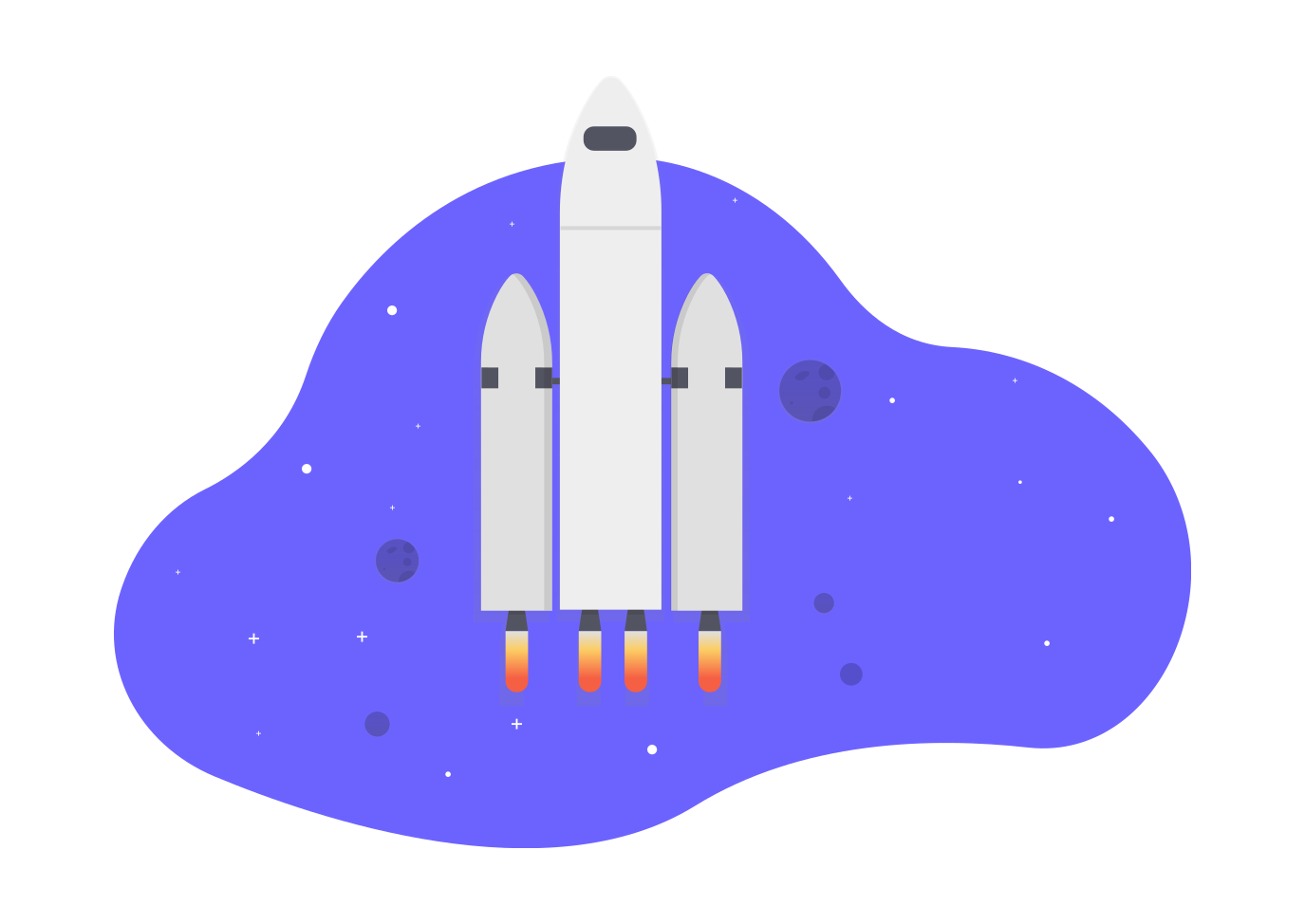
How to Use

ADB Controller

API Documentation Version 1.0



**M2 Team Solution Development Part**

How to Use ADB Controller

[1. ADB Controller 소개 3](#_Toc77602629)

[1) 프로그램 소개 3](#_Toc77602630)

[3) 실행 전 요구 사항 3](#_Toc77602631)

