

# 如何进行自动化检测



如果大家需要复制文字、下载文件，但是却发现无法执行，请大家直接在群里反馈，谢谢！

回顾：目演示：GUI Agent如何操作APP、我如何做自动化评测



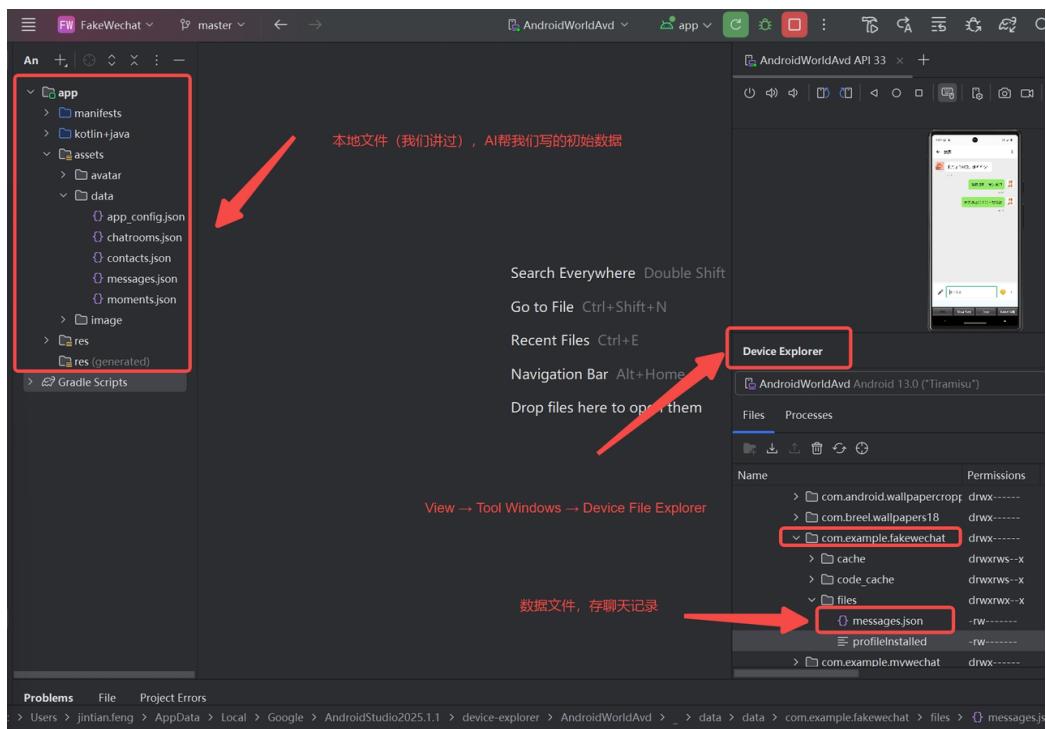
请在阅读下面的文字之前，检查github是否已经提交，确保代码是最新的。

## Windows (Mac) 本地存储和手机存储

大家先分清楚2个概念

- Windows (Mac) 本地存储就是你电脑的文件，例如"C盘/Desktop/sample.json"这就是你电脑的文件，大家应该都很熟悉
- 安卓开发是“开发APP，运行到手机上”，所以手机存储指的是**运行在手机上的APP所能访问的文件空间**，也就是**手机设备本地的存储区域**。例

如：`/data/data/com.example.fakewechat/files/` (具体可以看看下面这个图)



## 什么叫自动化检测

工作流程

1. 我提出任务，例如“订购一张10.20号武汉到北京的高铁票”
2. GUI Agent开始工作，这会经历很多个步骤，我不希望人工一步一步的检查，这太累了
3. 我希望有一个机制可以检查任务是否完成
  - a. 例如：维护一个高铁票购票历史（类似于图片中的messages.json），只要买票了我就把买票的信息（时间、出发地、到达地、价格）写入到这个文件
    - i. **这个购票历史是要维护的，也就是说这个文件和文件更新的逻辑是要你自己增加的，不会默认存在。**
  - b. Agent慢慢执行任务，等他执行完毕之后，我去上述文件检查有无“10.20 武汉-北京”的购票信息，例如下面的黄色部分：

#### 代码块

```

1  {
2      "history": [
3          {
4              "time": "2025-10-18",
5              "from": "武汉",
6              "to": "长沙",
7              "price": 185.0
8          },
9          {
10             "time": "2025-10-20",
11             "from": "武汉",
12             "to": "北京",
13             "price": 320.5
14         }
15     ]
16 }
```

0) 更多例子（要是不会，不要自己琢磨，群里问下，这个没有标准答案，只要能完成任务即可）

指令类型	检验逻辑
点击XXX（例如点击QQ音乐首页的“我喜欢”）	维护点击“我喜欢”的历史记录
数一下我有多少个待收货订单	维护一个函数，函数计算订单数量 $CheckOrderNumber(AI\_answer)$ step1. 计算真实的待收货订单数量，例如 $fact\_num$

