

mixpanel_api docs

version 1.2.0

Jared McFarland

January 27, 2017

Contents

Welcome to mixpanel_api's documentation!	1
mixpanel_api package	1
Module contents	1
Indices and tables	11
Index	13
Python Module Index	15

Welcome to mixpanel_api's documentation!

Contents:

mixpanel_api package

Module contents

```
class mixpanel_api.Mixpanel (api_secret, token=None, timeout=120, pool_size=None,
max_retries=10, debug=False)
```

Bases: **object**

An object for querying, importing, exporting and modifying Mixpanel data via their various APIs

```
FORMATTED_API = 'https://mixpanel.com/api'
```

```
IMPORT_API = 'https://api.mixpanel.com'
```

```
LOGGER = <logging.Logger object>
```

```
RAW_API = 'https://data.mixpanel.com/api'
```

```
VERSION = '2.0'
```

```
__init__ (api_secret, token=None, timeout=120, pool_size=None, max_retries=10, debug=False)
Initializes the Mixpanel object
```

Parameters:

- **api_secret** (*str*) -- API Secret for your project
- **token** (*str*) -- Project Token for your project, required for imports
- **timeout** (*int*) -- Time in seconds to wait for HTTP responses
- **pool_size** (*int*) -- Number of threads to use for multiprocessing, default is `cpu_count * 2`
- **max_retries** (*int*) -- Maximum number of times to retry when a 503 HTTP response is received
- **debug** (*bool*) -- Enable debug logging

```
__dispatch_batches (endpoint, item_list, prep_args)
```

Asynchronously sends batches of 50 items to the /import or /engage Mixpanel API endpoints

Parameters:

- **endpoint** (*str*) -- Can be either 'import' or 'engage'
- **item_list** (*list*) -- List of Mixpanel event data or People profiles
- **prep_args** (*list*) -- List of arguments to be provided to the appropriate `_prep` method in addition to the profile or event

```
static __export_data (data, output_file, format='json', compress=False)
```

Writes and optionally compresses Mixpanel data to disk in json or csv format

Parameters:

- **data** (*list*) -- A list of Mixpanel events or People profiles
- **output_file** (*str*) -- Name of file to write to
- **format** (*str*) -- Output format can be 'json' or 'csv' (Default value = 'json')
- **compress** (*bool*) -- Option to gzip output (Default value = False)

```
__get_engage_page (params)
```

Fetches and returns the response from an /engage request

Parameters: **params** (*dict*) -- Query parameters for the /engage API

Returns: /engage API response object

Return type: dict

`_import_data` (*data, endpoint, timezone_offset=None, ignore_alias=False*)

Base method to import either event data or People profile data as a list of dicts or from a JSON array file

Parameters:

- **data** (*list* / *str*) -- A list of event or People profile dicts or the name of a file containing a JSON array or CSV of events or People profiles
- **endpoint** (*str*) -- can be 'import' or 'engage'
- **timezone_offset** (*int*) -- UTC offset (number of hours) for the project that exported the data. Used to convert the event timestamps back to UTC prior to import. (Default value = 0)
- **ignore_alias** (*bool*) -- Option to bypass Mixpanel's alias lookup table (Default value = False)

`static _prep_event_for_import` (*event, token, timezone_offset*)

Takes an event dict and modifies it to meet the Mixpanel /import HTTP spec or dumps it to disk if it is invalid

Parameters:

- **event** (*dict*) -- A Mixpanel event dict
- **token** (*str*) -- A Mixpanel project token
- **timezone_offset** (*int*) -- UTC offset (number of hours) of the timezone setting for the project that exported the data. Needed to convert the timestamp back to UTC prior to import.

Returns: Mixpanel event dict with token added and timestamp adjusted to UTC

Return type: dict

`static _prep_params_for_profile` (*profile, token, operation, value, ignore_alias, dynamic*)

Takes a People profile dict and returns the parameters for an /engage API update

Parameters:

- **profile** (*dict*) -- A Mixpanel People profile dict
- **token** (*str*) -- A Mixpanel project token
- **operation** (*str*) -- A Mixpanel /engage API update operation
<https://mixpanel.com/help/reference/http#update-operations>
- **value** -- The value to use for the operation or a function that takes a People profile and returns the value to use
- **ignore_alias** (*bool*) -- Option to bypass Mixpanel's alias lookup table
- **dynamic** (*bool*) -- Should be set to True if value param is a function, otherwise false.

