

Factsheet

Agent SDKs – Android SDK

Android SDK

Android SDK

The SamKnows Android SDK allows the suite of SamKnows measurements to be incorporated into an Android app. The SDK is delivered as a series of libraries, which provide the measurements functionality, environmental data capture, and data reporting. The library is primarily written in Kotlin, while maintaining interoperability with Java 6+.

Requirements

The Android SDK supports Kit Kat (API level 19) through to Oreo (API level 27) and the following CPU architectures only:

- x86
- x86_64
- armeabi-v7a
- arm64-v8a

Please note that armeabi and 32/64-bit mips are officially deprecated.

Available tests

The measurements included in the SamKnows Android SDK are Download/Upload throughput, Latency, Jitter, Packet Loss, YouTube Video Streaming and Web Page Loading Times.

Download and upload speed

These tests measure the download or upload HTTP throughput in bits per second, they are located in the HttpModule. There are static initialisers provided to get instances of the DownloadTestRunner or the UploadTestRunner and returns a HttpResult object.

Latency, jitter and packet loss

This is a single test that provides three separate metrics upon completion: latency, jitter and packet loss. The test is contained in the PerformanceModule which provides a static initialiser to get an instance of thePerformanceTestRunner and returns a LatencyResult object.

YouTube

The YouTube test measures the streaming performance of the user's internet connection with a developer-definable YouTube video URL. This test is contained within the YouTubeModule which provides a static initialiser to get an instance of the YouTubeTestRunner and returns a YouTubeResult object.

WebGet

The webget test measures the performance of the user's internet connection of loading developer definable web page. This test is contained within the WebGetModule which provides a static initialiser to get an instance of the WebGetRunner and returns a BrowsingResult object.

Obtaining the SDK

The SDK is available to licenced users only and is delivered via a secure SamKnows FTP server. If you are already licenced to use the SDK, contact your SamKnows account manager to obtain the credentials to download it.

A first look at the SDK

The SDK has been designed to be modular and plugin based, and is split into five separate AAR (Android Archive) files as below: (Please see page 6)

Measurement servers

The measurement clients in the SamKnows Android SDK carry out their measurements against dedicated measurement servers provided by SamKnows. Some examples of these are included in the sample app. When using the SDK, you will need to configure a list of measurement servers for it to use.

A list of dedicated measurement servers appropriate for your geography and use-case can be obtained from SamKnows. Alternatively, you may wish to host your own measurement servers. In this instance, SamKnows can provide a software package suitable for installation on RHEL/CentOS 6.x or 7.x servers. For more information about measurement servers please contact your SamKnows account manager.

Data collection

The Android SDK reports its measurements to one or more data collection endpoints. Data is transmitted to the collection endpoint in JSON format over a HTTPS connection. The data collected is limited to the measurement results themselves, environmental information (e.g. Wi-Fi and cellular connectivity status) and a persistent unique identifier for the handset. No personal information is collected. The SDK is configured to transmit measurement results to the SamKnows data collection endpoint exclusively.

Environmental data

The following environmental information is collected for each test run:
(Please see pages 4-5)

Data sample

Below is a sample of the data collected and submitted for a test sequence that only includes the WebGet test:

```
{
  "agentId": "<SUPPLIED_BY_SAMKNOWS_SERVERS>",
  "data": {
    "device_environment": {
```

```
      "carrier_name": "EE",
      "iso_country_code": "GB",
      "manufacturer": "Samsung",
      "mobile_country_code": "+44",
      "mobile_network_code": "GB",
      "model": "SM-G965F",
      "operating_system_version": "Android
8.0.0"
    },
    "metadata": {
      "app_version": "1.0"
    },
    "tests": {
      "www": {
        "bytes_transferred": 2900556,
        "duration": 1412995,
        "environment": {
          "cross_traffic": {
            "rx_avg_bps": 8302,
            "rx_peak_bps": 9352,
            "tx_avg_bps": 2928,
            "tx_peak_bps": 5051
          },
          "dhcp": {
            "dns1": "10.0.0.2",
            "dns2": "0.0.0.0",
            "gateway": "10.0.0.1",
            "ip_address": "10.0.1.104",
            "lease_duration": 7200,
            "netmask": "0.0.0.0",
            "server_address": "10.0.0.2"
          },
          "location": {
            "lat": -1.0,
            "lon": -1.0
          },
          "memory": {
            "app_used": 6,
            "device_free": 2006,
            "device_used": 3189
          },
          "network": {
            "is_available": true,
            "is_connected": true,
            "is_connected_or_connecting": true,
            "is_failover": false,
            "is_roaming": false,
            "state": "connected",
            "subtype": 0,
            "subtype_name": "",
            "is_connected_or_connecting": true,
            "is_failover": false,
            "is_roaming": false,
            "state": "connected",
            "subtype": 0,
            "subtype_name": ""
```

```

        "type":"mobile"
    },
    "telephony":{
        "cellular_technology":"4g",
        "connection_type":"wi-fi"
    },
    "wifi":{
        "frequency":2437,
        "hidden_ssid":false,
        "link_speed":144,
        "network_id":2,
        "rssi":-50,
        "ssid":"sk",
        "supplicant_state":"completed"
    }
},
"failures":0,
"local_datetime":"2018-06-
12T10:59:51+01:00",
"successes":1,
"target":"139.162.186.25",
"time_to_connect":23198,
"time_to_first_byte":202145,
"time_to_page_load":203583,
"utc_datetime":"2018-06-
12T09:59:51Z"
}
},
"version":1
}