[2. Support API 4](#_Toc77602632)

[1) 프로그램 지원 함수 4](#_Toc77602633)

[init 4](#_Toc77602634)

[setup\_test 5](#_Toc77602635)

[set\_print 6](#_Toc77602636)

[execute\_cmd 6](#_Toc77602637)

[save\_dump\_xml 7](#_Toc77602638)

[get\_pos\_elements 8](#_Toc77602639)

[2) Value Return 함수 9](#_Toc77602640)

[cmd\_return\_getPid 9](#_Toc77602641)

[cmd\_return\_getCallState 10](#_Toc77602642)

[3) Status Return 함수 11](#_Toc77602643)

[cmd\_status\_currentActivity 11](#_Toc77602644)

[cmd\_status\_killApp 12](#_Toc77602645)

[cmd\_status\_click 13](#_Toc77602646)

[3. How to Executes 14](#_Toc77602647)

# ADB Controller 소개

1. 프로그램 소개

ADB Controller는 Python 기반으로 개발된 Android Phone 전용 자동화 실행 지원 프로그램 입니다. 해당 프로그램에서는 각종 실행 method(문자발신, 전화발수신, 비행기모드, App 실행 등등) 및 정보 확인 method API를(App PID, mCallStatus 등등) 지원하여 자동화 컨트롤이 가능하도록 하였으며 순수 ADB를 통해 Device 자동 Control 될 수 있도록 하여 별도의 라이브러리가 필요하지 않도록 하였습니다.

1. 프로그램 특징

* 복잡하지 않은 구조로 되어 있으며 다양한 자동 실행 함수 지원과 실행 자유도가 높음
* ADB 특성 상 원래 한글 입력이 어렵지만 ADB Keyboard 자동 설치 지원을 통해 가능하도록 변경(별도 설정 과정은 필요)
* Screen Capture 이미지 별도의 특정된 Folder로 저장 지원
* ADB가 Click 또는 Swipe등의 Action에서 절대 좌표 값으로 지정해야 하는 문제가 있었으나 프로그램에서 xml dump 파일을 통한 element 좌표 찾기 기능의 method를 제공하여 디바이스 변경에 따른 해상도 문제를 해결함
* 각각의 실행 스크립트를 독립적으로 따로 실행됨으로 동시에 여러 디바이스가 병렬적으로 동작함

1. 실행 전 요구 사항

* Python이(3.x 버전 이상) 설치되어 있어야 합니다. 설치는 Anaconda3 패키지로 설치하는 것을 권장합니다. 설치 관련 글은 [여기](https://somjang.tistory.com/entry/Windows-Windows%EC%97%90%EC%84%9C-Anaconda-%EC%84%A4%EC%B9%98%ED%95%98%EA%B8%B0?category=348812)를 Click 하여 참고하시기 바랍니다.
* ADB도 로컬 PC에 설치되어 있어야 합니다. 현재 Android Studio를 설치하면 자동으로 ADB가 설치되지만 용량이 무거운 관계로 Platform-Tool만 설치하고 싶다면 [여기](https://tnmsoft.tistory.com/214)를 Click하여 참고하시기 바랍니다.
* ADB가 Local PC에 설치된 상태에서 테스트 대상 단말기와 디버깅 연결을 미리 하셔야 원활한 테스트가 가능합니다.
* 전화, 문자 등등 기본 앱 설정을 미리 정하시면 ADB 실행 연결 시 팝업 발생으로 인한 테스트 중단 현상을 예방할 수 있습니다.

# Support API

1. 프로그램 지원 함수

### init

|  |
| --- |
| **Function Desc** |
| 자동화 스크립트 실행 전 CMDS Class Initialized method이며 Class instance 시 자동으로 실행되며 아래와 같이 작업됨   * 파라메터 uuid에 의해 serial\_num 값 설정 * 결정된 serial\_num에 의해 ADB cmds 값 설정 * 파라메터 divide\_window에 의해 wm\_divide\_count 값 설정 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| uuid | String | 테스트 대상 단말의 uuid(ADB상의 serial num) |
| divide\_window | Int | 테스트 대상 단말의 화면 X/Y 가상 구역 분리 개수  (Click 혹은 Swipe 시 선택될 가상의 구역) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
|  |  |  |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20) |

