osssss Documentation

Release 0.0.33

khazhyk

Nov 14, 2018

Contents

1 API Reference **3**
- 1.1 OsuApi . . . 3
- 1.2 Built-in Connectors . . . 5
- 1.3 Model . . . 5
- 1.4 Enums . . . 11
- 1.5 Thrown Errors . . . 14

2 Indices and tables **15**

Python Module Index **17**

Contents:

CHAPTER 1

API Reference

1.1 OsuApi

class osuapi.osu.**OsuApi**(*key*, *, *connector*)

osu! api client.

Parameters

- **key** – The osu! api key used for authorization.
- **connector** – The osuapi connector used for making requests. The library comes with two implementations, *osuapi.connectors.AHConnector* for using aiohttp, and *osuapi.connectors.ReqConnector* for using requests.

get_user(*username*, *, *mode=<OsuMode.osu: 0>*, *event_days=31*)

Get a user profile.

Parameters

- **username** (*str or int*) – A *str* representing the user's username, or an *int* representing the user's id.
- **mode** (*osuapi.enums.OsuMode*) – The osu! game mode for which to look up. Defaults to osu!standard.
- **event_days** (*int*) – The number of days in the past to look for events. Defaults to 31 (the maximum).

get_user_best(*username*, *, *mode=<OsuMode.osu: 0>*, *limit=50*)

Get a user's best scores.

Parameters

- **username** (*str or int*) – A *str* representing the user's username, or an *int* representing the user's id.
- **mode** (*osuapi.enums.OsuMode*) – The osu! game mode for which to look up. Defaults to osu!standard.

- **limit** – The maximum number of results to return. Defaults to 50, maximum 100.

get_user_recent(*username*, *, *mode=<OsuMode.osu: 0>*, *limit=10*)
Get a user's most recent scores, within the last 24 hours.

Parameters

- **username** (*str or int*) – A *str* representing the user's username, or an *int* representing the user's id.
- **mode** (*osuapi.enums.OsuMode*) – The osu! game mode for which to look up. Defaults to osu!standard.
- **limit** – The maximum number of results to return. Defaults to 10, maximum 50.

get_scores(*beatmap_id*, *, *username=None*, *mode=<OsuMode.osu: 0>*, *mods=None*, *limit=50*)
Get the top scores for a given beatmap.

Parameters

- **beatmap_id** – Individual Beatmap ID to lookup.
- **username** (*str or int*) – A *str* representing the user's username, or an *int* representing the user's id. If specified, restricts returned scores to the specified user.
- **mode** (*osuapi.enums.OsuMode*) – The osu! game mode for which to look up. Defaults to osu!standard.
- **mods** (osuap:class:`osuapi.enums.OsuMod) – If specified, restricts returned scores to the specified mods.
- **limit** – Number of results to return. Defaults to 50, maximum 100.

get_beatmaps(*, *since=None*, *beatmapset_id=None*, *beatmap_id=None*, *username=None*, *mode=None*, *include_converted=False*, *beatmap_hash=None*, *limit=500*)
Get beatmaps.

Parameters

- **since** (*datetime*) – If specified, restrict results to beatmaps *ranked* after this date.
- **beatmapset_id** – If specified, restrict results to a specific beatmap set.
- **beatmap_id** – If specified, restrict results to a specific beatmap.
- **username** (*str or int*) – A *str* representing the user's username, or an *int* representing the user's id. If specified, restrict results to a specific user.
- **mode** (*osuapi.enums.OsuMode*) – If specified, restrict results to a specific osu! game mode.
- **include_converted** (*bool*) – Whether or not to include autoconverts. Defaults to false.
- **beatmap_hash** – If specified, restricts results to a specific beatmap hash.
- **limit** – Number of results to return. Defaults to 500, maximum 500.

get_match(*match_id*)
Get a multiplayer match.

Parameters **match_id** – The ID of the match to retrieve. This is the ID that you see in a online multiplayer match summary. This does not correspond the in-game game ID.

