
PYGA Documentation

Release 2.0

Arun K R

Oct 02, 2018

Contents

1	Indices and tables	9
	Python Module Index	11

Contents:

Contents:

class `pyga.entities.Campaign` (*typ*)

A representation of Campaign

Properties: `_type` – See `TYPE_*` constants, will be mapped to “__utmz” parameter. `creation_time` – Time of the creation of this campaign, will be mapped to “__utmz” parameter. `response_count` – Response Count, will be mapped to “__utmz” parameter.

Is also used to determine whether the campaign is new or repeated, which will be mapped to “utmcn” and “utmcr” parameters.

id – Campaign ID, a.k.a. “utm_id” query parameter for ga.js Will be mapped to “__utmz” parameter.

source – Source, a.k.a. “utm_source” query parameter for ga.js. Will be mapped to “utmcsr” key in “__utmz” parameter.

g_click_id – Google AdWords Click ID, a.k.a. “gclid” query parameter for ga.js. Will be mapped to “utmglcid” key in “__utmz” parameter.

d_click_id – DoubleClick (?) Click ID. Will be mapped to “utmdclid” key in “__utmz” parameter. **name – Name, a.k.a. “utm_campaign” query parameter for ga.js.**

Will be mapped to “utmccn” key in “__utmz” parameter.

medium – Medium, a.k.a. “utm_medium” query parameter for ga.js. Will be mapped to “utmcmd” key in “__utmz” parameter.

term – Terms/Keywords, a.k.a. “utm_term” query parameter for ga.js. Will be mapped to “utmctr” key in “__utmz” parameter.

content – Ad Content Description, a.k.a. “utm_content” query parameter for ga.js. Will be mapped to “utmctt” key in “__utmz” parameter.

class `pyga.entities.CustomVariable` (*index=None, name=None, value=None, scope=3*)

Represent a Custom Variable

Properties: `index` – Is the slot, you have 5 slots `name` – Name given to custom variable `value` – Value for the variable `scope` – Scope can be any one of 1, 2 or 3.

WATCH OUT: It’s a known issue that GA will not decode URL-encoded characters in custom variable names and values properly, so spaces will show up as “%20” in the interface etc. (applicable to name & value) <http://www.google.com/support/forum/p/Google%20Analytics/thread?tid=2cdb3ec0be32e078>

validate ()

According to the GA documentation, there is a limit to the combined size of name and value of 64 bytes after URL encoding, see <http://code.google.com/apis/analytics/docs/tracking/gaTrackingCustomVariables.html#varTypes> and http://xahlee.org/js/google_analytics_tracker_2010-07-01_expanded.js line 563 This limit was increased to 128 bytes BEFORE encoding with the 2012-01 release of ga.js however, see http://code.google.com/apis/analytics/community/gajs_changelog.html

class `pyga.entities.Event` (*category=None, action=None, label=None, value=None, noninteraction=False*)

Represents an Event <https://developers.google.com/analytics/devguides/collection/gajs/eventTrackerGuide>

Properties: `category` – The general event category `action` – The action for the event `label` – An optional descriptor for the event `value` – An optional value associated with the event. You can see your

event values in the Overview, Categories, and Actions reports, where they are listed by event or aggregated across events, depending upon your report view.

noninteraction – By default, event hits will impact a visitor’s bounce rate. By setting this parameter to true, this event hit will not be used in bounce rate calculations. (default False)

class `pyga.entities.Item`

Represents an Item in Transaction

Properties: `order_id` – Order ID, will be mapped to “utmtid” parameter `sku` – Product Code. This is the sku code for a given product, will be mapped to “utmipc” parameter `name` – Product Name, will be mapped to “utmipn” parameter `variation` – Variations on an item, will be mapped to “utmiva” parameter `price` – Unit Price. Value is set to numbers only, will be mapped to “utmipr” parameter `quantity` – Unit Quantity, will be mapped to “utmigt” parameter

class `pyga.entities.Page(path)`

Contains all parameters needed for tracking a page

Properties: `path` – Page request URI, will be mapped to “utmp” parameter `title` – Page title, will be mapped to “utmdt” parameter `charset` – Charset encoding, will be mapped to “utmcs” parameter `referrer` – Referrer URL, will be mapped to “utmr” parameter `load_time` – Page load time in milliseconds, will be encoded into “utme” parameter.

class `pyga.entities.Session`

You should serialize this object and store it in the user session to keep it persistent between requests (similar to the “__utmb” cookie of the GA Javascript client).

