
Cassiopeia Documentation

Release 0.1.2

Rob Rua

November 11, 2016

1	Overview	1
2	Top Level APIs	9
3	Submodules used by APIs	29
4	Index and Search	191
	Python Module Index	193

Overview

1.1 Setup

1.1.1 Install using pip

Simply `pip install cassiopeia` to get the latest release. (See the [pip install](#) page if you do not have pip installed.)

1.1.2 Install from Source

If you would like to get Cassiopeia with the most recent updates (even before they have been pushed in an official release), you can clone the repository. Go to [Cassiopeia's Github page](#) and either download the zip or `git clone https://github.com/meraki-analytics/cassiopeia` into a directory of your choice.

Next, add the newly downloaded cassiopeia source directory to your `PYTHONPATH` environment variable. If a `PYTHONPATH` environment variable does not exist on your system (which may be true if you have a newly installed version of python), you will need to create it.

On Windows, follow the instructions [here](#). Note that if you need multiple paths on your `PYTHONPATH`, you can separate them by a `;`.

On Mac or Linux, add `export PYTHONPATH=$PYTHONPATH:<CASSIOPEIA_PATH>` to the end of your shell rc file (this should be `~/.bashrc` for most), where `<CASSIOPEIA_PATH>` is the path of the directory you cloned, or the cassiopeia.zip file you downloaded.

Restart your terminal.

For more information, consult Google.

Dependencies

Cassiopeia depends on [SQLAlchemy](<http://www.sqlalchemy.org/>). It should be automatically installed for you if you install with pip. Otherwise, do *pip install sqlalchemy*.

1.1.3 Setting Additional Environment Variables

By default, the examples in Cassiopeia look for an environment variable on your system called `DEV_KEY` to set your API key within `cassiopeia.riotapi` and `cassiopeia.baseriotapi`. You can create a new environment

variable called `DEV_KEY`, and a similarly named environment variable for your production key if you have one (although Cassiopeia will never use your production key unless you change the code).

To create an environment variable on Windows, follow the directions [here](#).

On Linux or Mac, add `export DEV_KEY='<DEVKEY>'` to the end of your shell rc file (this should be `~/.bashrc` for most), where `<DEVKEY>` is your Riot-issued API key

Restart your terminal.

1.2 Basic Usage

Below is a basic example illustrating how to use Cassiopeia. The code starts by importing the core of Cassiopeia, `riotapi`, which is used to pull data from the Riot API.

Next, the region and api key are set (you will need to input your own api key here).

The `get_summoner_by_name` method is then called to pull summoner information from the Summoner endpoint for FatalElement (the creator of this library). His summoner name and level are accessed and printed to screen.

In a similar manner, the `get_champions` and `get_challenger` methods are called to get data for every champion and to get the list of all summoners in the Challenger tier.

```
import random

from cassiopeia import riotapi

riotapi.set_region("NA")
riotapi.set_api_key("YOUR-API-KEY-HERE")

summoner = riotapi.get_summoner_by_name("FatalElement")
print("{name} is a level {level} summoner on the NA server.".format(name=summoner.name, level=summoner.level))

champions = riotapi.get_champions()
random_champion = random.choice(champions)
print("He enjoys playing LoL on all different champions, like {name}.".format(name=random_champion.name))

challenger_league = riotapi.get_challenger()
best_na = challenger_league[0].summoner
print("He's much better at writing Python code than he is at LoL. He'll never be as good as {name}."
```

You can find more examples within Cassiopeia's [examples](#) directory.

1.2.1 More Examples

- [Getting Champion Names and IDs](#)
- [Calculating K/D/A](#)
- [Checking if a Summoner is in Game](#)
- [Accessing Lane and Role Information from a Match](#)
- [Accessing More Match Data](#)
- [Working with Dates and Times](#)
- [Pull All Summoners in the Master Tier](#)
- [Advanced: Recursive Match Collection](#)

1.3 Riot API vs. Base Riot API

This section explains the differences between `cassiopeia.riotapi` and `cassiopeia.baseriotapi`.

We highly recommend using `riotapi` because it provides a suite of tools and usability improvements that make using the Riot API easy and fun.

1.3.1 Riot API

- Usage: `from cassiopeia import riotapi`
- Automatically throttles requests to fit rate limits
- Useability-focused type system that replaces foreign key ID values with the referenced object to make using the Riot API easy
 - e.g. `match.participants['Dyrus'].champion` returns a `Champion` object so you can easily access information such as the champion name or image url: `match.participants['Dyrus'].champion.name` or `match.participants['Dyrus'].champion.image.link`
- Option to lazy load referenced objects right when you need them or batch them up and eagerly load them to minimize API calls
- Caches static data and summoner information to accelerate access and reduce API load
- Available automatic databasing using [SQLAlchemy](#)

1.3.2 Base Riot API

- Usage: `from cassiopeia import baseriotapi`
- Automatically throttles requests to fit rate limits
- Meets the Riot API specification exactly and foreign keys are not auto-filled
 - e.g. `match.participants[3].championId`
- Make only the requests you want to make

1.4 Rate Limiting

If you don't know what a rate limit is, make sure you read [this page](#).

Cassiopeia will automatically throttle the number of API requests made to Riot to prevent you from going over your rate limit.

By default, Cassiopeia sets your rate limit to the standard development key rate limit: 10 requests every 10 seconds and 500 requests every 10 minutes (600 seconds) using `riotapi.set_rate_limits((10, 10), (500, 600))`.

You can override this default behavior (for example if you have a production key) using the `set_rate_limit` and `set_rate_limits` functions.

Rate limiting is provided for both `riotapi` and `baseriotapi`.

1.5 Pulling Data and Data Types

Pulling and interacting with data within `cassiopeia.riotapi` is designed to be simple and intuitive.

Let's work with a few short examples.

First, let's setup `cassiopeia.riotapi`:

```
from cassiopeia import riotapi
riotapi.set_region("NA")
riotapi.set_api_key("Put-Your-API-Key-Here")
```

Now let's pull Dyrus' summoner information:

```
dyrus = riotapi.get_summoner_by_name("Dyrus")
```

That was easy. Is he in game right now?

```
game = riotapi.get_current_game(dyrus)
if game is None:
    print("Dyrus is not in a game right now")
else:
    print("Dyrus is in a game!")
```

or better yet we can get the game using

```
game = dyrus.current_game()
```

Notice how we use Dyrus' Summoner object to access his current game. You can use Summoner objects to pull many different types of data that require Summoner information, including Leagues, Teams, MatchList, Stats, ChampionMasteries, etc.

If Dyrus is in game, what champion is he playing?

```
if game is not None:
    champion = game.participants["Dyrus"].champion
    print("Dyrus is playing {champion}".format(champion=champion.name))
```

Okay, you get the idea. Let's move on to a more in-depth example and look at one of Dyrus's recent matches.

```
match_list = dyrus.match_list()
matchreference = match_list[-1] # Get the last match reference in the list
match = matchreference.match()
```

Now that we have the match, let's see what champions each participant played:

```
for participant in match.participants:
    print(participant.champion.name)
```

That was easy! Just a for loop and print. Now let's look at some of the events that happened in the match:

```
from cassiopeia.type.core.common import EventType
for frame in match.timeline.frames:
    print("The following skill level up events occurred between minute {} and {}".format(frame.timestamp, frame.timestamp + 1))
    for event in frame:
        if event.type == EventType.skill_level_up:
            print(" {summoner} leveled up their {skill_slot}".format(summoner=event.creator.name, skill_slot=event.skill_slot))
```

We can also see how much gold every participant had at every minute:


```
for frame in match.timeline.frames:
    print("The amount of gold for each participant at minute {} was:".format(int(frame.timestamp.seconds)))
    for participant, participant_frame in frame.participant_frames.items():
        print("    {summoner} had {amount} gold".format(summoner=participant_frame.participant.summoner_name, amount=participant_frame.gold))
```

You can continue by pulling, accessing, and printing all of the information that the Riot API will return. By this point we hope you are beginning to understand the “flow” of Cassiopeia. Ideally, the code you write through Cassiopeia should be easily readable and understandable. That’s our goal. Sometimes the lines get long, but this is normal for a service like the Riot API.