1.2 Built-in Connectors

Build in connectors.

Connectors have to implement *process_request*.

class osuapi.connectors.**AHConnector**(*sess=None*, *loop=None*)
Connector implementation using aiohttp.

process_request(*endpoint*, *data*, *type_*, *retries=5*)
Make and process the request.

This can raise anything aiohttp.get() can raise, or osuapi.HTTPError if we run out of retries.

Parameters

- **endpoint** (*str*) – The HTTP endpoint to make a request to
- **data** (*dict*) – The parameters for making the HTTP request
- **type** (*type*) – A converter to which to pass the response json and return.
- **retries** (*int*) – Maximum number of times to try request.

class osuapi.connectors.**ReqConnector**(*sess=None*)
Connector implementation using requests.

process_request(*endpoint*, *data*, *type_*, *retries=5*)
Make and process the request.

This can raise anything requests.get() can raise, or osuapi.HTTPError if we run out of retries.

Parameters

- **endpoint** (*str*) – The HTTP endpoint to make a request to
- **data** (*dict*) – The parameters for making the HTTP request
- **type** (*type*) – A converter to which to pass the response json and return.
- **retries** (*int*) – Maximum number of times to try request.

1.3 Model

Different classes to parse dicts/lists returned from json into meaningful data objects.

class osuapi.model.**Score**(*dct*)
Abstract class representing a score.

score
int – The score value

maxcombo
int – Largest combo achieved

count50
int – Number of “50” hits. In catch: number of “droplet” hits

count100
int – Number of “100” hits In taiko: number of “good” hits In catch: number of “drop” hits

count300
int – Number of “300” hits In taiko: number of “great” hits In catch: number of “fruit” hits

countmiss
int – Number of misses In catch: number of "fruit" or "drop" misses

countkatu
int – Number of "katu" sections (only 100s and 300s) In taiko: number of "double good" hits In mania: number of "200" hits In catch: number of "droplet" misses

countgeki
int – Number of "geki" sections (only 300s) In taiko: number of "double great" hits In mania: number of "rainbow 300" hits

perfect
bool – If the play is a full combo (maxcombo is maximal)

user_id
int – ID of user who played.

rank
str – Letter rank achieved

See also:

<https://osu.ppy.sh/wiki/Score>

accuracy(*mode: osuapi.enums.OsuMode*)
Calculated accuracy.

See also:

<https://osu.ppy.sh/help/wiki/Accuracy>

class osuapi.model.**TeamScore**(*dct*)
Class representing a score in a multiplayer team game.

See *Score*

slot
int – Which multiplayer slot the player was in.

team
int – Which multiplayer team the player was in.

passed
bool – If the score is passing.

See also:

<https://osu.ppy.sh/wiki/Score>

class osuapi.model.**RecentScore**(*dct*)
Class representing a recent score.

See *Score*

beatmap_id
int – Beatmap the score is for.

enabled_mods
osuapi.enums.OsuMod – Enabled modifiers

date
datetime – When the score was played.

See also:

<https://osu.ppy.sh/wiki/Score>

class osuapi.model.**SoloScore**(*dct*)
Class representing a score in singleplayer.

See *Score*

beatmap_id
int – Beatmap the score is for.

pp
Optional[float] – How much PP the score is worth, or None if not eligible for PP.

enabled_mods
osuapi.enums.OsuMod – Enabled modifiers

date
datetime – When the score was played.

See also:

<https://osu.ppy.sh/wiki/Score>

class osuapi.model.**BeatmapScore**(*dct*)
Class representing a score attached to a beatmap.

See *Score*

username
str – Name of user.

pp
Optional[float] – How much PP the score is worth, or None if not eligible for PP.

enabled_mods
osuapi.enums.OsuMod – Enabled modifiers

date
datetime – When the score was played.

score_id
int – ID of score.

replay_available
bool – If a replay is available.