### setup\_test

|  |
| --- |
| **Function Desc** |
| CMDS Class 인스턴스 후 가장 먼저 호출해야 하는 함수로 여기서 디바이스 연결상태, 화면 Display 상태 등등을 Check하고 한글 지원 ADBKeyboard 가 설치됨   * ADBKeyboard 설치 절차 진행 * check\_device\_serial method 실행(단말 연결 상태 확인) * check\_device\_manuInfo method 실행(제조사 확인) * check\_device\_modelName method 실행(디바이스 모델명 확인) * check\_device\_displayOn method 실행(디바이스 화면 상태 확인) * 이미지 저장 Folder 생성 * Xml dump file 저장 Folder 생성 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
|  |  |  |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  cmd.setup\_test() |

### set\_print

|  |
| --- |
| **Function Desc** |
| Console에 print 하고 싶은 log 내용을 보여주는 method 시간 정보와 현재 디바이스 모델과 serial\_num가 log 내용과 함께 표시됨 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| text | String | Console에 보여주고 싶은 Text 내용 표시 |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
|  |  |  |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  cmd.setup\_test()  cmd.set\_print(‘테스트입니다.’)  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  테스트입니다. |

### execute\_cmd

|  |
| --- |
| **Function Desc** |
| ADB cmd를 parameter로 전달받아 해당 명령을 subprocess pipe로 Android device에 명령어 전달 지원 method이며 실행 상태에 따라 status 값과 cmd 실행 후 text를 return함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| cmd | String | ADB cmd 명령어 |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| return\_data | Array | **실행결과가 정상 Case**  [True, stdout\_text]  **실행결과가 비정상 Case**  [False, stderr\_text] |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  manu\_cmd = cmd.getManuInfo  returns = cmd.execute\_cmd(manu\_cmd)  cmd.set\_print(returns)  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  [True, '[ro.product.manufacturer]: [samsung]'] |

### save\_dump\_xml

|  |
| --- |
| **Function Desc** |
| Android activity xml dump 실행 method로 실행 시점의 화면 xml tree node 파일을 저장한다. 주로 get\_pos\_elements 실행 전에 호출하는 것이 대부분인 지원 method 이며 실행 후 ‘xmlDump’ Folder에 xml 파일이 저장함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
|  |  |  |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 Activity xml tree node dump 파일 저장  cmd.save\_dump\_xml() |

### get\_pos\_elements

|  |
| --- |
| **Function Desc** |
| 보통 save\_dump\_xml 함수 호출 후에 주로 쓰이며 앞서 저장된 xml dump file에서 parameter로 입력된 attribute와 찾고자 하는 속성 값을 입력하면 해당하는 좌표 Array 값을 return함  주요 attribute는 아래와 같으며 거의 모든 attr 유형을 지원함   * **class:** Element class => android.widget.TextView * **resource-id**: Element ID => com.android.settings:id/switch\_widget * **content-desc**: accessibility element desc => login * **text**: Element text => 사용 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| attr | String | Target Element의 attr 유형 입력(class ,resource-id..) |
| name | String | Target Element의 attr 유형 값 입력  \* 하나 이상 Case ‘|’을 구분자로 입력(확인|OK…) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| List\_pos | Array | Target Element에 해당하는 모든 elements의 좌표 값이 tuple 형태의 Array로 return 됨 |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 Activity xml tree node dump 파일 저장  cmd.save\_dump\_xml()  # Target Element get pos   list\_pos = cmd.get\_pos\_elements(attr=’content-desc’, name=’공유하기’)  # list\_pos 값 확인  cmd.set\_print(list\_pos)  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  [(235.25, 1220.73)] |

1. Value Return 함수

### cmd\_return\_getPid

|  |
| --- |
| **Function Desc** |
| App package name을 parameter로 입력하면 현재 android device process 목록에서 해당 App의  Pid를 return함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| name | String | Target App의 package 이름 입력(com.kakao.talk) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| pid | String | **실행결과가 정상 Case**  해당 App pid 값(27096)  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 Activity xml tree node dump 파일 저장  pid = cmd.cmd\_return\_getPid(name=’com.kakao.talk’)  cmd.set\_print(pid)  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  27096 |