Properties: `session_id` – A unique per-session ID, will be mapped to “utmhid” parameter `track_count` – The amount of pageviews that were tracked within this session so far,

will be part of the “__utmb” cookie parameter. Will get incremented automatically upon each request
`start_time` – Timestamp of the start of this new session, will be part of the “__utmb” cookie parameter

extract_from_utmb (*utmb*)

Will extract information for the “trackCount” and “startTime” properties from the given “__utmb” cookie value.

class `pyga.entities.SocialInteraction` (*action=None, network=None, target=None*)

Properties: `action` – Required. A string representing the social action being tracked,

will be mapped to “utmsa” parameter

network – Required. A string representing the social network being tracked, will be mapped to “utmsn” parameter

`target` – Optional. A string representing the URL (or resource) which receives the action.

class `pyga.entities.Transaction`

Represents parameters for a Transaction call

Properties: `order_id` – Order ID, will be mapped to “utmtid” parameter `affiliation` – Affiliation, Will be mapped to “utmtst” parameter `total` – Total Cost, will be mapped to “utmtto” parameter `tax` – Tax Cost, will be mapped to “utmttx” parameter `shipping` – Shipping Cost, values as for unit and price, will be mapped to “utmtsp” parameter `city` – Billing City, will be mapped to “utmtci” parameter `state` – Billing Region, will be mapped to “utmttrg” parameter `country` – Billing Country, will be mapped to “utmtco” parameter `items` – @entity.Items in a transaction

add_item (*item*)

item of type `entities.Item`

class `pyga.entities.Visitor`

You should serialize this object and store it in the user database to keep it persistent for the same user permanently (similar to the “__umta” cookie of the GA Javascript client).

Properties: `unique_id` – Unique user ID, will be part of the “__utma” cookie parameter `first_visit_time` – Time of the very first visit of this user, will be part of the “__utma” cookie parameter `previous_visit_time` – Time of the previous visit of this user, will be part of the “__utma” cookie parameter `current_visit_time` – Time of the current visit of this user, will be part of the “__utma” cookie parameter `visit_count` – Amount of total visits by this user, will be part of the “__utma” cookie parameter `ip_address` – IP Address of the end user, will be mapped to “utmip” parameter and “X-Forwarded-For” request header `user_agent` – User agent string of the end user, will be mapped to “User-Agent” request header `locale` – Locale string (country part optional) will be mapped to “utmhl” parameter `flash_version` – Visitor’s Flash version, will be mapped to “utmfl” parameter `java_enabled` – Visitor’s Java support, will be mapped to “utmje” parameter `screen_colour_depth` – Visitor’s screen color depth, will be mapped to “utmcs” parameter `screen_resolution` – Visitor’s screen resolution, will be mapped to “utmsr” parameter

add_session (*session*)

Updates the “previousVisitTime”, “currentVisitTime” and “visitCount” fields based on the given session object.

extract_from_server_meta (*meta*)

Will extract information for the “ip_address”, “user_agent” and “locale” properties from the given WSGI REQUEST META variable or equivalent.

extract_from_utma (*utma*)

Will extract information for the “unique_id”, “first_visit_time”, “previous_visit_time”, “current_visit_time” and “visit_count” properties from the given “__utma” cookie value.

generate_hash ()

Generates a hashed value from user-specific properties.

generate_unique_id ()

Generates a unique user ID from the current user-specific properties.