Returns: Parameters for a Mixpanel /engage API update

Return type: dict

`_send_batch` (*endpoint, batch, retries=0*)

POST a single batch of data to a Mixpanel API and return the response

Parameters:

- **endpoint** (*str*) -- Should be 'import' or 'engage'
- **batch** (*list*) -- List of Mixpanel event data or People updates to import.
- **retries** (*int*) -- Max number of times to retry if we get a HTTP 503 response (Default value = 0)

Raise: Raises for any HTTP error other than 503

Returns: HTTP response from Mixpanel API

Return type: dict

deduplicate_people (profiles=None, prop_to_match='\$email', merge_props=False, case_sensitive=False, backup=True, backup_file=None)

Determines duplicate profiles based on the value of a specified property. The profile with the latest \$last_seen is kept and the others are deleted. Optionally adds any properties from the profiles to be deleted to the remaining profile using \$set_once. Backup files are always created.

Parameters:

- **profiles** (*list* / *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. If this is None all profiles with prop_to_match set will be downloaded. (Default value = None)
- **prop_to_match** (*str*) -- Name of property whose value will be used to determine duplicates (Default value = '\$email')
- **merge_props** (*bool*) -- Option to call \$set_once on remaining profile with all props from profiles to be deleted. This ensures that any properties that existed on the duplicates but not on the remaining profile are preserved. (Default value = False)
- **case_sensitive** (*bool*) -- Option to use case sensitive or case insensitive matching (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

static dt_from_iso (profile)

Takes a Mixpanel People profile and returns a datetime object for the \$last_seen value or datetime.min if \$last_seen is not set

Parameters: **profile** (*dict*) -- A Mixpanel People profile dict

Returns: A datetime representing the \$last_seen value for the given profile

Return type: datetime

event_counts_to_people (from_date, events)

Sets the per user count of events in events list param as People properties

Parameters:

- **from_date** (*datetime* / *str*) -- A datetime or a date string of format YYYY-MM-DD to begin counting from
- **events** (*list[str]*) -- A list of strings of event names to be counted

static event_object_from_csv_row (row, header, event_index=None, distinct_id_index=None, time_index=None)

Converts a row from a csv file into a Mixpanel event dict

Parameters:

- **row** (*list*) -- A list containing the Mixpanel event data from a csv row
- **header** (*list*) -- A list containing the csv column headers
- **event_index** (*int*) -- Index of the event name in row list, if None this method will determine the index (Default value = None)
- **distinct_id_index** (*int*) -- Index of the distinct_id in row list, if None this method will determine the index (Default value = None)
- **time_index** (*int*) -- Index of the time property in row list, if None this method will determine the index (Default value = None)

Returns: A Mixpanel event object

Return type: dict

export_events (output_file, params, format='json', compress=False)

Queries the /export API and writes the Mixpanel event data to disk as a JSON or CSV file. Optionally gzip file.

<https://mixpanel.com/help/reference/exporting-raw-data#export-api-reference>

Parameters:

- **output_file** (*str*) -- Name of the file to write to
- **params** (*dict*) -- Parameters to use for the /export API request
- **format** (*str*) -- Can be either 'json' or 'csv' (Default value = 'json')
- **compress** (*bool*) -- Option to gzip output_file (Default value = False)

export_people (output_file, params={}, timezone_offset=None, format='json', compress=False)

Queries the /engage API and writes the Mixpanel People profile data to disk as a JSON or CSV file. Optionally gzip file.

<https://mixpanel.com/help/reference/data-export-api#people-analytics>

Parameters:

- **output_file** (*str*) -- Name of the file to write to
- **params** (*dict*) -- Parameters to use for the /engage API request (Default value = {})
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **format** (*str*) -- (Default value = 'json')
- **compress** (*bool*) -- (Default value = False)

formatter = <logging.Formatter object>

static gzip_file (filename)

gzip an existing file

Parameters: **filename** (*str*) -- Path to a file to be gzipped

import_events (data, timezone_offset)

Imports a list of Mixpanel event dicts or a file containing a JSON array of Mixpanel events.