See also:

<https://osu.ppy.sh/wiki/Score>

class osuapi.model.**UserEvent**(*dct*)
Class representing individual user events.

display_html
str – HTML for the event.

beatmap_id
Optional[int] – Beatmap this event occured on, or None if the event has no beatmap.

beatmapset_id
Optional[int] – Beatmap set this event occured on, or None if the event has no beatmap.

date
datetime – Date this event occured.

epicfactor
int – Epic factor (between 1 and 32)

class osuapi.model.**User**(*dct*)
Class representing a user.

user_id
int – User's unique identifier.

username
str – User's name.

count300
int – Career total of "300" hits.

count100
int – Career total of "100" hits.

count50
int – Career total of "50" hits.

playcount
int – Career total play count.

ranked_score
int – Total sum of the best scores from all the ranked beatmaps played online.

total_score
int – Total sum of all scores on ranked beatmaps, including failed trails.

pp_rank
int – Global ranking place.

level
float – User's level

pp_raw
float – User's performance points

total_seconds_played
int – User's total playtime

accuracy
float – Weighted average of accuracy on top plays.

count_rank_ssh
int – Career total of SSH ranks.

count_rank_ss
int – Career total of SS ranks.

count_rank_sh
int – Career total of SH ranks.

count_rank_s
int – Career total of S ranks.

count_rank_a
int – Career total of A ranks.

country
str – Country the user is registered to.

pp_country_rank
int – Country ranking place.

events
list[dict] – Information about recent "interesting" events.

See also:

<https://osu.ppy.sh/wiki/Score>

class osuapi.model.**Beatmap**(*dct*)
Class representing a beatmap

approved
BeatmapStatus – Whether or not the map has been ranked.

approved_date
Optional[datetime] – When the beatmap was ranked, or None.

last_update
datetime – Last time the map was updated.

artist
str – Music metadata.

beatmap_id
int – Unique identifier for beatmap.

beatmapset_id
int – Unique identifier for set this beatmap belongs to.

bpm
float – Speed of map in beats per minute.

creator
str – Username of map creator.

creator_id
int – ID of the map creator.

difficultyrating
float – Star rating of a map.

diff_size
float – Circle Size. (CS)

diff_overall
float – Overall Difficulty. (OD)

diff_approach
float – Approach rate. (AR)

diff_drain
float – Health Drain (HP)

hit_length
int – Playable time in seconds. (Drain time)

source
str – Source of the music

genre_id
osuapi.enums.BeatmapGenre – Genre of the music.

language_id
osuapi.enums.BeatmapLanguage – Language of the music.

title
str – Title of the song.

total_length
int – Total song length in seconds.

version
str – Difficulty name.

file_md5
str – md5 hash of map.

mode
osuapi.enums.OsuMode – Game mode for the map.

tags
str – Space delimited tags for the map.

favourite_count
int – Number of users that have favorited this map.

playcount
int – Number of times this map has been played (including fails)/

passcount
int – Number of times this map has been passed.

max_combo
Optional[int] – Maximum possible combo.

See also:

<https://osu.ppy.sh/wiki/Beatmaps>

class osuapi.model.**MatchMetadata**(*dct*)
Class representing info about a match.

match_id
int – Unique identifier for this match.

name
str – Name of the match when it was first created.

start_time
datetime – When the match was created.

end_time
Optional[datetime] – When the match was ended, or None.

class osuapi.model.**Game**(*dct*)
Class representing an individual multiplayer game.

game_id
int – Unique identifier for this game.

start_time
datetime – When the game started.

end_time
datetime – When the game ended.

beatmap_id
int – Beatmap played.

play_mode
osuapi.enums.OsuMode – Game mode.

match_type
Not really sure...

scoring_type
osuapi.enums.ScoringType – Scoring type of game.

team_type
osuapi.enums.TeamType – Team type of the game.

mods
osuapi.enums.OsuMod – Modifiers enabled for all players.

scores
list[*TeamScore*] – List of scores for all players.

class osuapi.model.**Match**(*dct*)
Class representing a match's info and collection of games.