Contents:

class `pyga.requests.Config`

Configurations for Google Analytics: Server Side

Properties: `error_severity` – How strict should errors get handled? After all,

we do just do some tracking stuff here, and errors shouldn’t break an application’s functionality in production. RECOMMENDATION: Exceptions during development, warnings in production.

queue_requests – Whether to just queue all requests on `HttpRequest.fire()` and actually send them on shutdown after all other tasks are done. This has two advantages: 1) It effectively doesn’t affect app performance 2) It can e.g. handle custom variables that were set after scheduling a request

fire_and_forget – Whether to make asynchronous requests to GA without waiting for any response (speeds up doing requests).

logging_callback – Logging callback, registered via `setLoggingCallback()`. Will be fired whenever a request gets sent out and receives the full HTTP request as the first and the full HTTP response (or null if the “fireAndForget” option or simulation mode are used) as the 2nd argument.

request_timeout – Seconds (float allowed) to wait until timeout when connecting to the Google analytics endpoint host.

endpoint – Google Analytics tracking request endpoint. Can be set to null to silently simulate (and log) requests without actually sending them.

anonymize_ip_address – Whether to anonymize IP addresses within Google Analytics by stripping the last IP address block, will be mapped to “aip” parameter.

site_speed_sample_rate – Defines a new sample set size (0-100) for Site Speed data collection. By default, a fixed 1% sampling of your site visitors make up the data pool from which the Site Speed metrics are derived.

```
class pyga.requests.EventRequest (config, tracker, visitor, session, event)
```

```
    build_parameters ()
        Marker implementation
```

```
class pyga.requests.GIFRequest (config)
```

Properties: type – Indicates the type of request, will be mapped to “utmt” parameter config – base.Config object
x_forwarded_for – user_agent – User Agent String

```
    build_parameters ()
        Marker implementation
```

```
    fire ()
        Simply delegates to send() if config option “queue_requests” is disabled else enqueues the request into Q
        object: you should call pyga.shutdown as last statement, to actually send out all queued requests.
```

```
class pyga.requests.ItemRequest (config, tracker, visitor, session, item)
```

```
    build_custom_variable_parameters (parameters)
        The GA Javascript client doesn’t send any custom variables for e-commerce requests, so we don’t either.
```

```
    build_parameters ()
        Marker implementation
```

```
    build_visitor_parameters (parameters)
        The GA Javascript client doesn’t send any visitor information for e-commerce requests, so we don’t either.
```

```
class pyga.requests.PageViewRequest (config, tracker, visitor, session, page)
```

```
    build_parameters ()
        Marker implementation
```

```
class pyga.requests.Parameters
```

This simple class is mainly meant to be a well-documented overview of all possible GA tracking parameters.

<http://code.google.com/apis/analytics/docs/tracking/gaTrackingTroubleshooting.html#gifParameters>

General Parameters: utmwv – Google Analytics client version utmac – Google Analytics account ID utmhn – Host Name utmt – Indicates the type of request, which is one of null (for page),

“event”, “tran”, “item”, “social”, “var” (deprecated) or “error” (used by ga.js for internal client error logging).

utms – Contains the amount of requests done in this session. Added in ga.js v4.9.2. utmn – Unique ID (random number) generated for each GIF request utmcc – Contains all cookie values, see below utme – Extensible Parameter, used for events and custom variables utmni – Event “non-interaction” parameter. By default, the event hit will impact a visitor’s bounce rate.

By setting this parameter to 1, this event hit will not be used in bounce rate calculations.

aip – Whether to anonymize IP addresses within Google Analytics by stripping the last IP address block, either null or 1 utmu – Used for GA-internal statistical client function usage and error tracking,

not implemented in php-ga as of now, but here for documentation completeness. <http://glucik.blogspot.com/2011/02/utmu-google-analytics-request-parameter.html>

Page Parameters: utmp – Page request URI utmdt – Page title utmcs – Charset encoding (default “-“) utmr – Referrer URL (default “-” or “0” for internal purposes)

Visitor Parameters: utmip – IP Address of the end user, found in GA for Mobile examples, but sadly seems to be ignored in normal GA use utmul – Visitor’s locale string (all lower-case, country part optional) utmfl – Visitor’s Flash version (default “-“) utmje – Visitor’s Java support, either 0 or 1 (default “-“) utmsc – Visitor’s screen color depth utmsr – Visitor’s screen resolution _utma – Visitor tracking cookie parameter.