1.6 Load Policies and Lazy Loading

Understanding Cassiopeia’s load policy is key to using the library to its full potential.

Note that the load policy only matters when using `riotapi`. `baseriotapi` does not use the advanced loading policies or lazy loading.

1.6.1 Object-Level Lazy Loading

Many of the API objects that the `riotapi` produces use Object-Level Lazy Loading to achieve the Cassiopeia’s usability goals. It is very rare to use all the information available from an API call, and usually you are only looking for a few specific pieces of data. Object-Level Lazy Loading ensures that time and memory are not wasted on data you are not using.

Cassiopeia will delay the loading of some objects’ attributes if those attributes require a noticeable amount of time to load. For example, when you pull a match using the `get_match` method, Cassiopeia does not immediately load the match’s `timeline` because this is a very large subset of information that you may never use. Instead, the first time you try to access `match.timeline` this data is loaded, and if you never access `match.timeline` then the data is never rendered and computation time is saved.

1.6.2 Load Policies

There are two types of load policies in Cassiopeia, Eager and Lazy, which determine how calls to the Riot API are handled.

You should think of Eager loading as a useful extension of the usual way a library would handle API requests, and Lazy loading as the “standard” way. Note that the Lazy load policy is distinct from the Object-Level Lazy Loading described above (which is always done regardless of load policy).

Normally, when you want to access information from Riot, you send them an API request. However, if you know in advance that you will need data for a bunch of different objects, you can group those calls together. For example, Riot’s Summoner endpoint allows 40 summoners to be queried at once so you shouldn’t make 40 different calls to get data for 40 summoners, you should only make one. Cassiopeia will automatically group these calls together when using its Eager loading policy.

Lazy

The Lazy loading policy will only pull additional data from Riot when you attempt to access it for the first time. This is the “standard” way of making API requests.

The Lazy loading policy can be set with `riotapi.set_load_policy("lazy")` and should be used when you only want the requested data and will not use other information that needs to be requested from Riot.

Eager

When using the Eager loading policy, Cassiopeia may perform more calls than the one you wrote. Cassiopeia will pull all data that is referenced within any objects that were pulled from Riot.

The Eager loading policy can be set with `riotapi.set_load_policy("eager")` and should be used when you need additional information about the objects that require additional API calls.

1.6.3 Example

Consider the following example to determine when to use Eager loading and when to use Lazy loading.

We will pull all summoners from the Challenger league and either print their names, or print their names and the date of their most recent game.

Lazy

Use Lazy loading when you only need to access data that is returned directly from the call you made (for this example see [Riot's documentation](#) and Cassiopeia's documentation).

```
riotapi.set_load_policy("lazy")
riotapi.print_calls(True)
challenger = riotapi.get_challenger()

for entry in challenger:
    name = entry.summoner_name
    print(" {name} ".format(name=name))
```

Eager

However, when you want to access additional information about an object that was not returned from Riot (in this case the each summoner's last modification date) you should use Eager loading.

The Riot API allows users to pull data for up to 40 summoners with one call. When `get_challenger` is called, Cassiopeia sees that `Summoner` objects are referenced within each `Entry` in the return value. After Cassiopeia has finished with the `get_challenger` request, it will then perform one or more additional API calls to `get_summoners_by_id` and pass in a list of summoner ids that were returned from `get_challenger`.

```
riotapi.set_load_policy("eager")
riotapi.print_calls(True)
challenger = riotapi.get_challenger()

for entry in challenger:
    name = entry.summoner_name
    date = entry.summoner.modify_date
    print("{name} last played a game on {date}".format(name=name, date=date))
```

If you run this code with the Lazy loading policy, when the line `date = entry.summoner.modify_date` is run, Cassiopeia will try to access `entry.summoner` but that object will not exist. Cassiopeia will then make a `get_summoner_by_id` (note `summoner` and not `summoners`) call to Riot within the `for` loop. This will require up to 40x more requests to Riot (which takes far longer and uses up requests in your rate limiter) than the Eager loading case because each summoner is pulled individually.

1.7 Caching and Automatic Databasing

Caching and databasing in Cassiopeia is being reworked. We will update this section when it is finished.

1.8 Advanced Topics

1.8.1 Understanding Core and Dto Datatypes

Core datatypes (those found in `cassiopeia.types.core`) are used by `riotapi`, and Dto datatypes (those found in `cassiopeia.types.dto`) are used by `baseriotapi`. The Dto layer fits the Riot API's specification exactly, meaning all variables have the same structure and same names as those returned from the Riot API. The Core layer is a usability layer that provides additional functionality and fixes many of the unusual variable names and structures returned by the Riot API. When Core objects are interacted with, they call lookups on the Dto layer, so modifying a Dto layer's variables will change the returned values of the Core layer. Read some of the `@property` attributes in a core layer for more information on exactly how they interact with the Dto layer.

1.8.2 Using Both Core and Dto Datatypes

You can use both `cassiopeia.riotapi` and `cassiopeia.baseriotapi` within the same program and the rate limiting will still work correctly. This may be useful if you want to use advanced functionality for some types but not others.

1.8.3 Retry 500s

By default, Cassiopeia will wait and retry if a request returns a 429 (although this should rarely happen). If you are running a long gather data script, it can be helpful to do the same on 500s. See the decorator [here](#) for an example on how to extend Cassiopeia's request functionality to retry under certain conditions (such as 500s).

1.8.4 Changing the Value of Attributes Cassiopeia Objects

All Core objects in Cassiopeia are immutable. This is deliberate to prevent users from modifying the underlying data which may break functionality.

For example, if a user was able to run

```
match.red_team = 5 # This raises AttributeError: can't set attribute
```

this would break functionality that relied on `match.red_team` being a `Team` object.

If you wish to modify Cassiopeia objects you can create a new class that uses the Cassiopeia class to during initialization. For example:

```
class Champion(cassiopeia.type.core.staticdata.Champion):
    def __init__(self, champion):
        self.name = champion.name
        self.id = champion.id

annie = riotapi.get_champion_by_name("Annie")
annie = Champion(annie)
annie.id = 100 # Does not raise an exception
print(annie.id)
```

Alternatively, you can edit the underlying Dto object (which is mutable) to alter the return values from the Core type. For example:

```
annie = riotapi.get_champion_by_name("Annie")
dto = annie.data # Get the underlying Dto object
dto.id = 100
print(annie.id)
```

Be careful when using the second method, as radically changing object types could break code which relies on data having a certain type.

1.8.5 Additional Setup

During development it can be very useful to quickly boot up a terminal to test a command. To make this easy, you can create an entirely new python package (which you can call `cass`, for example) that automatically runs the usual setup functions. Here is an example:

```
cass/__init__.py:
    from cassiopeia import riotapi

    # Sets the region, API key, and output for riotapi
    def setup(region="NA", print_calls=True, key="development"):
        riotapi.set_region(region)
        riotapi.print_calls(print_calls)

        key = key.lower()
        if(key in ("d", "dev", "development")):
            key = os.environ["DEV_KEY"]
        elif(key in ("p", "prod", "production")):
            key = os.environ["PROD_KEY"]
        riotapi.set_rate_limits((3000, 10), (180000, 600))
        riotapi.set_api_key(key)

    setup()
```

After this, you can run `from cass import riotapi` and the `set_region` and `set_api_key` functions will be run for you automatically. Also, `print_calls` will be true (which is helpful for development).

Top Level APIs

2.1 cassiopeia.riotapi

This is the primary entry point for Cassiopeia. Accesses the LoL REST API (<https://developer.riotgames.com/>) and provides the results in easy-to-use Python objects.

