

Event API

Authentication:

We authenticate every request sent to the Tenjin server using the Tenjin API Key as the username in basic auth or as a parameter `api_key`.

POST or GET URL:

<https://track.tenjin.io/v0/event>

Parameters:

Parameter	Type	Description
advertising_id*	String	The raw advertising ID of the device. Required for Android. For iOS, it can be empty or all zeros.
developer_device_id*	String	The identifier for vendor (iOS only)
bundle_id*	String	Bundle ID of the app.
platform*	String	ios, android, windows, amazon.
limit_ad_tracking	String	1 for yes, 0 for no.
os_version*	String	For Android: String os_version = String.valueOf(Build.VERSION.SDK_INT) For iOS: [[UIDevice currentDevice] systemVersion]
app_version	Float, Decimal	The version of your app For Android: String app_version = context.getPackageManager().getPackageInfo(context.getPackageName(), 0).versionName + "." + context.getPackageManager().getPackageInfo(context.getPackageName(), 0).versionCode For iOS: [[[NSBundle mainBundle] infoDictionary] objectForKey:@"CFBundleShortVersionString"]
sdk_version*	String	"server"
country*	String	The ISO locale country codes standard (see Apple documentation for example).
ip_address	String	The User's IP Address. This is used to lookup country.

iad	Integer (required to track iAd)	1=claimed by iAd, 0=not claimed by iAd (only used if you track iAd campaigns)
iad_impression_ts	String (required to track iAd)	Timestamp if claimed by iAd
referrer	String (required to track Adwords)	Collected for Android installs tracked
os_version_release	String (required to track Adwords)	For Android: String os_version_release = String.valueOf(Build.VERSION.RELEASE) For iOS: [[UIDevice currentDevice] systemVersion]
build_id	String (required to track AdWords)	For Android: String build = String.valueOf(Build.ID) For iOS: @import Darwin.sys.sysctl; NSString *build(void) { size_t bufferSize = 64; NSMutableData *buffer = [[NSMutableData alloc] initWithLength:bufferSize]; int status = sysctlbyname("kern.osversion", buffer.mutableBytes, &bufferSize, NULL, 0); if (status != 0) { return nil; } return [[NSString alloc] initWithCString:buffer.mutableBytes encoding:NSUTF8StringEncoding]; }
locale	String (required to track AdWords)	For Android: String locale = Locale.getDefault().toString() For iOS: [[NSLocale currentLocale] localeIdentifier]
device_model	String (required to track AdWords)	For Android: String device = Build.MODEL;

		<p>For iOS:</p> <pre>@import Darwin.sys.sysctl; NSString *device(void) { size_t bufferSize = 64; NSMutableData *buffer = [[NSMutableData alloc] initWithLength:bufferSize]; int status = sysctlbyname("hw.machine", buffer.mutableBytes, &bufferSize, NULL, 0); if (status != 0) { return nil; } return [[NSString alloc] initWithCString:buffer.mutableBytes encoding:NSUTF8StringEncoding]; }</pre>
--	--	--

*: required

Purchase API

Authentication:

We authenticate every request sent to the Tenjin server using the Tenjin API Key as the username in basic auth.

POST or GET URL:

<https://track.tenjin.io/v0/purchase>

Parameters:

Parameter	Type	Description
advertising_id*	String	The advertising ID of the device. Required for Android. For iOS, it can be empty or all zeros.
developer_device_id*	String	The identifier for vendor (iOS only)
bundle_id*	String	Bundle ID of the app.
platform*	String	ios, android, windows, amazon.
limit_ad_tracking	String	1 for yes, 0 for no.
os_version	String	For Android: <code>String.valueOf(Build.VERSION.SDK_INT)</code> For iOS: <code>[[UIDevice currentDevice] systemVersion]</code>
app_version	Float, Decimal	The version of your app For Android: <code>String app_version = context.getPackageManager().getPackageInfo(con text.getPackageName(), 0).versionName + "." + context.getPackageManager().getPackageInfo(con text.getPackageName(), 0).versionCode)</code> For iOS: <code>[[[NSBundle mainBundle] infoDictionary] objectForKey:@"CFBundleShortVersionString"]</code>
sdk_version*	String	"server"
ip_address	String	The User's IP Address. This is used to lookup country.

Confidential

postcut	Bool	1 for sending purchases after the platform cut
product_id*	String	Unique name for the product purchased.
price*	Float, Decimal	The unit price of the item purchased in the currency standard unit (ex. USD is reported in dollars).
quantity*	Integer	The number of units purchased.
country*	String	The ISO locale country codes standard (see Apple documentation for example).
currency*	String	The ISO currency codes used by Apple and Google (see Apple documentation for example).
os_version_release	String (required to track Adwords)	For Android: <pre>String os_version_release = String.valueOf(Build.VERSION.RELEASE)</pre> For iOS: <pre>[[UIDevice currentDevice] systemVersion]</pre>
build_id	String (required to track AdWords)	For Android: <pre>String build = String.valueOf(Build.ID)</pre> For iOS: <pre>@import Darwin.sys.sysctl; NSString *build(void) { size_t bufferSize = 64; NSMutableData *buffer = [[NSMutableData alloc] initWithLength:bufferSize]; int status = sysctlbyname("kern.osversion", buffer.mutableBytes, &bufferSize, NULL, 0); if (status != 0) { return nil; } return [[NSString alloc] initWithCString:buffer.mutableBytes encoding:NSUTF8StringEncoding]; }</pre>
locale	String (required to track AdWords)	For Android: <pre>String locale = Locale.getDefault().toString()</pre> For iOS: <pre>[[NSLocale currentLocale] localeIdentifier]</pre>