<https://mixpanel.com/help/reference/importing-old-events>

Parameters:

- **data** (*list* | *str*) -- A list of Mixpanel event dicts or the name of a file containing a JSON array or CSV of Mixpanel events
- **timezone_offset** (*int*) -- UTC offset (number of hours) for the project that exported the data. Used to convert the event timestamps back to UTC prior to import.

import_people (data, ignore_alias=False)

Imports a list of Mixpanel People profile dicts or a file containing a JSON array or CSV of Mixpanel People

profiles

<https://mixpanel.com/help/reference/data-export-api#people-analytics>

Parameters:

- **data** (*list* | *str*) -- A list of Mixpanel People profile dicts or the name of a file containing a JSON array of Mixpanel People profiles.
- **ignore_alias** (*bool*) -- Option to bypass Mixpanel's alias lookup table (Default value = False)

jql_operation (jql_script, people_operation, update_value, jql_params=None, ignore_alias=False, backup=True, backup_file=None)

Perform a JQL query to return a JSON array of objects that can then be used to dynamically construct People

updates via the update_value

Parameters:

- **jql_script** (*str*) -- String containing a JQL script to run. The result should be an array of objects. Those objects should contain at least a \$distinct_id key
- **people_operation** (*str*) -- A Mixpanel People update operation
- **update_value** -- Can be a static value applied to all \$distinct_ids or a user defined function or lambda that expects a dict containing at least a \$distinct_id key as its only parameter and returns the value to use for the update on that \$distinct_id
- **jql_params** (*dict*) -- Optional dict that will be made available to the script as the params global variable. (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

static list_from_argument (*arg*)

Returns a list given a string with the path to file of Mixpanel data or an existing list

Parameters: **arg** (*list | str*) -- A string file path or a list

Returns: A list of Mixpanel events or profiles

Return type: list

static list_from_items_filename (*filename*)

Returns a list of Mixpanel events or profiles given the path to a file containing such data

Parameters: **filename** (*str*) -- Path to a file containing a JSON array of Mixpanel event or People profile data

Returns: A list of Mixpanel events or profiles

Return type: list

log_level = None

The logger is a singleton for the Mixpanel class, so multiple instances of the Mixpanel class will use the same logger. Subsequent instances can upgrade the logging level to debug but they cannot downgrade it.

people_add (*value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None*)

Increments numeric properties on the specified profiles using the \$add operation and optionally creates a backup file

Parameters:

- **value** (*dict[str, float] | (profile) -> dict[str, float]*) -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list | str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_append (*value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None*)

Appends values to list properties on the specified profiles using the \$append operation and optionally creates a backup file.

Parameters:

- **value** (*dict* | (*profile*) -> *dict*) -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list* | *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_change_property_name (old_name, new_name, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None, unset=True)

Copies the value of an existing property into a new property and optionally unsets the existing property. Optionally creates a backup file.

Parameters:

- **old_name** -- The name of an existing property.
- **new_name** -- The new name to replace the old_name with
- **profiles** (*list* | *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. If both query_params and profiles are None all profiles with old_name set are targeted. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)
- **unset** (*bool*) -- Option to unset the old_name property (Default value = True)

people_delete (profiles=None, query_params=None, timezone_offset=None, ignore_alias=True, backup=True, backup_file=None)

Deletes the specified People profiles with the \$delete operation and optionally creates a backup file

Parameters:

- **profiles** (*list / str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** -- True or False (Default value = True)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

static people_object_from_csv_row (row, header, distinct_id_index=None)

Converts a row from a csv file into a Mixpanel People profile dict

Parameters:

- **row** (*list*) -- A list containing the Mixpanel event data from a csv row
- **header** (*list*) -- A list containing the csv column headers
- **distinct_id_index** (*int*) -- Index of the distinct_id in row list, if None this method will determine the index (Default value = None)

Returns: A Mixpanel People profile object

Return type: dict

people_operation (operation, value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=False, backup_file=None)

Base method for performing any of the People analytics update operations

<https://mixpanel.com/help/reference/http#update-operations>

Parameters:

- **operation** (*str*) -- A string with name of a Mixpanel People operation, like \$set or \$delete
- **value** -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list / str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_remove (value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None)

Removes values from list properties on the specified profiles using the \$remove operation and optionally creates a backup file

Parameters:

- **value** (*dict* | (*profile*) -> *dict*) -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list* | *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_revenue_property_from_transactions (profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None)

Creates a property named 'Revenue' for the specified profiles by summing their \$transaction \$amounts and optionally creates a backup file

Parameters:

- **profiles** (*list* | *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. If both query_params and profiles are None, all profiles with \$transactions are targeted. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_set (value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None)

Sets People properties for the specified profiles using the \$set operation and optionally creates a backup file

Parameters:

- **value** -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list* | *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_set_once (value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=False, backup_file=None)

Sets People properties for the specified profiles only if the properties do not yet exist, using the \$set_once operation and optionally creates a backup file

Parameters:

- **value** -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list* / *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_union (value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None)

Union a list of values with list properties on the specified profiles using the \$union operation and optionally create a backup file

Parameters:

- **value** (*dict[str, list]* | (*profile*) -> *dict[str, list]*) -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list* / *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

people_unset (value, profiles=None, query_params=None, timezone_offset=None, ignore_alias=False, backup=True, backup_file=None)

Unsets properties from the specified profiles using the \$unset operation and optionally creates a backup file

Parameters:

- **value** (*list* | (*profile*) -> *list*) -- Can be a static value applied to all profiles or a user-defined function (or lambda) that takes a profile as its only parameter and returns the value to use for the operation on the given profile
- **profiles** (*list* | *str*) -- Can be a list of profiles or the name of a file containing a JSON array or CSV of profiles. Alternative to query_params. (Default value = None)
- **query_params** (*dict*) -- Parameters to query /engage API. Alternative to profiles param. (Default value = None)
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)
- **ignore_alias** (*bool*) -- True or False (Default value = False)
- **backup** (*bool*) -- True to create backup file otherwise False (default)
- **backup_file** (*str*) -- Optional filename to use for the backup file (Default value = None)

static_properties_from_csv_row (*row*, *header*, *ignored_columns*)
Converts a row from a csv file into a properties dict

Parameters:

- **row** (*list*) -- A list containing the csv row data
- **header** (*list*) -- A list containing the column headers (property names)
- **ignored_columns** (*list*) -- A list of columns (properties) to exclude

query_engage (*params*={}, *timezone_offset*=None)
Queries the /engage API and returns a list of Mixpanel People profile dicts
<https://mixpanel.com/help/reference/data-export-api#people-analytics>

Parameters:

- **params** (*dict*) -- Parameters to use for the /engage API request. Defaults to returning all profiles. (Default value = {})
- **timezone_offset** (*int*) -- UTC offset in hours of project timezone setting, used to calculate as_of_timestamp parameter for queries that use behaviors. Required if query_params contains behaviors (Default value = None)

Raises: **RuntimeError** -- Raises Runtime error if params include behaviors and timezone_offset is None

Returns: A list of Mixpanel People profile dicts

Return type: list

query_export (*params*)
Queries the /export API and returns a list of Mixpanel event dicts
<https://mixpanel.com/help/reference/exporting-raw-data#export-api-reference>

Parameters: **params** (*dict*) -- Parameters to use for the /export API request

Returns: A list of Mixpanel event dicts

Return type: list

query_jql (*script*, *params*=None)
Query the Mixpanel JQL API
<https://mixpanel.com/help/reference/jql/api-reference#api/access>

Parameters:

- **script** (*str*) -- String containing a JQL script to run
- **params** (*dict*) -- Optional dict that will be made available to the script as the params global variable.

request (*base_url*, *path_components*, *params*, *method*='GET')
Base method for sending HTTP requests to the various Mixpanel APIs

Parameters:

- **base_url** (*str*) -- Ex: <https://api.mixpanel.com>
- **path_components** (*list*) -- endpoint path as list of strings
- **params** (*dict*) -- dictionary containing the Mixpanel parameters for the API request
- **method** (*str*) -- GET or POST (Default value = 'GET')

Returns: JSON data returned from API

Return type: str

static response_handler_callback (*response*)

Takes a Mixpanel API response and checks the status

Parameters: **response** (*str*) -- A Mixpanel API JSON response

Raises: **RuntimeError** -- Raises RuntimeError if status is not equal to 1

sh = <logging.StreamHandler object>

static sum_transactions (*profile*)