`cassiopeia.riotapi.create_tournament(provider_id, name='')`

Creates a tournament

Parameters

- **provider_id** (*int*) – the provider ID to specify the regional registered provider data to associate this tournament
- **name** (*str*) – the optional name of the tournament (default “”)

Returns the tournament ID

Return type *int*

`cassiopeia.riotapi.create_tournament_codes(tournament_id, team_size, spectator_type, pick_type, map_type, allowed_summoners=[], meta_data='', count=1)`

Creates tournament codes for a tournament

Parameters

- **tournament_id** (*int*) – the tournament ID to generate codes for
- **team_size** (*int*) – the team size for the tournament (1-5)
- **spectator_type** (*str* | *SpectatorType*) – the spectator availability for the tournament
- **pick_type** (*str* | *PickType*) – the pick type for the tournament
- **map_type** (*str* | *MapType*) – the map the tournament is played on
- **allowed_summoners** (*list*<*Summoner*>) – the summoners who are allowed to participate in the tournament (default [])
- **meta_data** (*str* | *object*) – meta data to be included with the tournament. Any non-string value will be cast to a string. (default “”)
- **count** (*int*) – the number of codes to generate (max 1000) (default 1)

Returns the created tournament codes

Return type *list*<*str*>

`cassiopeia.riotapi.create_tournament_provider(region, url)`

Creates a tournament provider

Parameters

- **region** (*str* | *TournamentRegion*) – the region in which the provider will be running tournaments
- **url** (*str*) – the provider’s callback URL to which tournament game results in this region should be posted. The URL must be well-formed, use the http or https protocol, and use the default port for the protocol (http URLs must use port 80, https URLs must use port 443).

Returns the tournament provider ID

Return type *int*

`cassiopeia.riotapi.get_challenger(queue_type=<Queue.ranked_solo:
‘RANKED_SOLO_5x5’>)`

Gets the challenger league

Parameters **queue_type** (*Queue*) – the queue to get the challenger league for (default *Queue.ranked_solo*)

Returns the challenger league for that queue

Return type *League*

`cassiopeia.riotapi.get_champion_by_id(id_)`

Gets a champion by ID

Parameters **id** (*int*) – the ID of the champion to get

Returns the champion

Return type *Champion*

`cassiopeia.riotapi.get_champion_by_name(name)`

Gets a champion by name

Parameters **name** (*str*) – the name of the champion to get

Returns the champion

Return type *Champion*

`cassiopeia.riotapi.get_champion_masteries(summoner)`

Gets all the ChampionMastery objects for the specified summoner

Parameters **summoner** (*Summoner*) – the summoner to get champion mastery for

Returns the summoner’s champion masteries

Return type *dict<Champion, ChampionMastery>*

`cassiopeia.riotapi.get_champion_mastery(summoner, champion)`

Gets the ChampionMastery object for the specified summoner and champion

Parameters

- **summoner** (*Summoner*) – the summoner to get champion mastery for
- **champion** (*Champion*) – the desired champion

Returns the summoner’s champion mastery value for the specified champion

Return type *ChampionMastery*

`cassiopeia.riotapi.get_champion_mastery_score(summoner)`

Gets the total champion mastery score for the specified summoner

Parameters `summoner` (`Summoner`) – the summoner to get champion mastery for

Returns the summoner's total champion mastery score

Return type `int`

`cassiopeia.riotapi.get_champions()`

Gets all the champions

Returns all the champions

Return type `list<Champion>`

`cassiopeia.riotapi.get_champions_by_id(ids)`

Gets a bunch of champions by ID

Parameters `ids` (`list<int>`) – the IDs of the champions to get

Returns the requested champions

Return type `list<Champion>`

`cassiopeia.riotapi.get_champions_by_name(names)`

Gets a bunch of champions by name

Parameters `names` (`list<str>`) – the names of the champions to get

Returns the requested champions

Return type `list<Champion>`

`cassiopeia.riotapi.get_current_game(summoner)`

Gets the game a summoner is currently in, if they're in one

Parameters `summoner` (`Summoner`) – the summoner to find an active game for

Returns the game they're in (or `None` if they aren't in one)

Return type `Game`

`cassiopeia.riotapi.get_featured_games()`

Gets the current featured game list

Returns the featured games

Return type `list<Game>`

`cassiopeia.riotapi.get_item(id_)`

Gets an item

Parameters `id` (`int`) – the ID of the item to get

Returns the item

Return type `Item`

`cassiopeia.riotapi.get_items(ids=None)`

Gets a bunch of items (or all of them)

Parameters `ids` (`list<int>`) – the IDs of the items to get (or `None` to get all items) (default `None`)

Returns the items

Return type `list<Item>`

`cassiopeia.riotapi.get_language_strings()`

Gets the locale-based string replacements for various game constants

Returns dict<str, str> the replacements

Return type return

`cassiopeia.riotapi.get_languages()`

Gets the valid locales (languages) that can be used with the API

Returns the valid locales

Return type list<str>

`cassiopeia.riotapi.get_league_entries_by_summoner(summoners)`

Gets the leagues that the summoner(s) belong(s) to, including only the requested summoner(s)' entries

Parameters **summoners** (*Summoner* | list<*Summoner*>) – the summoner(s) to get leagues for

Returns the leagues that the requested summoner(s) belong(s) to

Return type list<League> | list<list<League>>

`cassiopeia.riotapi.get_league_entries_by_team(teams)`

Gets the leagues that the team(s) belong(s) to, including only the requested team(s)' entries

Parameters **teams** (*Team* | list<*Team*>) – the team(s) to get leagues for

Returns the leagues that the requested team(s) belong(s) to

Return type list<League> | list<list<League>>

`cassiopeia.riotapi.get_leagues_by_summoner(summoners)`

Gets the leagues that the summoner(s) belong(s) to. You probably don't want to call this with LoadPolicy.eager set.

Parameters **summoners** (*Summoner* | list<*Summoner*>) – the summoner(s) to get leagues for

Returns the leagues that the requested summoner(s) belong(s) to

Return type list<League> | list<list<League>>

`cassiopeia.riotapi.get_leagues_by_team(teams)`

Gets the leagues that the team(s) belong(s) to. You probably don't want to call this with LoadPolicy.eager set.

Parameters **teams** (*Team* | list<*Team*>) – the team(s) to get leagues for

Returns the leagues that the requested team(s) belong(s) to

Return type list<League> | list<list<League>>

`cassiopeia.riotapi.get_lobby_events(tournament_code)`

Gets the lobby events that have occurred for the tournament code

Parameters **tournament_code** (*str* | *TournamentCode*) – the tournament code to get lobby events for

Returns the lobby events for that tournament code

Return type list<LobbyEvent>

`cassiopeia.riotapi.get_map_information()`

Gets specific information about each map

Returns the map information

Return type `list<MapDetails>`

`cassiopeia.riotapi.get_master(queue_type=<Queue.ranked_solo: 'RANKED_SOLO_5x5'>)`
Gets the master league

Parameters `queue_type` (`Queue`) – the queue to get the master league for (default `Queue.ranked_solo`)

Returns the master league for that queue

Return type `League`

`cassiopeia.riotapi.get_masterships(ids=None)`
Gets a bunch of masteries (or all of them)

Parameters `ids` (`list<int>`) – the IDs of the masteries to get (or `None` to get all masteries) (default `None`)

Returns the masteries

Return type `list<Mastery>`

`cassiopeia.riotapi.get_mastery(id_)`
Gets a mastery

Parameters `id` (`int`) – the ID of the mastery to get

Returns the mastery

Return type `Mastery`

`cassiopeia.riotapi.get_mastery_pages(summoners)`
Get the mastery pages for (a) summoner(s).