Session Parameters: utmhid – Hit id for revenue per page tracking for AdSense, a random per-session ID _utmb – Session timeout cookie parameter. _utmc – Session tracking cookie parameter. utmipc – Product Code. This is the sku code for a given product. utmipn – Product Name utmipr – Unit Price. Value is set to numbers only. utmiqt – Unit Quantity. utmiva – Variations on an item. utmtid – Order ID. utmtst – Affiliation utmtto – Total Cost utmttx – Tax Cost utmtsp – Shipping Cost utmtci – Billing City utmtrg – Billing Region utmtco – Billing Country

Campaign Parameters: utmcn – Starts a new campaign session. Either utmcn or utmcr is present on any given request,

but never both at the same time. Changes the campaign tracking data; but does not start a new session. Either 1 or not set. Found in gaforflash but not in ga.js, so we do not use it, but it will stay here for documentation completeness.

utmcr – Indicates a repeat campaign visit. This is set when any subsequent clicks occur on the same link. Either utmcn or utmcr is present on any given request, but never both at the same time. Either 1 or not set. Found in gaforflash but not in ga.js, so we do not use it, but it will stay here for documentation completeness.

utmcid – Campaign ID, a.k.a. “utm_id” query parameter for ga.js utmcsr – Source, a.k.a. “utm_source” query parameter for ga.js utmgclid – Google AdWords Click ID, a.k.a. “gclid” query parameter for ga.js utmdclid – Not known for sure, but expected to be a DoubleClick Ad Click ID. utmcen – Name, a.k.a. “utm_campaign” query parameter for ga.js utmcmd – Medium, a.k.a. “utm_medium” query parameter for ga.js utmctr – Terms/Keywords, a.k.a. “utm_term” query parameter for ga.js utmctt – Ad Content Description, a.k.a. “utm_content” query parameter for ga.js utmcvr – Unknown so far. Found in ga.js. _utmz – Campaign tracking cookie parameter.

Social Tracking Parameters: utmsn – The network on which the action occurs utmsa – The type of action that happens utmsid – The page URL from which the action occurred.

Google Website Optimizer (GWO) parameters: _utmz – Website Optimizer cookie parameter.

Custom Variables parameters (deprecated): _utmv – Deprecated custom variables cookie parameter.

get_parameters ()

Get all gif request parameters out of the class in a dict form. Attributes starting with _ are cookie names, so we dont need them.

utmcvr = None

Campaign tracking cookie parameter.

This cookie stores the type of referral used by the visitor to reach your site, whether via a direct method, a referring link, a website search, or a campaign such as an ad or an email link.

It is used to calculate search engine traffic, ad campaigns and page navigation within your own site. The cookie is updated with each page view to your site.

Expiration: 6 months from set/update.

Format: __utmz=<domainHash>.<campaignCreation>.<campaignSessions>.<responseCount>.<campaignTracking>

utmhid = None

Session timeout cookie parameter __utmb

Will never be sent with requests, but stays here for documentation completeness.

This cookie is used to establish and continue a user session with your site. When a user views a page on your site, the Google Analytics code attempts to update this cookie. If it does not find the cookie, a new one is written and a new session is established.

Each time a user visits a different page on your site, this cookie is updated to expire in 30 minutes, thus continuing a single session for as long as user activity continues within 30-minute intervals.

This cookie expires when a user pauses on a page on your site for longer than 30 minutes. You can modify the default length of a user session with the `setSessionTimeout()` method.

Expiration: 30 minutes from set/update.