### cmd\_return\_getCallState

|  |
| --- |
| **Function Desc** |
| Adnroid Device에서 Call State 값을 확인하여 return 함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| mCallState | String | **실행결과가 정상 Case**  “**idle**”(대기), “**ringing**”(수신 중), “**activity call**”(통화 중) 이들 중 하나가 리턴  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 Activity xml tree node dump 파일 저장  mCallState = cmd.cmd\_return\_getCallState()  cmd.set\_print(mCallState)  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  idle |

1. Status Return 함수

### cmd\_status\_currentActivity

|  |
| --- |
| **Function Desc** |
| Foreground에서 현재 노출되고 있는 App의 package name을 parameter로 입력한 package name과 비교하여 일치 불일치 상태를 return 함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| name | String | 확인해야 할 App package 이름 입력(com.kakao.talk) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| status | Int | **실행결과가 정상 Case**  **1** (일치), **2**(불일치), **0**(비정상 동작) 이들 중 하나가 리턴  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 Foreground Activity package name 확인  status = cmd.cmd\_status\_currentActivity(name=’com.kakao.talk’)  cmd.set\_print(status)  # 현재 Foreground Activity package name이 com.kakao.talk로 일치  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1 |

### cmd\_status\_killApp

|  |
| --- |
| **Function Desc** |
| Parameter로 전달 받은 App의 package name을 기준으로 강제 종료 실행 후 상태를 return 함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| name | String | 종료해야 할 App package 이름 입력(com.kakao.talk) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| status | Int | **실행결과가 정상 Case**  **1** (정상 종료), **2**(이미 종료됨), **0**(비정상 동작) 이들 중 하나가 리턴  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # parameter package name App 종료  status = cmd.cmd\_status\_killApp(name=’com.kakao.talk’)  cmd.set\_print(status)  # com.kakao.talk App 정상 종료  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1 |

### cmd\_status\_click

|  |
| --- |
| **Function Desc** |
| Parameter로 전달받은 X/Y absolute Position 또는 divided\_window로 나누어진 구역을 기준으로 click event를 실행 후 상태를 return 함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| width | Float/Int | X 좌표값으로 입력 값은 아래와 같음  **pos\_type이 abs Case**  X의 절대 좌표값(230.54)  **Pos\_type이 rate Case**  X의 선택한 구역 중간 값 |
| height | Float/Int | Y 좌표값으로 입력 값은 아래와 같음  **pos\_type이 abs Case**  Y의 절대 좌표값(230.54)  **Pos\_type이 rate Case**  Y의 선택한 구역 중간 값 |
| pos\_type | String | width / height의 입력 타입(‘abs’ 또는 ‘rate’) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| status | Int | **실행결과가 정상 Case**  **1** (정상 동작), **0**(비정상 동작) 이들 중 하나가 리턴  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 abs type으로 절대좌표에 click event 실행  status = cmd.cmd\_status\_click(width=273.14, height=1150.12,  pos\_type=’abs’)  cmd.set\_print(status)  # 정상 실행  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1  # 현재 rate type으로 좌표에 click event 실행(divided count 가 10인 경우)  status = cmd.cmd\_status\_click(width=0, height=2,  pos\_type=’rate’)  cmd.set\_print(status)  # 정상 실행  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1  # Element으로 절대좌표에 click event 실행  cmd.save\_dump\_xml()  # Target Element get pos   list\_pos = cmd.get\_pos\_elements(attr=’content-desc’, name=’공유하기’)  status = cmd.cmd\_status\_click(width=list\_pos[0][0],  height=list\_pos[0][1], pos\_type=’abs’)  cmd.set\_print(status)  # 정상 실행  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1 |