Parameters `ids` (`Summoner` | `list<Summoner>`) – the summoner(s) to get mastery pages for

Returns the requested summoner(s)' mastery pages

Return type `list<MasteryPage>` | `list<list<MasteryPage>>`

`cassiopeia.riotapi.get_match(id_, include_timeline=True, tournament_code='')`
Gets a match

Parameters

- `id` (`int` | `MatchReference`) – the ID of or reference to the match to get
- `include_timeline` (`bool`) – whether to include timeline data in the returned match
- `tournament_code` (`str`) – the tournament code if the match to be retrieved is from a tournament

Returns the match

Return type `Match`

`cassiopeia.riotapi.get_match_list(summoner, num_matches=0, begin_index=0, begin_time=0, end_time=0, champions=None, ranked_queues=None, seasons=None)`
Gets a summoner's match history

Parameters

- `summoner` (`Summoner`) – the summoner to get match history for
- `num_matches` (`int`) – the maximum number of matches to retrieve. 0 will get as many as possible. (default 0)

- **begin_index** (*int*) – the game index to start from (default 0)
- **begin_time** (*int* | *datetime*) – the begin time to use for fetching games (default 0)
- **end_time** (*int* | *datetime*) – the end time to use for fetching games (default 0)
- **champions** (*Champion* | *list<Champion>*) – the champion(s) to limit the results to (default None)
- **Queue** | **list<Queue>** (*ranked_queues*) – the ranked queue(s) to limit the results to (default None)
- **seasons** (*Season* | *list<Season>*) – the season(s) to limit the results to (default None)

Returns the summoner's match history

Return type *list<MatchReference>*

```
cassiopeia.riotapi.get_matches(ids, include_timeline=True, tournament_code='')
```

Gets a bunch of matches

Parameters

- **ids** (*list<int>* | *list<MatchReference>*) – the IDs of or references to the matches to get
- **include_timeline** (*bool*) – whether to include timeline data in the returned matches
- **tournament_code** (*str*) – the tournament code if the match to be retrieved is from a tournament

Returns the matches

Return type *list<Match>*

```
cassiopeia.riotapi.get_ranked_stats(summoner, season=None)
```

Gets a summoner's ranked stats

Parameters

- **summoner** (*Summoner*) – the summoner to get ranked stats for
- **season** (*Season*) – the season to get ranked stats for (None will give current season stats) (default None)

Returns the summoner's ranked stats divided by champion. The entry for None contains combined stats for all champions.

Return type *dict<Champion, AggregatedStats>*

```
cassiopeia.riotapi.get_realm()
```

Gets the realm for the current region

Returns the realm

Return type *Realm*

```
cassiopeia.riotapi.get_recent_games(summoner)
```

Gets the most recent games a summoner played

Parameters **summoner** (*Summoner*) – the summoner to get recent games for

Returns the summoner's recent games

Return type *list<Game>*

`cassiopeia.riotapi.get_requests_count (tournament=False)`

Returns the number of successful requests (no exceptions in the call) and total requests issued up to now

Parameters `tournament` (*bool*) – get the request counts for the tournament requests

Returns A (successful calls, total calls) tuple

Return type *tuple*

`cassiopeia.riotapi.get_rune (id_)`

Gets a rune

Parameters `id` (*int*) – the ID of the rune to get

Returns the rune

Return type *Rune*

`cassiopeia.riotapi.get_rune_pages (summoners)`

Get the rune pages for (a) summoner(s).

Parameters `ids` (*Summoner* | *list*<*Summoner*>) – the summoner(s) to get rune pages for

Returns the requested summoner(s)' rune pages

Return type *list*<*RunePage*> | *list*<*list*<*RunePage*>>

`cassiopeia.riotapi.get_runes (ids=None)`

Gets a bunch of runes (or all of them)

Parameters `ids` (*list*<*int*>) – the IDs of the runes to get (or *None* to get all runes) (default *None*)

Returns the runes

Return type *list*<*Rune*>

`cassiopeia.riotapi.get_shard ()`

Gets the status of the current region's shard

Returns the status of the current region's shard

Return type *ShardStatus*

`cassiopeia.riotapi.get_shards ()`

Get the list of server shards

Returns the shards

Return type *list*<*Shard*>

`cassiopeia.riotapi.get_stats (summoner, season=None)`

Gets a summoner's stats

Parameters

- **summoner** (*Summoner*) – the summoner to get stats for
- **season** (*Season*) – the season to get stats for (*None* will give current season stats) (default *None*)

Returns the summoner's stats divided by queue type

Return type *dict*<*StatSummaryType*, *StatsSummary*>

`cassiopeia.riotapi.get_summoner_by_id (id_)`

Gets a summoner by ID

Parameters `id (int)` – the ID of the summoner

Returns the summoner

Return type *Summoner*

`cassiopeia.riotapi.get_summoner_by_name (name)`

Gets a summoner by name

Parameters `name (str)` – the name of the summoner

Returns the summoner

Return type *Summoner*

`cassiopeia.riotapi.get_summoner_name (id_)`

Gets the name of a summoner by ID

Parameters `id (id)` – the summoner's ID

Returns the summoner's name

Return type *str*

`cassiopeia.riotapi.get_summoner_names (ids)`

Gets the names of a bunch of summoners by ID

Parameters `ids (list<id>)` – the summoners' IDs

Returns the summoners' names

Return type *list<str>*

`cassiopeia.riotapi.get_summoner_spell (id_)`

Gets a summoner spell

Parameters `id (int)` – the ID of the summoner spell to get

Returns the summoner spell

Return type *SummonerSpell*

`cassiopeia.riotapi.get_summoner_spells (ids=None)`

Gets a bunch of summoner spells (or all of them)

Parameters `ids (list<int>)` – the IDs of the summoner spells to get (or None to get all summoner spells) (default None)

Returns the summoner spells

Return type *list<SummonerSpell>*

`cassiopeia.riotapi.get_summoners_by_id (ids)`

Gets a bunch of summoners by ID

Parameters `ids (list<int>)` – the IDs of the summoners

Returns the summoners

Return type *list<Summoner>*

`cassiopeia.riotapi.get_summoners_by_name (names)`

Gets a bunch of summoners by name

Parameters `names (list<str>)` – the names of the summoners

Returns the summoners

Return type *list<Summoner>*

`cassiopeia.riotapi.get_team(id_)`

Gets a team by ID

id_ str the ID of the team

return Team the team

`cassiopeia.riotapi.get_teams(ids)`

Gets teams by ID

ids list<str> the IDs of the teams

return list<Team> the teams

`cassiopeia.riotapi.get_teams_by_summoner(summoners)`

Gets (a) summoner(s)' teams

summoners Summoner | list<Summoner> the summoner(s) to get teams for

return list<Team> | list<list<Team>> the summoner(s)' teams

`cassiopeia.riotapi.get_top_champion_masteryes(summoner, max_entries=3)`

Gets the top ChampionMastery objects for the specified summoner

Parameters

- **summoner** (Summoner) – the summoner to get champion mastery for
- **max_entries** (int) – the maximum number of entries to retrieve (default 3)

Returns the summoner's top champion masteryes

Return type list<ChampionMastery>

`cassiopeia.riotapi.get_tournament_code(tournament_code)`

Gets information about the tournament code

Parameters **tournament_code** (str) – the tournament code

Returns the tournament code information

Return type TournamentCode

`cassiopeia.riotapi.get_tournament_match_ids(tournament_code)`

Gets the IDs for a tournament's matches

Parameters **tournament_code** (str) – the tournament code

Returns the match ids for the tournament

Return type list<int>

`cassiopeia.riotapi.get_versions()`

Gets the valid versions of the API

Returns the valid versions

Return type list<str>

`cassiopeia.riotapi.print_calls(on)`

Sets whether to print calls to stdout as they are made

Parameters **on** (bool) – the region to query against

`cassiopeia.riotapi.set_api_key(key)`

Set your API key

Parameters **key** (str) – the key to use

`cassiopeia.riotapi.set_data_store(store)`

Sets the data store to use for caching the results of API calls.

Parameters `store` (`cassiopeia.type.api.store.DataStore`) – the data store to use for storing results

`cassiopeia.riotapi.set_load_policy(policy)`

Sets the load policy to use. Keep your load policy in mind when making calls, as different policies are better for different applications.

Parameters `policy` (`str` | `cassiopeia.type.core.common.LoadPolicy`) – the load policy to use for calls to the API

`cassiopeia.riotapi.set_locale(locale)`

Sets the locale (language) to use for calls to the Riot API. Use `get_languages()` to find valid locales.

