
);
```

```
"android_explicit_intent": {  
    "package": "com.cimc.app",  
    "init_url": "com.cimc.app.activity",  
    "params": {  
        "xxx": "xxx",  
        "aaa": "bbb"  
    }  
}  
}  
]  
);  
}
```

说明:

请求输入数据:

```
[  
{  
    "scheme_url": "",  
    "android_explicit_intent": {  
        "package": "com.cimc.app", //包名  
        "init_url": "com.cimc.app.activity", //跳转的具体页面  
        "params": { //携带参数  
            "xxx": "xxx",  
            "aaa": "bbb"  
        }  
    }  
}  
]  
scheme_url: ios 跟 android 都支持  
android_explicit_intent: 为 android 的传参形式, ios 不支持该方式.
```

请求返回数据: 无

公有云插件(WorkPlus_PublicCloud)

1. 跳转到个人详情页面

通过轻应用数据，跳转到个人详情页面

请求调用示例:

```
function toPersonal(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_PublicCloud",
    "toPersonal",
    [
        {
            "finish_view":false
            "user_id":"用户id",
            "username":"用户username",
            "mobile":"用户手机",
            "domain_id":"用户域id"
        }
    ]
);
}
```

说明:

请求输入数据:

```
[
{
    "user_id":"用户id",
    "domain_id":"用户域id",
    "username":"用户username",
    "mobile":"用户手机",
    "registered": false, 是否为注册用户(非必填, 默认 false)
    "finish_view": true, 是否关闭当前页面 (非必填, 默认 true)
}
]
```

请求返回数据: 无

2. 获取WorkPlus信息

获取workplus版本信息

请求调用示例:

```
function getAppInfo(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_PublicCloud",
    "getAppInfo",
    []
);
}
```

说明:

请求输入数据: 无

请求返回数据:

```
{
    "version_code": "应用当前版本versioncode",
    "version_name": "应用当前版本名称",
    "app_name": "应用名",
    "bundle_id": "应用包名/bundleId",
    "app_icon": "应用图标",
}
```

换肤主题(WorkPlus_Theme)

1. 更新皮肤

更新对应组织的皮肤包,若该组织是当前组织,则马上显示更换效果

请求调用示例:

```
function changeTheme(){
    cordova.exec(function(result) {
        alert(JSON.stringify(result, null, 4));
    },
    function(error) {
        alert("调用失败");
    },
    "WorkPlus_Theme",
    "changeTheme",
    [{"orgCode": "xxx", "theme": "xxx"}]
);
}
```

说明:

请求输入数据:

```
[  
  {  
    "orgCode": "xxx",  
    "theme": "xxx"  
  }  
]
```

theme表示主题名称, 暂时系统提供如下8种:

skyblue	天空清澈蓝
businessblue	商务蓝
shakin	耀沙金
glacierblue	冰川蓝
chinared	中国红
blackjadegreen	墨玉绿
bluestars	繁星蓝
vibrantorange	活力珠光橙

请求返回数据: 无

app native 支付(WorkPlus_Pay)

1. 微信支付

提供微信支付的能力

请求调用示例:

```
cordova.exec(function(result) {
    alert(JSON.stringify(result, null, 4));
},
function(error) {
    alert("调用失败:" + error);
},
"Workplus_Pay",
"wxPay",
[{"app_id": "abcd",
 "request_data": "xxxx"
}]);
```

说明:

请求参数 :

```
{
    "app_id": "abcd", // sdk appId
    "request_data": "xxxx" //sdk请求所需要参数，包括具体商户 id，金额，数据加密等，格式参考微信
文档PayReq类
}
```

返回参数:

```
{
    "code": "xxx", //微信支付回调结果
    "message": "xxx" //结果描述
}
```

网络连接信息(WorkPlus_NetInfo)

1. 获取当前连接Wifi信息

获取当前连接Wifi信息, 包括 bssid, 名字等

请求调用示例:

```
function getWifiInfo() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_NetInfo",
    "getWifiInfo",
    []
);
}
```

说明:

请求返回数据:

```
{
    "bssid" : "xxxx", //wifi 接入点标识
    "name" : "foss-mobile", //wifi 名称
}
```

Umeeting 会议相关接口(WorkPlus_Umeeting)

1. 开启会议

Umeeting 开启会议

请求调用示例:

```
function startMeeting() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_Umeeting",
    "startMeeting",
    [
        {
            "user_id": "xxxx",
            "zoom_token": "xxxx",
            "user_type": 99,
            "display_name": "dasynsy",
            "meeting_id": "18820274107"
        }
    ]
);
```

说明:

请求输入数据:

```
[{
    "user_id": "xxxx",
    "zoom_token": "xxxx",
    "user_type": 99,
    "display_name": "dasynsy", //昵称
    "meeting_id": "18820274107"//会议号
}]
```

请求返回数据:

```
{
    "result" : 0 //0表示调用 umeeting 接口成功, 其他状态码具体参考 umeeting 开发文档
}
```

user_type 为会议类型, umeeting启动会议所需要参数, 可不传, 默认是99(USER_TYPE_API_USER), 总共有以下三种类型(详情可参考 umeeting 开发文档):

```
int USER_TYPE_API_USER = 99;
int USER_TYPE_ZOOM = 100;
int USER_TYPE_SSO = 101;
```





2. 加入会议

Umeeting 加入会议 请求调用示例:

```
function joinMeeting() {
    cordova.exec(function (result) {
        alert(JSON.stringify(result, null, 4));
    },
    function (error) {
        alert("调用失败");
    },
    "WorkPlus_Umeeting",
    "joinMeeting",
    [
    {
        "display_name": "dasynsy",
        "meeting_id": "18820274107"
    }
];
}
```

说明:

请求输入数据:

```
[{
    "display_name": "dasynsy", //昵称
    "meeting_id": "18820274107" //会议号
}]
```

请求返回数据:

```
{
    "result" : 0 //0表示调用 umeeting 接口成功, 其他状态码具体参考 umeeting 开发文档
}
```