1.4 Enums

Enums and flags.

class osuapi.enums.**OsuMode**
Enum representing osu! game mode.

osu = 0

taiko = 1

ctb = 2

mania = 3

class osuapi.enums.**OsuMod**(*value*, *shortname=''*)
Bitwise Flags representing osu! mods.

Notes

```
# Check if a given flag is set.
OsuMod.HardRock in flags

# Check if a given flag is not set.
OsuMod.HardRock not in flags

# Check if all given flags are set.
flags.contains_all(OsuMod.Hidden | OsuMod.HardRock)

# Check if any of given flags are set.
OsuMod.keyMod in flags
```

NoMod = <OsuMod NoMod>

NoFail = <OsuMod NoFail>

Easy = <OsuMod Easy>

NoVideo = <OsuMod NoVideo>

Hidden = <OsuMod Hidden>

HardRock = <OsuMod HardRock>

SuddenDeath = <OsuMod SuddenDeath>

DoubleTime = <OsuMod DoubleTime>

Relax = <OsuMod Relax>

HalfTime = <OsuMod HalfTime>

Nightcore = <OsuMod Nightcore>

Flashlight = <OsuMod Flashlight>

Autoplay = <OsuMod Autoplay>

SpunOut = <OsuMod SpunOut>

Autopilot = <OsuMod Autopilot>

Perfect = <OsuMod Perfect>

Key4 = <OsuMod Key4>

Key5 = <OsuMod Key5>

Key6 = <OsuMod Key6>

Key7 = <OsuMod Key7>

Key8 = <OsuMod Key8>

FadeIn = <OsuMod FadeIn>

Random = <OsuMod Random>

LastMod = <OsuMod LastMod>

Key9 = <OsuMod Key9>

Key10 = <OsuMod Key10>

Key1 = <OsuMod Key1>

Key3 = <OsuMod Key3>

Key2 = <OsuMod Key2>

shortname
The initialism representing this mod. (e.g. HDHR)

longname
The long name representing this mod. (e.g. Hidden DoubleTime)

FreeModAllowed = <OsuMod NoFail | Easy | Hidden | HardRock | SuddenDeath | Relax | Fla

keyMod = <OsuMod Key4 | Key5 | Key6 | Key7 | Key8>

class osuapi.enums.**BeatmapStatus**
Enum representing the ranked status of a beatmap.

See also:

<https://osu.ppy.sh/wiki/Beatmaps>

graveyard = -2

wip = -1

pending = 0

ranked = 1

approved = 2

qualified = 3

loved = 4

class osuapi.enums.**BeatmapGenre**
Enum represeting the genre of a beatmap.

any = 0

unspecified = 1

video_game = 2

anime = 3

rock = 4

pop = 5

other = 6

novelty = 7

hip_hop = 9

electronic = 10

class osuapi.enums.**BeatmapLanguage**
Enum represeting the language of a beatmap.

any = 0

other = 1

english = 2

japanese = 3

chinese = 4

instrumental = 5

korean = 6

french = 7

german = 8

swedish = 9

spanish = 10

italian = 11

class osuapi.enums.**ScoringType**
Enum representing the scoring type of a multiplayer game.

score = 0

accuracy = 1

combo = 2

score_v2 = 3

class osuapi.enums.**TeamType**
Enum representing the team type of a multiplayer game.

head_to_head = 0

tag_coop = 1

team_vs = 2

tag_team_vs = 3

1.5 Thrown Errors

exception osuapi.errors.**HTTPError**(*code*, *reason*, *body*)

CHAPTER 2

Indices and tables

- genindex
- modindex
- search

Python Module Index

o

osuapi.connectors, 5
osuapi.enums, 11
osuapi.errors, 14
osuapi.model, 5
osuapi.osu, 3