Parameters `locale` (`str`) – the locale to use for calls to the API

`cassiopeia.riotapi.set_proxy(url, port=80)`

Sets a proxy server to tunnel requests to the Riot API through

Parameters

- `url` (`str`) – the URL of the proxy server, without port number or protocol
- `port` (`int`) – the port number to connect to (default 80)

`cassiopeia.riotapi.set_rate_limit(calls_per_epoch, seconds_per_epoch)`

Sets the rate limit for cassiopeia to manage internally

Parameters

- `calls_per_epoch` (`int`) – the number of calls allowed in each epoch
- `seconds_per_epoch` (`int`) – the number of seconds per epoch

`cassiopeia.riotapi.set_rate_limits(*limits)`

Sets the rate limits for cassiopeia to manage internally

Parameters `*limits` (`tuple...`) – the rate limits to apply. Rate limits are of the form (`calls_per_epoch`, `seconds_per_epoch`)

`cassiopeia.riotapi.set_region(region)`

Set the region to run API queries against

Parameters `region` (`str` | `cassiopeia.type.core.common.Region`) – the region to query against

`cassiopeia.riotapi.set_tournament_api_key(key)`

Set your tournament API key

Parameters `key` (`str`) – the key to use

`cassiopeia.riotapi.set_tournament_rate_limit(calls_per_epoch, seconds_per_epoch)`

Sets the tournament rate limit for cassiopeia to manage internally

Parameters

- `calls_per_epoch` (`int`) – the number of calls allowed in each epoch
- `seconds_per_epoch` (`int`) – the number of seconds per epoch

`cassiopeia.riotapi.set_tournament_rate_limits(*limits)`

Sets the tournament rate limits for cassiopeia to manage internally

Parameters **limits* (*tuple...*) – the rate limits to apply. Rate limits are of the form (calls_per_epoch, seconds_per_epoch)

```
cassiopeia.riotapi.update_tournament_code(tournament_code, allowed_summoners=[],
                                           spectator_type=None, pick_type=None,
                                           map_type=None)
```

Updates a tournament code

Parameters

- **tournament_code** (*str* | *TournamentCode*) – the tournament code to update
- **allowed_summoners** (*list*<*Summoner*>) – the summoners who are allowed to participate in the tournament (default [])
- **spectator_type** (*str* | *SpectatorType*) – the spectator availability for the tournament (default None)
- **pick_type** (*str* | *PickType*) – the pick type for the tournament (default None)
- **map_type** (*str* | *MapType*) – the map the tournament is played on (default None)

2.2 cassiopeia.baseriotapi

Accesses the LoL REST API (<https://developer.riotgames.com/>), returning Python objects matching the exact API spec.

```
cassiopeia.baseriotapi.create_tournament(parameters)
https://developer.riotgames.com/api/methods#!/1057/3649
```

Parameters *parameters* (*TournamentRegistrationParameters*) – the parameters for the tournament

Returns the tournament ID

Return type *int*

```
cassiopeia.baseriotapi.create_tournament_codes(tournament_id, parameters, count=1)
https://developer.riotgames.com/api/methods#!/1063
```

Parameters

- **tournament_id** (*int*) – the tournament ID to generate codes for
- **parameters** (*TournamentCodeParameters*) – the parameters for the tournament codes
- **count** (*int*) – the number of codes to generate (max 1000) (default 1)

Returns the created tournament codes

Return type *list*<*str*>

```
cassiopeia.baseriotapi.create_tournament_provider(parameters)
https://developer.riotgames.com/api/methods#!/1057/3646
```

Parameters *parameters* (*ProviderRegistrationParameters*) – the parameters for the provider

Returns the provider ID

Return type *int*

`cassiopeia.baseriotapi.get_challenger(queue_type)`
<https://developer.riotgames.com/api/methods#!/985/3353>

Parameters `queue_type` (*str*) – the queue type to get the challenger league for (“RANKED_SOLO_5x5”, “RANKED_TEAM_3x3”, “RANKED_TEAM_5x5”)

Returns the challenger league

Return type *League*

`cassiopeia.baseriotapi.get_champion(id_)`
<https://developer.riotgames.com/api/methods#!/968/3322>

Parameters `id` (*int*) – the ID of the champion to get

Returns the champion

Return type *Champion*

`cassiopeia.baseriotapi.get_champion_masteries(summoner_id)`
<https://developer.riotgames.com/api/methods#!/1034/3544>

Parameters `summoner_id` (*int*) – the summoner ID to get champion masteries for

Returns the summoner’s champion masteries

Return type `list<ChampionMastery>`

`cassiopeia.baseriotapi.get_champion_mastery(summoner_id, champion_id)`
<https://developer.riotgames.com/api/methods#!/1034/3545>

Parameters

- `summoner_id` (*int*) – the summoner ID to get champion mastery for
- `champion_id` (*int*) – the champion ID for the desired champion

Returns the summoner’s champion mastery value for the specified champion

Return type `list<ChampionMastery>`

`cassiopeia.baseriotapi.get_champion_mastery_score(summoner_id)`
<https://developer.riotgames.com/api/methods#!/1034/3546>

Parameters `summoner_id` (*int*) – the summoner ID to get champion masteries for

Returns the summoner’s total champion mastery score

Return type *int*

`cassiopeia.baseriotapi.get_champion_status(id_)`
<https://developer.riotgames.com/api/methods#!/1015/3443>

Parameters `id` (*int*) – the ID of the champion to look up

Returns the champion

Return type *Champion*

`cassiopeia.baseriotapi.get_champion_statuses(freeToPlay=False)`
<https://developer.riotgames.com/api/methods#!/1015/3444>

Parameters `freeToPlay` (*bool*) – whether to only get free to play champions (default False)

Returns all the champions

Return type `list<Champion>`

`cassiopeia.baseriotapi.get_champions()`
<https://developer.riotgames.com/api/methods#!/968/3326>

Returns all the champions

Return type *ChampionList*

`cassiopeia.baseriotapi.get_current_game(summoner_id)`
<https://developer.riotgames.com/api/methods#!/976/3336>

Parameters `summoner_id` (*int*) – the ID of the summoner to find an active game for

Returns the summoner's current game (or None if they aren't in one)

Return type *CurrentGameInfo*

`cassiopeia.baseriotapi.get_featured_games()`
<https://developer.riotgames.com/api/methods#!/977/3337>

Returns the current featured game list

Return type *FeaturedGames*

`cassiopeia.baseriotapi.get_item(id_)`
<https://developer.riotgames.com/api/methods#!/968/3319>

Parameters `id` (*int*) – the ID of the item to get

Returns the item

Return type *Item*

`cassiopeia.baseriotapi.get_items()`
<https://developer.riotgames.com/api/methods#!/968/3314>

Returns all the items

Return type *ItemList*

`cassiopeia.baseriotapi.get_language_strings()`
<https://developer.riotgames.com/api/methods#!/968/3316>

Returns the locale-based string replacements for various game constants

Return type *LanguageStrings*

`cassiopeia.baseriotapi.get_languages()`
<https://developer.riotgames.com/api/methods#!/968/3324>

Returns the valid locale strings for the API

Return type `list<str>`

`cassiopeia.baseriotapi.get_league_entries_by_summoner(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/985/3356>

Parameters `summoner_ids` (*int* | *list<int>*) – the summoner ID(s) to get league entries for

Returns the summoner(s)' league entries

Return type `dict<str, list<League>>`

`cassiopeia.baseriotapi.get_league_entries_by_team(team_ids)`
<https://developer.riotgames.com/api/methods#!/985/3355>

Parameters `team_ids` (*str* | *list<str>*) – the team ID(s) to get league entries for

Returns the team(s)' league entries

Return type dict<str, list<League>>

`cassiopeia.baseriotapi.get_leagues_by_summoner(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/985/3351>

Parameters `summoner_ids` (*int* | *list*<*int*>) – the summoner ID(s) to get leagues for

Returns the summoner(s)' leagues

Return type dict<str, list<League>>

`cassiopeia.baseriotapi.get_leagues_by_team(team_ids)`
<https://developer.riotgames.com/api/methods#!/985/3352>

Parameters `team_ids` (*str* | *list*<*str*>) – the team ID(s) to get leagues for