```

Configuring the SDK

There are two methods to include the AAR artefacts in your Android Studio project

- Either copy/paste the AAR files directly into the project libs folder. Add the following code fragment to your app level build.gradle file:
- OR use the Android Studio import wizard:
- The SDK relies on some third party libraries which are deliberately not included in the compiled binaries to keep file size as small as possible, but

they do need to be added to your project dependencies list specifically in the app build.gradle file.

- Hit the gradle sync button on the Android Studio toolbar, this is 'explode' the AAR artefacts into your project allowing you to access SDK internal functions and use code completion in the editor.

Getting started

- The first step is to initialise the SamKnows SDK, this is done inside a custom Application class, if you haven't already created one you will need to do so first.
- Depending on the measurement plugins you want to use and have included in your project you can setup a test schema and execute it, as shown below:

Additional features

The Core SDK AAR contains additional features around loading a list of servers from a specified URL or aservers.json application assets file and then performing best target server analysis using ICMP pings (if the ICMP process is available on the Android device, if not it falls back to a TCP ping analysis). If you wish to use these features you can refer to the sample code in the demo applications provided in the repository.

Environmental data

Category	Field	Description
	agent_id	Provided by the SamKnows servers when a user device is first registered.
	version	
Device	carrier_name	Mobile carrier, e.g. EE, Vodafone etc
	iso_country_code	ISO ALPHA-2 Country Code, e.g. GB, US etc
	manufacturer	Device manufacturer, e.g. Samsung, HTC etc
	mobile_country_code	International dialling code, e.g. +44 for the UK
	mobile_network_code	Sim card network code, e.g. GB
	model	Device model, e.g. SM-G965F
	operating_system_version	Android version, e.g. Android 8.0.0
Metadata	app_version	Your application version, e.g. 1.0.0
Environment	rx_avg_bps	Cross traffic average received in bytes per second.
	rx_peak_bps	Cross traffic peak received in bytes per second.
	tx_avg_bps	Cross traffic averaged transmitted in bytes per second.
	tx_peak_bps	Cross traffic peak transmitted in bytes per second.
	dns1	DHCP DNS server 1
	dns2	DHCP DNS server 2
	gateway	DCHP Gateway
	ip_address	Device IP Address
	lease_duration	DHCP lease duration
	netmask	DHCP net mask
	server_address	DHCP Server IP

Category	Field	Description
	lat	Location - Latitude
	lon	Location - Longitude
	app_used	Memory - Used by your application.
	device_free	Memory - Available on device
	device_used	Memory - In use by other apps/operating system.
	is_available	Network - Is available?
	is_connected	Network - Is connected?
	is_connected_or_connecting	Network - Is connected or trying to connect?
	is_failover	Network - Has network connection switched due to failover?
	is_roaming	Network - Is device roaming?
	state	Network state, e.g. connected.
	subtype	Network subtype, a numerical value.
	subtype_name	Network subtype name, not generally used.
	type	Network type, e.g. mobile, wi-fi
	cellular_technology	Mobile network type, e.g. 4G
	connection_type	Network connection type, e.g. wi-fi, mobile, simless
	frequency	Wi-Fi connection frequency.
	hidden_ssid	Wi-Fi is the SSID hidden?
	link_speed	Wi-Fi link speed
	network_id	Wi-Fi network ID
	rssi	Wi-Fi network RSSI
	ssid	Wi-Fi network SSID

Category	Field	Description
	supplicant_state	Wi-Fi network supplicant state, e.g. completed.

A first look at the SDK

Name	Required	AAR Artefact	Measurement Metrics
Core	Y	sdk-core.aar	None
HTTP	N	sdk-http-plugin.aar	Download/Upload throughput
Performance	N	sdk-performance-plugin.aar	Latency, Jitter and Packet Loss
YouTube	N	sdk-youtube-plugin.aar	YouTube video streaming performance
WebGet	N	sdk-webget-plugin.aar	Web page loading performance