device_model	String (required to track AdWords)	<p>For Android:</p> <pre>String device = Build.MODEL;</pre> <p>For iOS:</p> <pre>@import Darwin.sys.sysctl; NSString *device(void) { size_t bufferSize = 64; NSMutableData *buffer = [[NSMutableData alloc] initWithLength:bufferSize]; int status = sysctlbyname("hw.machine", buffer.mutableBytes, &bufferSize, NULL, 0); if (status != 0) { return nil; } return [[NSString alloc] initWithCString:buffer.mutableBytes encoding:NSUTF8StringEncoding]; }</pre>
transaction_id (required for receipt validation) (required for Apple only)	String	<p>The Apple IAP transaction ID. See transactionIdentifier documentation.</p> <p>For example:</p> <pre>120000449972561</pre>
receipt (required for receipt validation and dashboard reporting) (required for both Apple and Android)	String (*should be encoded)	<p>The IAP receipt.</p> <p>Apple:</p> <p>Apple base64 encoded ASN.1 receipt. See documentation.</p> <p>Android:</p> <p>Android Receipt JSON data. The INAPP_PURCHASE_DATA response code from getBuyIntent(). See documentation. For example:</p> <pre>{ "orderId": "12999763169054705758.1371079406387615", "packageName": "com.tenjin.android.testapp", "productId": "exampleSku", "purchaseTime": 1345678900000, "purchaseState": 0, "developerPayload": "bGoa+V7g/yqDXvKRqq+JTfn4uQZbPiQJo4pf9RzJ", "purchaseToken": "rojeslcdyyiiapnqcynkjyyjh" }</pre>

signature (required for receipt validation and dashboard reporting) (required for Android only)	String (*should be encoded)	The Android verification signature. The INAPP_DATA_SIGNATURE response code from getBuyIntent(). See documentation. For example: OTCzm6DczVc46awg3qcG6/UoUJ/1pobHf00KmA0zUmAK8Wgh+ngnhFNEy4uh/x25 6BEgLKFo9KgyhpAZEaBz1HOUamnCNLSJ58KxKzup/Dddig4jd4JQn9lu8ObaLHKc Y8PPbn+qod401P8xSTKCinP9LQ93/g2koaWR2/8NzzLe1ydyiRSBIdEnLmBxLrew +Fdfgic8336AyhPm7tdWkQbf7zriqeUF1D17pXtb4ht3plp2wnrTFpWLJaxO9nSH bc+vyQFe1PNlmrFMpiEtQYPgwwfDpCeQvrGkbLKMj48gOhfCvo3oWcp5OiZRzNsS 1EAMBbhaaWkrsXFu5igAlg==
---	--------------------------------	--

*: required

Custom Event API

Authentication:

We authenticate every request sent to the Tenjin server using the Tenjin API Key as the username in basic auth.

POST or GET URL:

<https://track.tenjin.io/v0/event>

Parameters:

Parameter	Type	Description
advertising_id*	String	The advertising ID of the device. Required for Android. For iOS, it can be empty or all zeros.
developer_device_id*	String	The identifier for vendor (iOS only)
bundle_id*	String	Bundle ID of the app.
platform*	String	ios, android, windows, amazon.
limit_ad_tracking	String	1 for yes, 0 for no.
os_version	String	For Android: <code>String.valueOf(Build.VERSION.SDK_INT)</code> For iOS: <code>[[[UIDevice currentDevice] systemVersion]</code>
app_version	Float, Decimal	The version of your app For Android: <code>String app_version = context.getPackageManager().getPackageInfo(con text.getPackageName(), 0).versionName + "." + context.getPackageManager().getPackageInfo(con text.getPackageName(), 0).versionCode)</code> For iOS: <code>[[[NSBundle mainBundle] infoDictionary] objectForKey:@"CFBundleShortVersionString"]</code>
sdk_version*	String	“server”
ip_address	String	The User’s IP Address
event*	String	This is the name of the event - event name (e.g. “tutorial_complete”, “registration_complete”, or

Confidential

		"level_1_complete")
value*	Nil, Integer	Value tied to the event name (above). This can be used for calculating values related to an event name. (e.g want to calculate the number of virtual currency for a specific event).
os_version_release	String (required to track AdWords)	For Android: String os_version_release = String.valueOf(Build.VERSION.RELEASE) For iOS: [[UIDevice currentDevice] systemVersion]
build_id	String (required to track AdWords)	For Android: String build = String.valueOf(Build.ID) For iOS: @import Darwin.sys.sysctl; NSString *build(void) { size_t bufferSize = 64; NSMutableData *buffer = [[NSMutableData alloc] initWithLength:bufferSize]; int status = sysctlbyname("kern.osversion", buffer.mutableBytes, &bufferSize, NULL, 0); if (status != 0) { return nil; } return [[NSString alloc] initWithCString:buffer.mutableBytes encoding:NSUTF8StringEncoding]; }
locale	String (required to track AdWords)	For Android: String locale = Locale.getDefault().toString() For iOS: [[NSLocale currentLocale] localeIdentifier]
device_model	String (required to track AdWords)	For Android: String device = Build.MODEL; For iOS: @import Darwin.sys.sysctl; NSString *device(void) { size_t bufferSize = 64; }

		<pre>NSMutableData *buffer = [[NSMutableData alloc] initWithLength:bufferSize]; int status = sysctlbyname("hw.machine", buffer.mutableBytes, &bufferSize, NULL, 0); if (status != 0) { return nil; } return [[NSString alloc] initWithCString:buffer.mutableBytes encoding:NSUTF8StringEncoding]; }</pre>
--	--	--

*: required