Returns the team(s)' leagues

Return type dict<str, list<League>>

`cassiopeia.baseriotapi.get_lobby_events(tournament_code)`
<https://developer.riotgames.com/api/methods#!/1057/3653>

Parameters `tournament_code` (*str*) – the tournament code to get lobby events for

Returns the lobby events for that tournament code

Return type *LobbyEventWrapper*

`cassiopeia.baseriotapi.get_maps()`
<https://developer.riotgames.com/api/methods#!/968/3328>

Returns specific information about each map

Return type *MapData*

`cassiopeia.baseriotapi.get_master(queue_type)`
<https://developer.riotgames.com/api/methods#!/985/3354>

Parameters `queue_type` (*str*) – the queue type to get the master league for
("RANKED_SOLO_5x5", "RANKED_TEAM_3x3", "RANKED_TEAM_5x5")

Returns the master league

Return type *League*

`cassiopeia.baseriotapi.get_masteries()`
<https://developer.riotgames.com/api/methods#!/968/3317>

Returns all the masteries

Return type *MasteryList*

`cassiopeia.baseriotapi.get_mastery(id_)`
<https://developer.riotgames.com/api/methods#!/968/3318>

Parameters `id` (*int*) – the ID of the mastery to get

Returns the mastery

Return type *Mastery*

`cassiopeia.baseriotapi.get_match(id_, include_timeline=True, tournament_code='')`
<https://developer.riotgames.com/api/methods#!/1014/3442>

Parameters

- **id** (*int*) – the ID of the match to get
- **include_timeline** (*bool*) – whether to include timeline data in the returned match
- **tournament_code** (*str*) – the tournament code if the match to be retrieved is from a tournament

Returns the match

Return type *MatchDetail*

```
cassiopeia.baseriotapi.get_match_list(summoner_id, num_matches=0, begin_index=0,
                                       begin_time=0, end_time=0, champion_ids=None,
                                       ranked_queues=None, seasons=None)
```

<https://developer.riotgames.com/api/methods#!/1013/3439>

Parameters

- **summoner_id** (*int*) – the ID of the summoner to get the match history for
- **num_matches** (*int*) – the maximum number of matches to retrieve. 0 will get as many as possible. (default 0)
- **begin_index** (*int*) – the game index to start from (default 0)
- **begin_time** (*int*) – the begin time to use for fetching games specified as epoch milliseconds (default 0)
- **end_time** (*int*) – the end time to use for fetching games specified as epoch milliseconds (default 0)
- **champion_ids** (*int* | *list<int>*) – the champion ID(s) to limit the results to (default None)
- **ranked_queues** (*str* | *list<str>*) – the ranked queue(s) to limit the results to (“RANKED_SOLO_5x5”, “RANKED_TEAM_3x3”, “RANKED_TEAM_5x5”) (default None)
- **seasons** (*str* | *list<str>*) – the season(s) to limit the results to (“PRESEASON3”, “SEASON3”, “PRESEASON2014”, “SEASON2014”, “PRESEASON2015”, “SEASON2015”, “PRESEASON2016”, “SEASON2016”) (default None)

Returns the summoner’s match history

Return type *MatchList*

```
cassiopeia.baseriotapi.get_ranked_stats(summoner_id, season=None)
```

<https://developer.riotgames.com/api/methods#!/1018/3452>

Parameters

- **summoner_id** (*int*) – the summoner to get ranked stats for
- **season** (*str*) – the season to get ranked stats for (“SEASON2015”, “SEASON2014”, “SEASON3”) (default None)

Returns the ranked stats for the summoner and season specified

Return type *RankedStats*

```
cassiopeia.baseriotapi.get_realm()
```

<https://developer.riotgames.com/api/methods#!/968/3325>

Returns the realm for the current region

Return type *Realm*

`cassiopeia.baseriotapi.get_recent_games(summoner_id)`
<https://developer.riotgames.com/api/methods#!/1016/3445>

Parameters `summoner_id` (*int*) – the ID of the summoner to find recent games for

Returns the summoner’s recent games

Return type *RecentGames*

`cassiopeia.baseriotapi.get_requests_count(tournament=False)`

Returns the number of successful requests (no exceptions in the call) and total requests issued up to now

Parameters `tournament` (*bool*) – get the request counts for the tournament requests

Returns A (successful calls, total calls) tuple

Return type *tuple*

`cassiopeia.baseriotapi.get_rune(id_)`
<https://developer.riotgames.com/api/methods#!/968/3321>

Parameters `id` (*int*) – the ID of the rune to get

Returns the rune

Return type *Rune*

`cassiopeia.baseriotapi.get_runes()`
<https://developer.riotgames.com/api/methods#!/968/3315>

Returns all the runes

Return type *RuneList*

`cassiopeia.baseriotapi.get_shard()`
<https://developer.riotgames.com/api/methods#!/908/3142>

Returns the status of the current region’s shard

Return type *ShardStatus*

`cassiopeia.baseriotapi.get_shards()`
<https://developer.riotgames.com/api/methods#!/908/3143>

Returns the shards (unfortunately neither Crystal nor Kirby’s)

Return type `list<Shard>`

`cassiopeia.baseriotapi.get_stats(summoner_id, season=None)`
<https://developer.riotgames.com/api/methods#!/1018/3453>

Parameters

- `summoner_id` (*int*) – the summoner to get ranked stats for
- `season` (*str*) – the season to get ranked stats for (“SEASON2015”, “SEASON2014”, “SEASON3”) (default None)

Returns the ranked stats for the summoner and season specified

Return type *PlayerStatsSummaryList*

`cassiopeia.baseriotapi.get_summoner_masteries(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3450>

Parameters `summoner_ids` (*int* | *list<int>*) – the summoner ID(s) to get mastery pages for

Returns the requested summoners' mastery pages

Return type dict<str, MasteryPages>

`cassiopeia.baseriotapi.get_summoner_names(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3451>

Parameters `summoner_ids` (`int` | `list<int>`) – the summoner ID(s) to get names for

Returns the requested summoners' names

Return type dict<str, str>

`cassiopeia.baseriotapi.get_summoner_runes(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3449>

Parameters `summoner_ids` (`int` | `list<int>`) – the summoner ID(s) to get rune pages for

Returns the requested summoners' rune pages

Return type dict<str, RunePages>

`cassiopeia.baseriotapi.get_summoner_spell(id_)`
<https://developer.riotgames.com/api/methods#!/968/3320>

Parameters `id` (`int`) – the ID of the summoner spell to get

Returns the summoner spell

Return type *SummonerSpell*

`cassiopeia.baseriotapi.get_summoner_spells()`
<https://developer.riotgames.com/api/methods#!/968/3327>

Returns all the summoner spells

Return type *SummonerSpellList*

`cassiopeia.baseriotapi.get_summoners_by_id(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3447>

Parameters `summoner_ids` (`int` | `list<int>`) – the summoner ID(s) to look up

Returns the requested summoners

Return type dict<str, Summoner>

`cassiopeia.baseriotapi.get_summoners_by_name(summoner_names)`
<https://developer.riotgames.com/api/methods#!/1017/3446>

Parameters `summoner_names` (`str` | `list<str>`) – the summoner name(s) to look up

Returns the requested summoners

Return type dict<str, Summoner>

`cassiopeia.baseriotapi.get_teams_by_id(team_ids)`
<https://developer.riotgames.com/api/methods#!/986/3358>

Parameters `team_ids` (`str` | `list<str>`) – the team ID(s) to look up

Returns the requested teams

Return type dict<str, Team>

`cassiopeia.baseriotapi.get_teams_by_summoner_id(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/986/3358>

Parameters `summoner_ids` (`int` | `list<int>`) – the summoner ID(s) to look up teams for

Returns the requested summoners' teams

Return type dict<str, list<Team>>

```
cassiopeia.baseriotapi.get_top_champion_masteryes (summoner_id, count=3)  
https://developer.riotgames.com/api/methods#!/1034/3540
```

Parameters

- **summoner_id** (*int*) – the summoner ID to get champion masteryes for
- **count** (*int*) – the maximum number of entirees to retrieve (default 3)

