**使用场景**

申请各种权限, 以及一些坑

**autojs版本**



**截图权限**

**申请截图代码**

if (!requestScreenCapture()) {

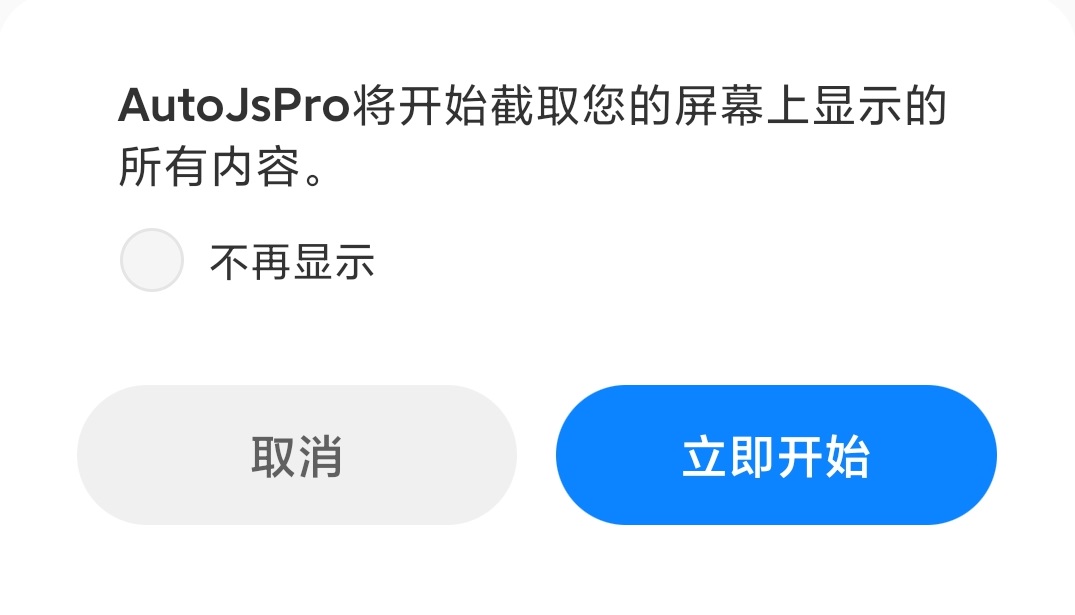
toastLog("请求截图权限 失败");

} else {

toastLog("请求截图权限 成功");

}

执行以上代码后, 正常应该弹出一个框, 让用户选择是否允许截图



但是有的时候, **不会弹出这个界面**, 这个时候,

你就要去 设置-->应用管理-->AutoJsPro-->权限管理-->后台弹出界面-->允许



autojs版本号8.8.12更新日志

更新 shizuku相关API到最新版本  
新增 images.getScreenCaptureOptions()查询当前截图配置  
新增 images.stopScreenCapture()停止截图

// false表示没有截图权限, true表示有截图权限

log(!!images.getScreenCaptureOptions());

log(images.getScreenCaptureOptions())

requestScreenCapture()

log(images.getScreenCaptureOptions())

images.stopScreenCapture()

log(images.getScreenCaptureOptions())

requestScreenCapture({async: true});

log(images.getScreenCaptureOptions())

images.stopScreenCapture()

log(images.getScreenCaptureOptions())

**悬浮窗权限**

**判断是否有悬浮窗权限的代码**

floaty.checkPermission();

返回一个布尔值, true就是有权限, false就是没有权限

如果没有, 我们就需要提示用户, 使用app需要悬浮窗权限, 然后跳转悬浮窗权限设置界面, 方便用户勾选

**跳转悬浮窗权限设置界面代码**

app.startActivity({

packageName: "com.android.settings",

className: "com.android.settings.Settings$AppDrawOverlaySettingsActivity",

data: "package:" + context.getPackageName(),

});



**后台自启权限**

以华为手机为例, 跳转启动管理页面的代码

let intent = new Intent();

intent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

let pkg = "com.huawei.systemmanager";

let cls = "com.huawei.systemmanager.startupmgr.ui.StartupNormalAppListActivity";

let componentName = new android.content.ComponentName(pkg, cls);

intent.setComponent(componentName);

context.startActivity(intent);

**微信公众号 牙叔教程**



**QQ群**

747748653

**下面的权限列表**

* 后台弹出界面权限
* 忽略电池优化权限
* 可修改系统设置权限
* 通知使用权限
* 无障碍权限
* GPS是否为打开状态
* 申请普通权限
* 调用摄像头权限
* 申请权限回调
* 跳转应用信息页

**后台弹出界面权限**

以小米手机为例, 判断是否有**后台弹出界面**权限的代码

function checkOp(ctx, op) {

try {

return ctx.getSystemService("appops").checkOp(op, android.os.Binder.getCallingUid(), ctx.getPackageName()) == 0; //MODE\_ALLOWED