Index

A

accuracy (osuapi.enums.ScoringType attribute), 13
accuracy (osuapi.model.User attribute), 8
accuracy() (osuapi.model.Score method), 6
AHConnector (class in osuapi.connectors), 5
anime (osuapi.enums.BeatmapGenre attribute), 13
any (osuapi.enums.BeatmapGenre attribute), 13
any (osuapi.enums.BeatmapLanguage attribute), 13
approved (osuapi.enums.BeatmapStatus attribute), 13
approved (osuapi.model.Beatmap attribute), 9
approved_date (osuapi.model.Beatmap attribute), 9
artist (osuapi.model.Beatmap attribute), 9
Autopilot (osuapi.enums.OsuMod attribute), 12
Autoplay (osuapi.enums.OsuMod attribute), 12

B

Beatmap (class in osuapi.model), 9
beatmap_id (osuapi.model.Beatmap attribute), 9
beatmap_id (osuapi.model.Game attribute), 10
beatmap_id (osuapi.model.RecentScore attribute), 6
beatmap_id (osuapi.model.SoloScore attribute), 7
beatmap_id (osuapi.model.UserEvent attribute), 7
BeatmapGenre (class in osuapi.enums), 13
BeatmapLanguage (class in osuapi.enums), 13
BeatmapScore (class in osuapi.model), 7
beatmapset_id (osuapi.model.Beatmap attribute), 9
beatmapset_id (osuapi.model.UserEvent attribute), 7
BeatmapStatus (class in osuapi.enums), 12
bpm (osuapi.model.Beatmap attribute), 9

C

chinese (osuapi.enums.BeatmapLanguage attribute), 13
combo (osuapi.enums.ScoringType attribute), 13
count100 (osuapi.model.Score attribute), 5
count100 (osuapi.model.User attribute), 8
count300 (osuapi.model.Score attribute), 5
count300 (osuapi.model.User attribute), 8
count50 (osuapi.model.Score attribute), 5
count50 (osuapi.model.User attribute), 8
count_rank_a (osuapi.model.User attribute), 8
count_rank_s (osuapi.model.User attribute), 8
count_rank_sh (osuapi.model.User attribute), 8
count_rank_ss (osuapi.model.User attribute), 8
count_rank_ssh (osuapi.model.User attribute), 8
countgeki (osuapi.model.Score attribute), 6
countkatu (osuapi.model.Score attribute), 6
countmiss (osuapi.model.Score attribute), 5
country (osuapi.model.User attribute), 8
creator (osuapi.model.Beatmap attribute), 9
creator_id (osuapi.model.Beatmap attribute), 9
ctb (osuapi.enums.OsuMode attribute), 11

D

date (osuapi.model.BeatmapScore attribute), 7
date (osuapi.model.RecentScore attribute), 6
date (osuapi.model.SoloScore attribute), 7
date (osuapi.model.UserEvent attribute), 7
diff_approach (osuapi.model.Beatmap attribute), 9
diff_drain (osuapi.model.Beatmap attribute), 9
diff_overall (osuapi.model.Beatmap attribute), 9
diff_size (osuapi.model.Beatmap attribute), 9
difficultyrating (osuapi.model.Beatmap attribute), 9
display_html (osuapi.model.UserEvent attribute), 7
DoubleTime (osuapi.enums.OsuMod attribute), 12

E

Easy (osuapi.enums.OsuMod attribute), 11
electronic (osuapi.enums.BeatmapGenre attribute), 13
enabled_mods (osuapi.model.BeatmapScore attribute), 7
enabled_mods (osuapi.model.RecentScore attribute), 6
enabled_mods (osuapi.model.SoloScore attribute), 7
end_time (osuapi.model.Game attribute), 10
end_time (osuapi.model.MatchMetadata attribute), 10
english (osuapi.enums.BeatmapLanguage attribute), 13
epicfactor (osuapi.model.UserEvent attribute), 7
events (osuapi.model.User attribute), 8