Returns the summoner's top champion masteryes

Return type list<ChampionMastery>

```
cassiopeia.baseriotapi.get_tournament_code (tournament_code)  
https://developer.riotgames.com/api/methods#!/1057/3643
```

Parameters **tournament_code** (*str*) – the tournament code to get information about

Returns information about the tournament code

Return type *TournamentCode*

```
cassiopeia.baseriotapi.get_tournament_match_ids (tournament_code)  
https://developer.riotgames.com/api/methods#!/1058/3656
```

Parameters **tournament_code** (*str*) – the tournament code

Returns the match ids for the tournament

Return type list<int>

```
cassiopeia.baseriotapi.get_versions ()  
https://developer.riotgames.com/api/methods#!/968/3323
```

Returns the valid API versions

Return type list<str>

```
cassiopeia.baseriotapi.print_calls (on)  
Sets whether to print calls to stdout as they are made
```

Parameters **on** (*bool*) – whether to print calls to stdout

```
cassiopeia.baseriotapi.set_api_key (key)  
Set your API key
```

Parameters **key** (*str*) – the key to use

```
cassiopeia.baseriotapi.set_locale (locale)  
Sets the locale (language) to use for calls to the Riot API. Use get_languages() to find valid locales.
```

Parameters **locale** (*str*) – the locale to use for calls to the API

```
cassiopeia.baseriotapi.set_proxy (url, port=80)  
Sets a proxy server to tunnel requests to the Riot API through
```

Parameters

- **url** (*str*) – the URL of the proxy server, without port number or protocol
- **port** (*int*) – the port number to connect to (default 80)

```
cassiopeia.baseriotapi.set_rate_limit (calls_per_epoch, seconds_per_epoch)  
Sets the rate limit for cassiopeia to manage internally
```

Parameters

- **calls_per_epoch** (*int*) – the number of calls allowed in each epoch
- **seconds_per_epoch** (*int*) – the number of seconds per epoch

`cassiopeia.baseriotapi.set_rate_limits(*limits)`

Sets the rate limits for cassiopeia to manage internally

Parameters ***limits** (*tuple...*) – the rate limits to apply. Rate limits are of the form (calls_per_epoch, seconds_per_epoch)

`cassiopeia.baseriotapi.set_region(region)`

Set the region to run API queries against

Parameters **region** (*str*) – the region to query against

`cassiopeia.baseriotapi.set_tournament_api_key(key)`

Set your tournament API key

Parameters **key** (*str*) – the key to use

`cassiopeia.baseriotapi.set_tournament_rate_limit(calls_per_epoch, seconds_per_epoch)` *sec-*

Sets the tournament rate limit for cassiopeia to manage internally

Parameters

- **calls_per_epoch** (*int*) – the number of calls allowed in each epoch
- **seconds_per_epoch** (*int*) – the number of seconds per epoch

`cassiopeia.baseriotapi.set_tournament_rate_limits(*limits)`

Sets the tournament rate limits for cassiopeia to manage internally

Parameters ***limits** (*tuple...*) – the rate limits to apply. Rate limits are of the form (calls_per_epoch, seconds_per_epoch)

`cassiopeia.baseriotapi.update_tournament_code(tournament_code, parameters)`

<https://developer.riotgames.com/api/methods#!/1057/3647>

Parameters

- **tournament_code** (*str*) – the tournament code to update
- **parameters** (`TournamentCodeUpdateParameters`) – the new parameters for the tournament code

Submodules used by APIs

3.1 API Methods

3.1.1 cassiopeia.core

`cassiopeia.core.championapi.get_champion_status` (*champion*)

Gets the status for a champion (whether they are disabled, etc.)

Parameters `champion` (*Champion*) – the champion to get the status of

Returns the champion's status

Return type *ChampionStatus*

`cassiopeia.core.championapi.get_champion_statuses` (*free_to_play=False*)

Gets the statuses for all champions (whether they are disabled, etc.)

Parameters `free_to_play` (*bool*) – whether to only return free champions (default False)

Returns the statuses for all the champions

Return type dict<Champion, ChampionStatus>

`cassiopeia.core.championmasteryapi.get_champion_masteries` (*summoner*)

Gets all the ChampionMastery objects for the specified summoner

Parameters `summoner` (*Summoner*) – the summoner to get champion mastery for

Returns the summoner's champion masteries

Return type dict<Champion, ChampionMastery>

`cassiopeia.core.championmasteryapi.get_champion_mastery` (*summoner, champion*)

Gets the ChampionMastery object for the specified summoner and champion

Parameters

- `summoner` (*Summoner*) – the summoner to get champion mastery for
- `champion` (*Champion*) – the desired champion

Returns the summoner's champion mastery value for the specified champion

Return type *ChampionMastery*

`cassiopeia.core.championmasteryapi.get_champion_mastery_score` (*summoner*)

Gets the total champion mastery score for the specified summoner

Parameters `summoner` (*Summoner*) – the summoner to get champion mastery for

Returns the summoner's total champion mastery score

Return type `int`

`cassiopeia.core.championmasteryapi.get_top_champion_masteries` (*summoner*,
max_entries=3)

Gets the top ChampionMastery objects for the specified summoner

Parameters

- **summoner** (`Summoner`) – the summoner to get champion mastery for
- **max_entries** (`int`) – the maximum number of entries to retrieve (default 3)

Returns the summoner's top champion masteries

Return type `list<ChampionMastery>`

`cassiopeia.core.currentgameapi.get_current_game` (*summoner*)

Gets the game a summoner is currently in, if they're in one

Parameters **summoner** (`Summoner`) – the summoner to find an active game for

Returns the game they're in (or `None` if they aren't in one)

Return type `Game`

`cassiopeia.core.featuredgamesapi.get_featured_games` ()

Gets the current featured game list

Returns the featured games

Return type `list<Game>`

`cassiopeia.core.gameapi.get_recent_games` (*summoner*)

Gets the most recent games a summoner played

Parameters **summoner** (`Summoner`) – the summoner to get recent games for

Returns the summoner's recent games

Return type `list<Game>`

`cassiopeia.core.leagueapi.get_challenger` (*queue_type=<Queue.ranked_solo:*
'RANKED_SOLO_5x5'>)

Gets the challenger league

Parameters **queue_type** (`Queue`) – the queue to get the challenger league for (default `Queue.ranked_solo`)

Returns the challenger league for that queue

Return type `League`

`cassiopeia.core.leagueapi.get_league_entries_by_summoner` (*summoners*)

Gets the leagues that the summoner(s) belong(s) to, including only the requested summoner(s)' entries

Parameters **summoners** (`Summoner` | `list<Summoner>`) – the summoner(s) to get leagues for

Returns the leagues that the requested summoner(s) belong(s) to

Return type `list<League>` | `list<list<League>>`

`cassiopeia.core.leagueapi.get_league_entries_by_team` (*teams*)

Gets the leagues that the team(s) belong(s) to, including only the requested team(s)' entries

Parameters **teams** (`Team` | `list<Team>`) – the team(s) to get leagues for

Returns the leagues that the requested team(s) belong(s) to

Return type `list<League> | list<list<League>>`

`cassiopeia.core.leagueapi.get_leagues_by_summoner` (*summoners*)

Gets the leagues that the summoner(s) belong(s) to. You probably don't want to call this with `LoadPolicy.eager` set.

Parameters **summoners** (*Summoner* | *list<Summoner>*) – the summoner(s) to get leagues for

Returns the leagues that the requested summoner(s) belong(s) to

Return type `list<League> | list<list<League>>`

`cassiopeia.core.leagueapi.get_leagues_by_team` (*teams*)

Gets the leagues that the team(s) belong(s) to. You probably don't want to call this with `LoadPolicy.eager` set.