### cmd\_status\_swipe

|  |
| --- |
| **Function Desc** |
| Parameter로 전달받은 X/Y absolute Position 또는 divided\_window로 나누어진 구역을 기준으로 Touch swipe event를 실행 후 상태를 return 함 |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| s\_width | Float/Int | Touch start X 좌표값으로 입력 값은 아래와 같음  **pos\_type이 abs Case**  X의 절대 좌표값(230.54)  **Pos\_type이 rate Case**  X의 선택한 구역 중간 값 |
| s\_height | Float/Int | Touch Start Y 좌표값으로 입력 값은 아래와 같음  **pos\_type이 abs Case**  Y의 절대 좌표값(230.54)  **Pos\_type이 rate Case**  Y의 선택한 구역 중간 값 |
| e\_width | Float/Int | Touch End X 좌표값으로 입력 값은 아래와 같음  **pos\_type이 abs Case**  X의 절대 좌표값(230.54)  **Pos\_type이 rate Case**  X의 선택한 구역 중간 값 |
| e\_height | Float/Int | Touch End Y 좌표값으로 입력 값은 아래와 같음  **pos\_type이 abs Case**  Y의 절대 좌표값(230.54)  **Pos\_type이 rate Case**  Y의 선택한 구역 중간 값 |
| duration | Int | Swipe event 동작 시간 (ms) |
| pos\_type | String | width / height의 입력 타입(‘abs’ 또는 ‘rate’) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| status | Int | **실행결과가 정상 Case**  **1** (정상 동작), **0**(비정상 동작) 이들 중 하나가 리턴  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 abs type으로 절대좌표에 swipe event 실행  status = cmd.cmd\_status\_swipe(s\_width=273.14, S\_height=1150.12,  s\_width=273.14, S\_height=1150.12, duration=300, pos\_type=’abs’)  cmd.set\_print(status)  # 정상 실행  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1  # 현재 rate type으로 좌표에 swipe event 실행(divided count 가 10인 경우)  status = cmd.cmd\_status\_swipe s\_width=2, S\_height=6,  s\_width=2, S\_height=2, duration=300, pos\_type=’rate)  cmd.set\_print(status)  # 정상 실행  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1  # Element으로 절대좌표에 swipe event 실행  cmd.save\_dump\_xml()  # Target Element get pos   s\_list\_pos = cmd.get\_pos\_elements(attr=’content-desc’, name=’공유하기’)  e\_list\_pos = cmd.get\_pos\_elements(attr=’text’, name=’끝’)  status = cmd.cmd\_status\_swipe(s\_width=s\_list\_pos[0][0],  s\_height=s\_list\_pos[0][1], e\_width=e\_list\_pos[0][0],  e\_height=e\_list\_pos[0][1], duration=300, pos\_type=’abs’)  cmd.set\_print(status)  # 정상 실행  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1 |

### cmd\_status\_sendKey

|  |
| --- |
| **Function Desc** |
| Parameter로 전달한 message를 ADB cmd를 통해 입력되도록 실행 후 상태 return |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| message | String | 입력해야 할 Text 입력 |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| status | Int | **실행결과가 정상 Case**  **1** (일치), **0**(비정상 동작) 이들 중 하나가 리턴  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # 현재 abs type으로 android Input type절대좌표에 click event 실행 cmd.cmd\_status\_click(width=273.14, height=1150.12,  pos\_type=’abs’)  status = cmd.cmd\_status\_sendKey(message=’테스트입니다.’)  # sendKey event status print  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1 |

### cmd\_status\_install

|  |
| --- |
| **Function Desc** |
| Parameter로 전달한 apk path로 Android device에 App을 install 실행 후 상태를 return |

|  |  |  |
| --- | --- | --- |
| **Parameter** | | |
| **Name** | **Type** | **Description** |
| path | String | Install 대상 apk 파일 위치(Local PC) |

|  |  |  |
| --- | --- | --- |
| **Return** | | |
| **Name** | **Type** | **Description** |
| status | Int | **실행결과가 정상 Case**  **1** (일치), **0**(비정상 동작), **2**(설치 실패) 이들 중 하나가 리턴  **실행결과가 비정상 Case**  None |

|  |
| --- |
| **Exam** |
| if \_\_name\_\_ == "\_\_main\_\_":   serial\_num = '1c25c664460c7ece'  window\_count = 20  cmd = CMDS(serial\_num, 20)  # 필수 setup method 반드시 호출  cmd.setup\_test()  # test apk 설치 실행  status = cmd.cmd\_status\_install(path=’c://program/test.apk’)  # sendKey event status print  [SM-G960N(1c25c664460c7ece)]2021-07-19 11:28:15:  1 |

# How to Executes