F

FadeIn (osuapi.enums.OsuMod attribute), 12

favourite_count (osuapi.model.Beatmap attribute), 10
file_md5 (osuapi.model.Beatmap attribute), 10
Flashlight (osuapi.enums.OsuMod attribute), 12
FreeModAllowed (osuapi.enums.OsuMod attribute), 12
french (osuapi.enums.BeatmapLanguage attribute), 13

G

Game (class in osuapi.model), 10
game_id (osuapi.model.Game attribute), 10
genre_id (osuapi.model.Beatmap attribute), 9
german (osuapi.enums.BeatmapLanguage attribute), 13
get_beatmaps() (osuapi.osu.OsuApi method), 4
get_match() (osuapi.osu.OsuApi method), 4
get_scores() (osuapi.osu.OsuApi method), 4
get_user() (osuapi.osu.OsuApi method), 3
get_user_best() (osuapi.osu.OsuApi method), 3
get_user_recent() (osuapi.osu.OsuApi method), 4
graveyard (osuapi.enums.BeatmapStatus attribute), 12

H

HalfTime (osuapi.enums.OsuMod attribute), 12
HardRock (osuapi.enums.OsuMod attribute), 12
head_to_head (osuapi.enums.TeamType attribute), 14
Hidden (osuapi.enums.OsuMod attribute), 12
hip_hop (osuapi.enums.BeatmapGenre attribute), 13
hit_length (osuapi.model.Beatmap attribute), 9
HTTPError, 14

I

instrumental (osuapi.enums.BeatmapLanguage attribute), 13
italian (osuapi.enums.BeatmapLanguage attribute), 13

J

japanese (osuapi.enums.BeatmapLanguage attribute), 13

K

Key1 (osuapi.enums.OsuMod attribute), 12
Key10 (osuapi.enums.OsuMod attribute), 12
Key2 (osuapi.enums.OsuMod attribute), 12
Key3 (osuapi.enums.OsuMod attribute), 12
Key4 (osuapi.enums.OsuMod attribute), 12
Key5 (osuapi.enums.OsuMod attribute), 12
Key6 (osuapi.enums.OsuMod attribute), 12
Key7 (osuapi.enums.OsuMod attribute), 12
Key8 (osuapi.enums.OsuMod attribute), 12
Key9 (osuapi.enums.OsuMod attribute), 12
keyMod (osuapi.enums.OsuMod attribute), 12
korean (osuapi.enums.BeatmapLanguage attribute), 13

L

language_id (osuapi.model.Beatmap attribute), 9
last_update (osuapi.model.Beatmap attribute), 9
LastMod (osuapi.enums.OsuMod attribute), 12
level (osuapi.model.User attribute), 8
longname (osuapi.enums.OsuMod attribute), 12
loved (osuapi.enums.BeatmapStatus attribute), 13

M

mania (osuapi.enums.OsuMode attribute), 11
Match (class in osuapi.model), 11
match_id (osuapi.model.MatchMetadata attribute), 10
match_type (osuapi.model.Game attribute), 11
MatchMetadata (class in osuapi.model), 10
max_combo (osuapi.model.Beatmap attribute), 10
maxcombo (osuapi.model.Score attribute), 5
mode (osuapi.model.Beatmap attribute), 10
mods (osuapi.model.Game attribute), 11

N

name (osuapi.model.MatchMetadata attribute), 10
Nightcore (osuapi.enums.OsuMod attribute), 12
NoFail (osuapi.enums.OsuMod attribute), 11
NoMod (osuapi.enums.OsuMod attribute), 11
novelty (osuapi.enums.BeatmapGenre attribute), 13
NoVideo (osuapi.enums.OsuMod attribute), 11

O

osu (osuapi.enums.OsuMode attribute), 11
OsuApi (class in osuapi.osu), 3
osuapi.connectors (module), 5
osuapi.enums (module), 11
osuapi.errors (module), 14
osuapi.model (module), 5
osuapi.osu (module), 3
OsuMod (class in osuapi.enums), 11
OsuMode (class in osuapi.enums), 11
other (osuapi.enums.BeatmapGenre attribute), 13
other (osuapi.enums.BeatmapLanguage attribute), 13