	<i>step2. Check if AI_answer == fact_num</i>
计算10.20武汉往返长沙的高铁票总价	维护一个函数，函数计算真实价格 <i>CheckTicketPrice(AI_answer)</i> <i>step1. 计算真实的总价，例如fact_num</i> <i>step2. Check if AI_answer == fact_num</i>

## 1) 怎么“在电脑”访问“手机内部存储”

- 很简单，用adb指令即可，具体可以看这个代码（对应图片的例子）。

代码块

```

1 import subprocess
2 import json
3
4 # 从设备获取文件
5 # run里面的com.example.fakewechat是APP名字； files/messages.json是文件路径
6 subprocess.run(['adb', 'exec-out', 'run-as', 'com.example.fakewechat', 'cat',
7                 'files/messages.json'],
8                 stdout=open('messages.json', 'w'))
9
10 # 读取文件
11 with open('messages.json', 'r', encoding='utf-8') as f:
12     data = json.load(f)
13
14 print(data)

```

## 2) 怎么自动化检测

- 以任务“给北大李老师发送Hello”为例
  - 我刚才已经讲了如何访问手机内部存储
  - 我现在只需要判断我和北大李老师的聊天记录里面有没有这个“hello”

以下演示了一个自动化检验的脚本（重要）

代码块

```

1 import subprocess
2 import json
3
4 def MessageSendCheck(receiverId, senderId, message_content):
5     # 从设备获取文件
6     # run里面的com.example.fakewechat是APP名字； files/messages.json是文件路径

```

```

7     subprocess.run(['adb', 'exec-out', 'run-as', 'com.example.fakewechat',
8                     'cat', 'files/messages.json'],
9                     stdout=open('messages.json', 'w'))
10
10    # 读取文件
11    with open('messages.json', 'r', encoding='utf-8') as f:
12        data = json.load(f)
13
14    # 判断给定的信息是否存在与聊天记录里面
15    try:
16        item = data['privateChatMessages'][receiverId][-1]
17        if item['senderId'] == senderId and item['content'] == message_content:
18            return True
19        else:
20            return False
21    except:
22        return False
23
24 if __name__ == "__main__":
25     print(MessageSendCheck(
26         receiverId='user_2',      # 我的微信里面，北大李老师的id是user_2
27         senderId='current_user', # 这是“我”
28         message_content='hello'   # 发送内容
29     ))

```

### 3) 怎么检验自己的“**自动化检验的脚本**”是否正确

1. Step1: 人工模拟Agent进行任务执行（做对任务）
2. Step2: 运行脚本，看是否正确输出True
3. Step3: 人工模拟Agent进行任务执行（故意做错任务）
4. Step4: 运行脚本，看是否正确输出False

### 必读补充知识：json的格式化与非格式化

上面这个json是格式化的，也就是适合人类阅读的。

如果你在工作中遇到下面这样的json（又臭又长、难以阅读），那说明你遇到非格式化的json了，在让AI书写json的时候，一定要让他格式化保存json。

#### 代码块

```

1 {"history": [{"time": "2025-10-18", "from": "武汉", "to": "长沙", "price": 185.0},
2             {"time": "2025-10-20", "from": "武汉", "to": "北京", "price": 320.5}]}

```

# 我们要做什么

简单来说，要给每个指令写一个检验脚本，见“2) 怎么自动化检测”

(新建Autotest目录，把这个目录上传到GitHub，即可)：一个表格和一系列python文件，长成这样：

名称	修改日期	类型	大小
.claude	2025/10/15 13:54	文件夹	
.gradle	2025/11/3 7:49	文件夹	
.idea	2025/11/2 19:17	文件夹	
app	2025/10/29 17:18	文件夹	
AutoTest	2025/10/15 18:46	文件夹	
gradle	2025/9/11 14:16	文件夹	
UIReference	2025/11/2 17:06	文件夹	
.gitignore	2025/7/10 14:43	txtfile	1 KB
build.gradle.kts	2025/7/10 15:43	KTS 文件	1 KB
Develop_Requirements.md	2025/11/2 17:06	Markdown 源文件	4 KB
gradle.properties	2025/7/10 15:23	PROPERTIES 文件	2 KB
gradlew	2025/7/10 14:43	文件	6 KB
gradlew.bat	2025/7/10 14:43	Windows 批处理文件	3 KB
local.properties	2025/7/10 14:43	PROPERTIES 文件	1 KB
README.md	2025/11/2 19:17	Markdown 源文件	40 KB
settings.gradle.kts	2025/9/22 17:23	KTS 文件	2 KB

## A. 表格（飞书文档即可，模板已经搞定，大家自己填）

回任务设计&检验逻辑

## B. 系列Python文件（Autotest里面包含这些内容）

代码块

```
1 eval_1.py # 指令1的检验逻辑，见“2) 怎么自动化检测”
2 eval_2.py # 指令2的检验逻辑，见“2) 怎么自动化检测”
3 ...
```



建议每写一个eval\_xxx.py就提交一次git，一定要勤快！