Format: `__utmb=<domainHash>.<trackCount>.<token>.<lastTime>`

utmsid = None

Website Optimizer cookie parameter.

This cookie is used by Website Optimizer and only set when Website Optimizer is used in combination with GA. See the Google Website Optimizer Help Center for details.

Expiration: 2 years from set/update.

utmsr = None

Visitor tracking cookie `__utma`

This cookie is typically written to the browser upon the first visit to your site from that web browser. If the cookie has been deleted by the browser operator, and the browser subsequently visits your site, a new `__utma` cookie is written with a different unique ID.

This cookie is used to determine unique visitors to your site and it is updated with each page view. Additionally, this cookie is provided with a unique ID that Google Analytics uses to ensure both the validity and accessibility of the cookie as an extra security measure.

Expiration: 2 years from set/update. Format: `__utma=<domainHash>.<uniqueId>.<firstTime>.<lastTime>.<currentTime>.<`

class `pyga.requests.Request` (*config, tracker, visitor, session*)

TYPE_SOCIAL = 'social'

This type of request is deprecated in favor of encoding custom variables within the “utme” parameter, but we include it here for completeness

build_parameters ()

Marker implementation

class `pyga.requests.SocialInteractionRequest` (*config, tracker, visitor, session, social_interaction, page*)

build_parameters ()

Marker implementation

class `pyga.requests.Tracker` (*account_id="", domain_name="", conf=None*)

Act like a Manager of all files

Properties: `account_id` – Google Analytics account ID, will be mapped to “utmac” parameter `domain_name` – Host Name, will be mapped to “utmhn” parameter `allow_hash` – Whether to generate a unique domain hash,

default is true to be consistent with the GA Javascript Client

`custom_variables` – CustomVariable instances `campaign` – Campaign instance

add_custom_variable (*custom_var*)

Equivalent of `_setCustomVar()` in GA Javascript client <http://code.google.com/apis/analytics/docs/tracking/gaTrackingCustomVariables.html>

remove_custom_variable (*index*)

Equivalent of `_deleteCustomVar()` in GA Javascript client.

track_event (*event, session, visitor*)

Equivalent of `_trackEvent()` in GA Javascript client.

track_pageview (*page, session, visitor*)

Equivalent of `_trackPageview()` in GA Javascript client.

track_social (*social_interaction, page, session, visitor*)

Equivalent of `_trackSocial()` in GA Javascript client.

track_transaction (*transaction, session, visitor*)

Combines `_addTrans()`, `_addItem()` (indirectly) and `_trackTrans()` of GA Javascript client.

class `pyga.requests.TransactionRequest` (*config, tracker, visitor, session, transaction*)

build_custom_variable_parameters (*parameters*)

The GA Javascript client doesn't send any custom variables for e-commerce requests, so we don't either.

build_parameters ()

Marker implementation

build_visitor_parameters (*parameters*)

The GA Javascript client doesn't send any visitor information for e-commerce requests, so we don't either.

Contents:

`pyga.utils.encode_uri_components` (*value*)

Mimics Javascript's `encodeURIComponent()` function for consistency with the GA Javascript client.

`pyga.utils.stringify` (*s, stype=None, fn=None*)

Converts elements of a complex data structure to strings

The data structure can be a multi-tiered one - with tuples and lists etc This method will loop through each and convert everything to string. For example - it can be - `[[{'a1': {'a2': {'a3': ('a4', timedelta(0, 563)), 'a5': {'a6': datetime()}}}}]]` which will be converted to - `[[{'a1': {'a2': {'a3': ('a4', '0:09:23'), 'a5': {'a6': '2009-05-27 16:19:52.401500'}}}}]]`

@param stype: If only one type of data element needs to be converted to string without affecting others, `stype` can be used. In the earlier example, if it is called with `stringify(s, stype=datetime.timedelta)` the result would be `[[{'a1': {'a2': {'a3': ('a4', '0:09:23'), 'a5': {'a6': datetime()}}}}]]`

Also, even though the name is `stringify`, any function can be run on it, based on parameter `fn`. If `fn` is `None`, it will be stringified.