P

passcount (osuapi.model.Beatmap attribute), 10
passed (osuapi.model.TeamScore attribute), 6
pending (osuapi.enums.BeatmapStatus attribute), 13
Perfect (osuapi.enums.OsuMod attribute), 12
perfect (osuapi.model.Score attribute), 6
play_mode (osuapi.model.Game attribute), 10
playcount (osuapi.model.Beatmap attribute), 10
playcount (osuapi.model.User attribute), 8
pop (osuapi.enums.BeatmapGenre attribute), 13
pp (osuapi.model.BeatmapScore attribute), 7
pp (osuapi.model.SoloScore attribute), 7
pp_country_rank (osuapi.model.User attribute), 8
pp_rank (osuapi.model.User attribute), 8
pp_raw (osuapi.model.User attribute), 8
process_request() (osuapi.connectors.AHConnector method), 5

process_request() (osuapi.connectors.ReqConnector method), 5

Q

qualified (osuapi.enums.BeatmapStatus attribute), 13

R

Random (osuapi.enums.OsuMod attribute), 12
rank (osuapi.model.Score attribute), 6
ranked (osuapi.enums.BeatmapStatus attribute), 13
ranked_score (osuapi.model.User attribute), 8
RecentScore (class in osuapi.model), 6
Relax (osuapi.enums.OsuMod attribute), 12
replay_available (osuapi.model.BeatmapScore attribute), 7
ReqConnector (class in osuapi.connectors), 5
rock (osuapi.enums.BeatmapGenre attribute), 13

S

Score (class in osuapi.model), 5
score (osuapi.enums.ScoringType attribute), 13
score (osuapi.model.Score attribute), 5
score_id (osuapi.model.BeatmapScore attribute), 7
score_v2 (osuapi.enums.ScoringType attribute), 13
scores (osuapi.model.Game attribute), 11
scoring_type (osuapi.model.Game attribute), 11
ScoringType (class in osuapi.enums), 13
shortname (osuapi.enums.OsuMod attribute), 12
slot (osuapi.model.TeamScore attribute), 6
SoloScore (class in osuapi.model), 6
source (osuapi.model.Beatmap attribute), 9
spanish (osuapi.enums.BeatmapLanguage attribute), 13
SpunOut (osuapi.enums.OsuMod attribute), 12
start_time (osuapi.model.Game attribute), 10
start_time (osuapi.model.MatchMetadata attribute), 10
SuddenDeath (osuapi.enums.OsuMod attribute), 12
swedish (osuapi.enums.BeatmapLanguage attribute), 13

T

tag_coop (osuapi.enums.TeamType attribute), 14
tag_team_vs (osuapi.enums.TeamType attribute), 14
tags (osuapi.model.Beatmap attribute), 10
taiko (osuapi.enums.OsuMode attribute), 11
team (osuapi.model.TeamScore attribute), 6
team_type (osuapi.model.Game attribute), 11
team_vs (osuapi.enums.TeamType attribute), 14
TeamScore (class in osuapi.model), 6
TeamType (class in osuapi.enums), 14
title (osuapi.model.Beatmap attribute), 9
total_length (osuapi.model.Beatmap attribute), 10
total_score (osuapi.model.User attribute), 8
total_seconds_played (osuapi.model.User attribute), 8

U

unspecified (osuapi.enums.BeatmapGenre attribute), 13
User (class in osuapi.model), 7
user_id (osuapi.model.Score attribute), 6
user_id (osuapi.model.User attribute), 8
UserEvent (class in osuapi.model), 7
username (osuapi.model.BeatmapScore attribute), 7
username (osuapi.model.User attribute), 8

V

version (osuapi.model.Beatmap attribute), 10
video_game (osuapi.enums.BeatmapGenre attribute), 13

W

wip (osuapi.enums.BeatmapStatus attribute), 12