# **Chartbeat Android SDK Spec**

# **Chartbeat Pinger Basics**

Chartbeat pinger code loads at startup, monitors the user's behaviour and makes an image request to ping.chartbeat.net every X seconds. The image requests contain query parameters that provide data back to Chartbeat servers.

The overall design philosophy is:

- Minimize overall system impact (app load, battery, network, etc.)
- Simple installation and setup
- As broadly compatible as is practical (Target Android 2.2+)
- Adhere to the same overall design as the iOS SDK in order to make cross-platform installation easier to understand. The basic interface is defined in the <u>CBTracker.h</u>

## **SDK Installation and Use**

## Setup

Ideally developers can use the SDK with minimal effort, optionally including additional data.

#### Required for setup:

- Account ID Used for identifying the Chartbeat account
- View ID An identifier for a specific "page" or "view" to be tracked

#### Optional

- Host Override the dashboard domain, defaults to app package name (see "h" key)
- Title Override the view name, default to the name of the view (see "i" key)
- Interval override the default interval of the ping
- Sections Chartbeat Publishing section tags
- Authors Chartbeat Publishing author tags
- Referrer Where the App was opened from (see "r" key)
- Geo Longitude and Latitude (see "Ig" and "It")

The creation and use of CBTracker should be based on how it works in the iOS SDK. This is essentially calling startTrackerWithAccountID to initialize the pinger and calling trackView for every new view after that.

# **Ping Frequency**

The pinger should make a request immediately as soon as a trackView is called. After that, it should make a request every 15 seconds as long as the view is engaged (See definition of the

"E" key). If the view is unengaged for 15 seconds the the interval should switch to 30 seconds.

All pinging should be halted when the browser moves to the background. The "j" key should be reset to 15 (or whatever the code set it to) once the app moves back into the foreground

If the app is actively moved to the background by the user the session should be considered closed. The next ping should generate a new "t" key and send the prior "t" as the "D" key.

If the app is passively moved to the background (eg. sleeps) the pinger should immediately send a ping with a decay ("j" key) of 90 seconds, and subsequently ping every 60 seconds if the network connection is already active. If the network connection is not active it should not ping until the connection is active again.

## **Ping UserAgent**

The User Agent for the ping should match the browser of the Android version.

In addition, we should add "/App" to the end of the User Agent so we can distinguish App and Browser traffic.

#### **Return Codes**

Since the API consists of simple image requests we use HTTP response code to indicate issues that the pinger can act on.

500 - Indicates the ping has incomplete data. A new ping with all available keys (except "D"), should be sent immediately

503 - Indicates the client is sending more pings then they are contracted for. The interval should increase by 2x everytime this happens, with a maximum value of 5 mins. Once a 200 is received the interval should be reset to 15 (or whatever the code set it to).

All other return codes are processes normally, including other error codes, which should just retry at the normal ping interval.

# **Key Reference**

Here's a full reference of all the query parameters ("keys") that are sent with the image request.

**NOTE:** All keys passed in from the app (h, p, t, g0-2, etc.) should be URI encoded.

## Required Keys

Required keys are sent with every ping. Keys that are not required should only be sent when they change, or when a 500 error is returned from the ping request.

#### h: Host

The dashboard this visitor is reporting to. Defaults to the App Package but can be overridden during the pinger init process

#### d: Real Domain

Independent of the Host (h key), this is always the App Package

## p: Path

The view id for tracking that view

#### t: Token

Unique identifier for page-session (e.g. each call to trackView). Must be a 32 char string with at least 80 bits of entropy.

#### u: User ID

Unique identifier for a user, who could have multiple page-sessions. Must be stored Must be a 32 char string with at least 80 bits of entropy. It should be consistent with the <a href="Manual Content of the International Content of the Internationa

## g: Account ID

As set in the pinger initialization.

#### r: External referrer

This would ideally be URL or App PAckage of the page or app which opened the target app. If that's not available if should be empty (but still sent). This is only sent on the first trackView call for the session.

or

#### v: Internal referrer

If the view the user came from is considered within the same App, that "p" key should be supplied here.

## **Additional Keys**

## g[0-9]: "Group" keys for account specific meta data

- "g0" key should be a comma-delimited list of sections.
- "g1" key should be a comma-delimited list of authors.
- "g2" key should be a comma-delimited list of zones.

### i: View Title

This is what will show up in the dashboard

#### n: New User

0 = Returning, 1 = New

### f: Frequency with which a user visits

A hex string of 5 characters.

- 1st hex char: Number of days ago of real data. Ranges from 0 to 15 ("f"). If this is 15, all of the following bits are real data. If this is 7, only the 4th and 5th hex chars contain real data bits, etc.
- 2nd hex char: bits representing visits 15, 14, 13 and 12 days ago.
- 3rd hex char: bits representing visits 11, 10, 9, and 8 days ago.
- 4th hex char: bits representing visits 7, 6, 5, and 4 days ago.
- 5th hex char: bits representing visits 3, 2, 1, and 0 days ago.

# c: Time On View in minutes (with 1 decimal)

#### W: Width of the device

### w: Window Height

## j: Decay

More specifically, how long in seconds memoryfly should expect to wait until the next ping. Typically this will be 2x the current interval.

#### D: Force decay

The "t" key from the previous session, if the "j" for that session has not passed since the last ping

#### b: Page Load Time

Time it took for the vire to load

#### lg:longitude

If the app is already using Geo, pass this automatically. Otherwise use override.

#### It:latitude

If the app is already using Geo, pass this automatically. Otherwise use override.

#### V: Version

The current SDK version being used

## **Engagement Keys**

A view is considered engaged if the app is currently active and the user has interacted with the screen in the last X seconds. Engagement is updated every second.

The engagement window (X) should load dynamically from a config file when the app loads. The url for the config file will be: <a href="http://static.chartbeat.com/static/data/config.json">http://static.chartbeat.com/static/data/config.json</a>. The format will be:

```
{
"engagement_window": 5
}
```

## E: Engaged seconds

Engaged seconds spent on the page so far, as calculated above

## R: Reading

Set to 1 if the user is engaged in the last ping interval but did not engage in any keystrokes, otherwise 0

# W: Writing

Set to 1 if the user is engaged in the last ping interval and engaged in at least 1 "keystrokes, otherwise 0

#### I:Idle

Set to 1 if the user was not engaged, otherwise 0

# **Position Keys**

These keys assume the view scrolls. If the customer is creating shared paths ("p" keys) with another app or website they should only send it for one.

x: Scroll Position Top

y: Scroll Window Height

w: Height of the currently viewable window

o: Width of the document fully rendered

m: Max scroll depth during the session