`pyga.shutdown` ()

Fire all stored GIF requests One by One. You should call this if you set `Config.queue_requests = True`

CHAPTER 1

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pyga`, [7](#)
- `pyga.entities`, [1](#)
- `pyga.requests`, [3](#)
- `pyga.utils`, [7](#)

A

add_custom_variable() (pyga.requests.Tracker method), 6
 add_item() (pyga.entities.Transaction method), 2
 add_session() (pyga.entities.Visitor method), 3

B

build_custom_variable_parameters()
 (pyga.requests.ItemRequest method), 4
 build_custom_variable_parameters()
 (pyga.requests.TransactionRequest method), 7
 build_parameters() (pyga.requests.EventRequest
 method), 4
 build_parameters() (pyga.requests.GIFRequest method),
 4
 build_parameters() (pyga.requests.ItemRequest method),
 4
 build_parameters() (pyga.requests.PageViewRequest
 method), 4
 build_parameters() (pyga.requests.Request method), 6
 build_parameters() (pyga.requests.SocialInteractionRequest
 method), 6
 build_parameters() (pyga.requests.TransactionRequest
 method), 7
 build_visitor_parameters() (pyga.requests.ItemRequest
 method), 4
 build_visitor_parameters()
 (pyga.requests.TransactionRequest method), 7

C

Campaign (class in pyga.entities), 1
 Config (class in pyga.requests), 3
 CustomVariable (class in pyga.entities), 1

E

encode_uri_components() (in module pyga.utils), 7
 Event (class in pyga.entities), 1
 EventRequest (class in pyga.requests), 4
 extract_from_server_meta() (pyga.entities.Visitor
 method), 3

extract_from_utma() (pyga.entities.Visitor method), 3
 extract_from_utmb() (pyga.entities.Session method), 2

F

fire() (pyga.requests.GIFRequest method), 4

G

generate_hash() (pyga.entities.Visitor method), 3
 generate_unique_id() (pyga.entities.Visitor method), 3
 get_parameters() (pyga.requests.Parameters method), 5
 GIFRequest (class in pyga.requests), 4

I

Item (class in pyga.entities), 2
 ItemRequest (class in pyga.requests), 4

P

Page (class in pyga.entities), 2
 PageViewRequest (class in pyga.requests), 4
 Parameters (class in pyga.requests), 4
 pyga (module), 7
 pyga.entities (module), 1
 pyga.requests (module), 3
 pyga.utils (module), 7

R

remove_custom_variable() (pyga.requests.Tracker
 method), 7
 Request (class in pyga.requests), 6

S

Session (class in pyga.entities), 2
 shutdown() (in module pyga), 7
 SocialInteraction (class in pyga.entities), 2
 SocialInteractionRequest (class in pyga.requests), 6
 stringify() (in module pyga.utils), 7

T

track_event() (pyga.requests.Tracker method), 7

`track_pageview()` (`pyga.requests.Tracker` method), [7](#)
`track_social()` (`pyga.requests.Tracker` method), [7](#)
`track_transaction()` (`pyga.requests.Tracker` method), [7](#)
`Tracker` (class in `pyga.requests`), [6](#)
`Transaction` (class in `pyga.entities`), [2](#)
`TransactionRequest` (class in `pyga.requests`), [7](#)
`TYPE_SOCIAL` (`pyga.requests.Request` attribute), [6](#)

U

`utmcsr` (`pyga.requests.Parameters` attribute), [5](#)
`utmhid` (`pyga.requests.Parameters` attribute), [5](#)
`utmsid` (`pyga.requests.Parameters` attribute), [6](#)
`utmsr` (`pyga.requests.Parameters` attribute), [6](#)

V

`validate()` (`pyga.entities.CustomVariable` method), [1](#)
`Visitor` (class in `pyga.entities`), [2](#)