Returns a dict with a single key, 'Revenue' and the sum of all \$transaction \$amounts for the given profile as the value

Parameters: **profile** (*dict*) -- A Mixpanel People profile dict

Returns: A dict with key 'Revenue' and value containing the sum of all \$transactions for the give profile

Return type: dict

static unicode_urlencode (*params*)

URL encodes a dict of Mixpanel parameters

Parameters: **params** (*dict*) -- A dict containing Mixpanel parameter names and values

Returns: A URL encoded string

Return type: str

static write_items_to_csv (*items, output_file*)

Writes a list of Mixpanel events or profiles to a csv file

Parameters:

- **items** (*list*) -- A list of Mixpanel events or profiles
- **output_file** (*str*) -- A string containing the path and filename to write to

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

Index

`__init__()` (mixpanel_api.Mixpanel method)
`_dispatch_batches()` (mixpanel_api.Mixpanel method)
`_export_data()` (mixpanel_api.Mixpanel static method)
`_get_engage_page()` (mixpanel_api.Mixpanel method)
`_import_data()` (mixpanel_api.Mixpanel method)
`_prep_event_for_import()` (mixpanel_api.Mixpanel static method)
`_prep_params_for_profile()` (mixpanel_api.Mixpanel static method)
`_send_batch()` (mixpanel_api.Mixpanel method)

D

`deduplicate_people()` (mixpanel_api.Mixpanel method)
`dt_from_iso()` (mixpanel_api.Mixpanel static method)

E

`event_counts_to_people()` (mixpanel_api.Mixpanel method)
`event_object_from_csv_row()` (mixpanel_api.Mixpanel static method)
`export_events()` (mixpanel_api.Mixpanel method)
`export_people()` (mixpanel_api.Mixpanel method)

F

`FORMATTED_API` (mixpanel_api.Mixpanel attribute)
`formatter` (mixpanel_api.Mixpanel attribute)

G

`gzip_file()` (mixpanel_api.Mixpanel static method)

I

`IMPORT_API` (mixpanel_api.Mixpanel attribute)
`import_events()` (mixpanel_api.Mixpanel method)
`import_people()` (mixpanel_api.Mixpanel method)

J

`jql_operation()` (mixpanel_api.Mixpanel method)

L

`list_from_argument()` (mixpanel_api.Mixpanel static method)
`list_from_items_filename()` (mixpanel_api.Mixpanel static method)

`log_level` (mixpanel_api.Mixpanel attribute)
`LOGGER` (mixpanel_api.Mixpanel attribute)

M

`Mixpanel` (class in mixpanel_api)
`mixpanel_api` (module)

P

`people_add()` (mixpanel_api.Mixpanel method)
`people_append()` (mixpanel_api.Mixpanel method)
`people_change_property_name()` (mixpanel_api.Mixpanel method)
`people_delete()` (mixpanel_api.Mixpanel method)
`people_object_from_csv_row()` (mixpanel_api.Mixpanel static method)
`people_operation()` (mixpanel_api.Mixpanel method)
`people_remove()` (mixpanel_api.Mixpanel method)
`people_revenue_property_from_transactions()` (mixpanel_api.Mixpanel method)
`people_set()` (mixpanel_api.Mixpanel method)
`people_set_once()` (mixpanel_api.Mixpanel method)
`people_union()` (mixpanel_api.Mixpanel method)
`people_unset()` (mixpanel_api.Mixpanel method)
`properties_from_csv_row()` (mixpanel_api.Mixpanel static method)

Q

`query_engage()` (mixpanel_api.Mixpanel method)
`query_export()` (mixpanel_api.Mixpanel method)
`query_jql()` (mixpanel_api.Mixpanel method)

R

`RAW_API` (mixpanel_api.Mixpanel attribute)
`request()` (mixpanel_api.Mixpanel method)
`response_handler_callback()` (mixpanel_api.Mixpanel static method)

S

`sh` (mixpanel_api.Mixpanel attribute)
`sum_transactions()` (mixpanel_api.Mixpanel static method)

U

`unicode_urlencode()` (mixpanel_api.Mixpanel static method)

V

VERSION (mixpanel_api.Mixpanel attribute)

W

write_items_to_csv() (mixpanel_api.Mixpanel static method)

Python Module Index

m

[mixpanel_api](#)