} catch (e) {

log(e);

}

return false;

}

let op = 10021; // >= 23

let r = checkOp(context, op);

log(r);

// true 有权限 false 没有权限

**忽略电池优化权限**

以华为手机为例, 判断是否在忽略电池优化白名单中

function isIgnoringBatteryOptimizations() {

return context

.getSystemService(android.content.Context.POWER\_SERVICE)

.isIgnoringBatteryOptimizations(context.packageName);

}

// true 在白名单中 false 不在白名单中

以华为手机为例, 弹框是否忽略电池优化

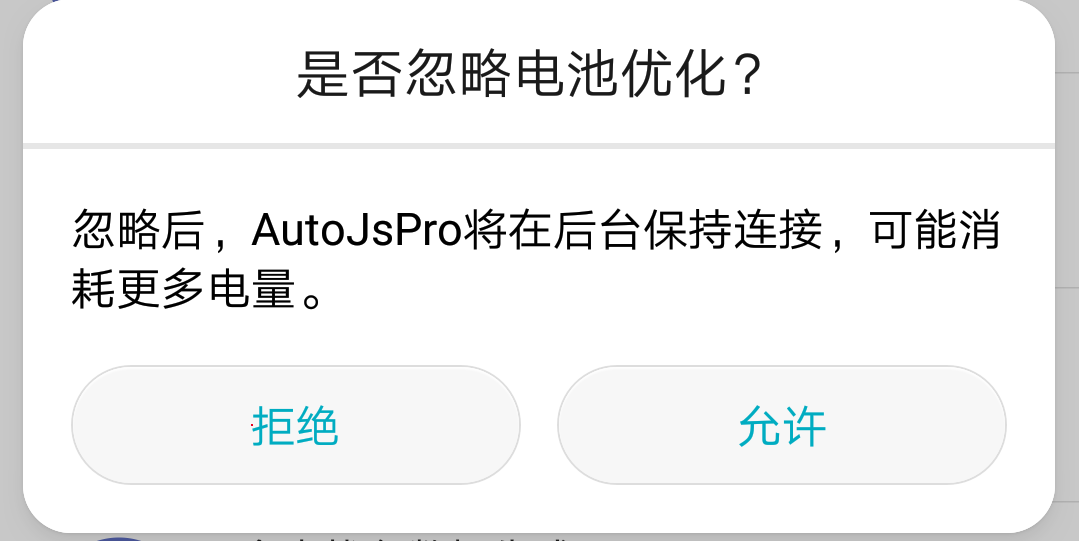
app.startActivity(

new android.content.Intent()

.setAction(android.provider.Settings.ACTION\_REQUEST\_IGNORE\_BATTERY\_OPTIMIZATIONS)

.setData(android.net.Uri.parse("package:" + context.packageName))

);



**可修改系统设置权限**

以华为手机为例,(若无特殊说明, 均为华为手机, 不一定所有华为手机都一样, 仅供参考)

是否有可修改系统设置权限

android.provider.Settings.System.canWrite(context)

跳转到可修改系统设置权限页面

app.startActivity(

new Intent(

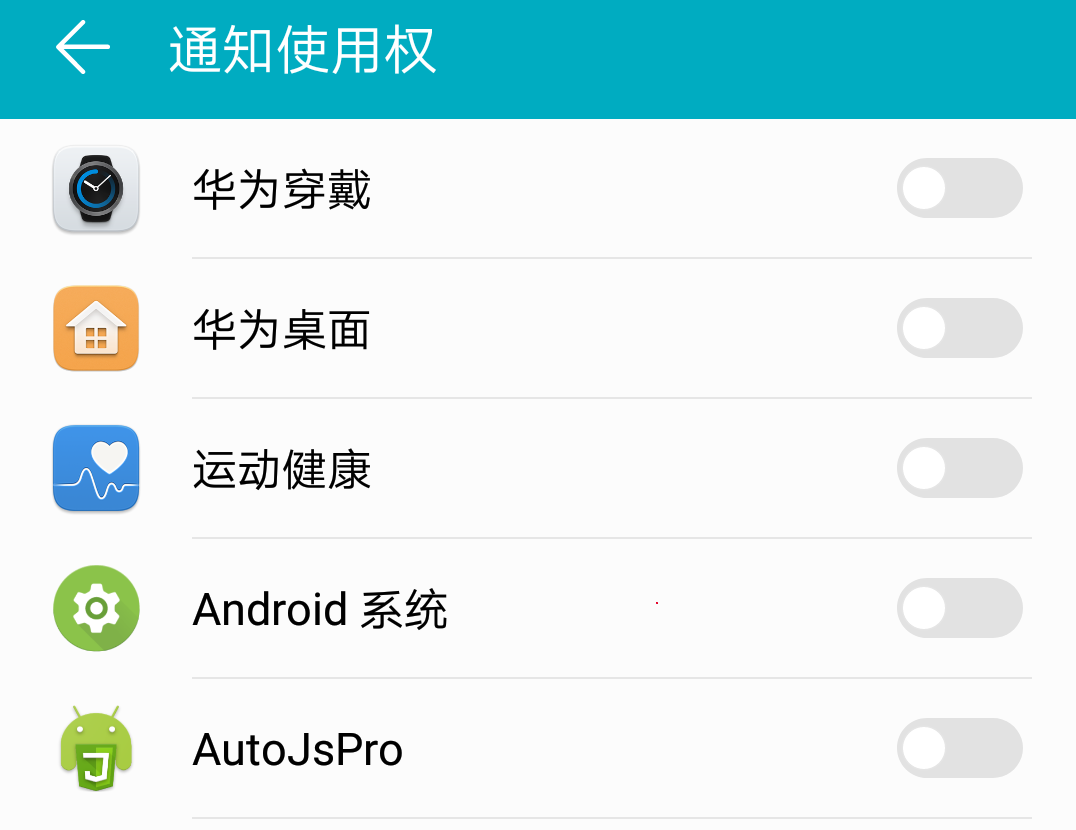
android.provider.Settings.ACTION\_MANAGE\_WRITE\_SETTINGS,

app.parseUri("package:" + context.getPackageName())

)

);

**通知使用权限**



是否有通知使用权限

function notificationListenerEnable() {

let packageName = context.getPackageName();

let data = android.provider.Settings.Secure.getString(context.getContentResolver(), "enabled\_notification\_listeners");

if (data != null) {

let enable = data.indexOf(packageName);

if (enable != -1) {

return true;

} else {

return false;

}

}

}

// true 有通知使用权 false 没有通知使用权

**无障碍权限**

是否有无障碍权限

auto.service == null

// true 没有无障碍权限 false 有无障碍权限

跳转无障碍页面

app.startActivity({

action: "android.settings.ACCESSIBILITY\_SETTINGS",

});

**GPS是否为打开状态**

判断GPS是否已经打开

var ls = context.getSystemService(context.LOCATION\_SERVICE);

if (ls.isProviderEnabled(android.location.LocationManager.GPS\_PROVIDER)) {

console.log("GPS已打开");

}

跳转到GPS设置页面

app.startActivity({

action: "android.settings.LOCATION\_SOURCE\_SETTINGS",

});

**申请普通权限**

runtime.requestPermissions([

"access\_fine\_location",

"access\_network\_state",

"access\_notification\_policy",

"body\_sensors",

"broadcast\_sticky",

"camera",

"change\_network\_state",

"expand\_status\_bar",

"foreground\_service",

"get\_accounts",

"get\_package\_size",

"install\_location\_provider",

"install\_packages",

"install\_shortcut",

"internal\_system\_window",

"internet",

"kill\_background\_processes",

"manage\_accounts",

"manage\_app\_tokens",

"master\_clear",

"modify\_audio\_settings",

"mtweak\_forum",

"mtweak\_user",

"persistent\_activity",

"process\_outgoing\_calls",

"read\_calendar",

"read\_call\_log",

"read\_cell\_broadcasts",

"read\_contacts",

"read\_external\_storage",

"read\_phone\_state",

"read\_sms",

"read\_sync\_settings",

"receive\_boot\_completed",

"receive\_mms",

"receive\_sms",

"record\_audio",

"reorder\_tasks",

"request\_install\_packages",

"restart\_packages",

"set\_alarm",

"set\_debug\_app",

"set\_time",

"system\_alert\_window",

"write\_calendar",

"write\_call\_log",

"write\_contacts",

"write\_external\_storage",

"write\_sync\_settings",

]);

**调用摄像头权限**

是否有调用摄像头权限

let permissionName = "camera";

let pm = context.getPackageManager().checkPermission("android.permission.CAMERA", context.getPackageName());

let r = android.content.pm.PackageManager.PERMISSION\_GRANTED == pm;

log(r);

// true 有权限 false 没有权限

申请调用摄像头方法一

let permissionName = "camera";

runtime.requestPermissions([permissionName]);

由于方法一触发不了**request\_permission\_result**事件

我们使用方法二申请摄像头权限

申请权限后, **判断用户是否授予权限**

ui.emitter.on("request\_permission\_result", function () {

toastLog(arguments); // { '0': 100, '1': [android.permission.CAMERA], '2': [-1] }

// java 对应的方法

// onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults)

});

androidx.core.app.ActivityCompat.requestPermissions(activity, ["android.permission.CAMERA"], 100);

**跳转应用信息页**

app.openAppSetting(context.packageName);

****