Parameters **teams** (*Team* | *list<Team>*) – the team(s) to get leagues for

Returns the leagues that the requested team(s) belong(s) to

Return type `list<League> | list<list<League>>`

`cassiopeia.core.leagueapi.get_master` (*queue_type=<Queue.ranked_solo: 'RANKED_SOLO_5x5'>*)

Gets the master league

Parameters **queue_type** (*Queue*) – the queue to get the master league for (default `Queue.ranked_solo`)

Returns the master league for that queue

Return type *League*

`cassiopeia.core.matchapi.get_match` (*id_, include_timeline=True, tournament_code=''*)

Gets a match

Parameters

- **id** (*int* | *MatchReference*) – the ID of or reference to the match to get
- **include_timeline** (*bool*) – whether to include timeline data in the returned match
- **tournament_code** (*str*) – the tournament code if the match to be retrieved is from a tournament

Returns the match

Return type *Match*

`cassiopeia.core.matchapi.get_matches` (*ids, include_timeline=True, tournament_code=''*)

Gets a bunch of matches

Parameters

- **ids** (*list<int>* | *list<MatchReference>*) – the IDs of or references to the matches to get
- **include_timeline** (*bool*) – whether to include timeline data in the returned matches
- **tournament_code** (*str*) – the tournament code if the match to be retrieved is from a tournament

Returns the matches

Return type `list<Match>`

`cassiopeia.core.matchapi.get_tournament_match_ids(tournament_code)`

Gets the IDs for a tournament's matches

Parameters `tournament_code` (*str*) – the tournament code

Returns the match ids for the tournament

Return type `list<int>`

`cassiopeia.core.matchlistapi.get_match_list(summoner, num_matches=0, begin_index=0, begin_time=0, end_time=0, champions=None, ranked_queues=None, seasons=None)`

Gets a summoner's match history

Parameters

- **summoner** (*Summoner*) – the summoner to get match history for
- **num_matches** (*int*) – the maximum number of matches to retrieve. 0 will get as many as possible. (default 0)
- **begin_index** (*int*) – the game index to start from (default 0)
- **begin_time** (*int* | *datetime*) – the begin time to use for fetching games (default 0)
- **end_time** (*int* | *datetime*) – the end time to use for fetching games (default 0)
- **champions** (*Champion* | *list<Champion>*) – the champion(s) to limit the results to (default None)
- **Queue** | **list<Queue>** (*ranked_queues*) – the ranked queue(s) to limit the results to (default None)
- **seasons** (*Season* | *list<Season>*) – the season(s) to limit the results to (default None)

Returns the summoner's match history

Return type `list<MatchReference>`

`cassiopeia.core.requests.call_with_ensured_size(method, max_size, arg)`

Breaks a list of arguments up into chunks of a maximum size and calls the given method on each chunk

Parameters

- **method** (*function*) – the method to call
- **max_size** (*int*) – the maximum number of arguments to include in a single call
- **arg** (*any* | *list<any>*) – the arguments to split up

Returns the combined results of the function calls on each chunk

Return type `list<any>` | `dict<any>`

`cassiopeia.core.staticdataapi.get_champion_by_id(id_)`

Gets a champion by ID

Parameters `id` (*int*) – the ID of the champion to get

Returns the champion

Return type *Champion*

`cassiopeia.core.staticdataapi.get_champion_by_name(name)`

Gets a champion by name

Parameters `name` (*str*) – the name of the champion to get

Returns the champion

Return type *Champion*

`cassiopeia.core.staticdataapi.get_champions()`

Gets all the champions

Returns all the champions

Return type `list<Champion>`

`cassiopeia.core.staticdataapi.get_champions_by_id(ids)`

Gets a bunch of champions by ID

Parameters `ids` (`list<int>`) – the IDs of the champions to get

Returns the requested champions

Return type `list<Champion>`

`cassiopeia.core.staticdataapi.get_champions_by_name(names)`

Gets a bunch of champions by name

Parameters `names` (`list<str>`) – the names of the champions to get

Returns the requested champions

Return type `list<Champion>`

`cassiopeia.core.staticdataapi.get_item(id_)`

Gets an item

Parameters `id` (*int*) – the ID of the item to get

Returns the item

Return type *Item*

`cassiopeia.core.staticdataapi.get_items(ids=None)`

Gets a bunch of items (or all of them)

Parameters `ids` (`list<int>`) – the IDs of the items to get (or `None` to get all items) (default `None`)

Returns the items

Return type `list<Item>`

`cassiopeia.core.staticdataapi.get_language_strings()`

Gets the locale-based string replacements for various game constants

Returns `dict<str, str>` the replacements

Return type `return`

`cassiopeia.core.staticdataapi.get_languages()`

Gets the valid locales (languages) that can be used with the API

Returns the valid locales

Return type `list<str>`

`cassiopeia.core.staticdataapi.get_map_information()`

Gets specific information about each map

Returns the map information

Return type `list<MapDetails>`

`cassiopeia.core.staticdataapi.get_masteries(ids=None)`

Gets a bunch of masteries (or all of them)

Parameters `ids` (`list<int>`) – the IDs of the masteries to get (or None to get all masteries) (default None)

Returns the masteries

Return type `list<Mastery>`

`cassiopeia.core.staticdataapi.get_mastery(id_)`

Gets a mastery

Parameters `id` (`int`) – the ID of the mastery to get

Returns the mastery

Return type *Mastery*

`cassiopeia.core.staticdataapi.get_realm()`

Gets the realm for the current region

Returns the realm

Return type *Realm*

`cassiopeia.core.staticdataapi.get_rune(id_)`

Gets a rune

Parameters `id` (`int`) – the ID of the rune to get

Returns the rune

Return type *Rune*

`cassiopeia.core.staticdataapi.get_runes(ids=None)`

Gets a bunch of runes (or all of them)

Parameters `ids` (`list<int>`) – the IDs of the runes to get (or None to get all runes) (default None)

Returns the runes

Return type `list<Rune>`

`cassiopeia.core.staticdataapi.get_summoner_spell(id_)`

Gets a summoner spell

Parameters `id` (`int`) – the ID of the summoner spell to get

Returns the summoner spell

Return type *SummonerSpell*

`cassiopeia.core.staticdataapi.get_summoner_spells(ids=None)`

Gets a bunch of summoner spells (or all of them)

Parameters `ids` (`list<int>`) – the IDs of the summoner spells to get (or None to get all summoner spells) (default None)

Returns the summoner spells

Return type `list<SummonerSpell>`

`cassiopeia.core.staticdataapi.get_versions()`

Gets the valid versions of the API

Returns the valid versions

Return type list<str>

`cassiopeia.core.statsapi.get_ranked_stats(summoner, season=None)`

Gets a summoner's ranked stats

Parameters

- **summoner** (*Summoner*) – the summoner to get ranked stats for
- **season** (*Season*) – the season to get ranked stats for (None will give current season stats) (default None)

Returns the summoner's ranked stats divided by champion. The entry for None contains combined stats for all champions.

Return type dict<Champion, AggregatedStats>

`cassiopeia.core.statsapi.get_stats(summoner, season=None)`

Gets a summoner's stats

Parameters

- **summoner** (*Summoner*) – the summoner to get stats for
- **season** (*Season*) – the season to get stats for (None will give current season stats) (default None)

Returns the summoner's stats divided by queue type

Return type dict<StatSummaryType, StatsSummary>

`cassiopeia.core.statusapi.get_shard()`

Gets the status of the current region's shard

Returns the status of the current region's shard

Return type *ShardStatus*

`cassiopeia.core.statusapi.get_shards()`

Get the list of server shards

Returns the shards

Return type list<Shard>

`cassiopeia.core.summonerapi.get_mastery_pages(summoners)`

Get the mastery pages for (a) summoner(s).

Parameters **ids** (*Summoner* | list<*Summoner*>) – the summoner(s) to get mastery pages for

Returns the requested summoner(s)' mastery pages

Return type list<MasteryPage> | list<list<MasteryPage>>

`cassiopeia.core.summonerapi.get_rune_pages(summoners)`

Get the rune pages for (a) summoner(s).

Parameters **ids** (*Summoner* | list<*Summoner*>) – the summoner(s) to get rune pages for

Returns the requested summoner(s)' rune pages

Return type list<RunePage> | list<list<RunePage>>

`cassiopeia.core.summonerapi.get_summoner_by_id(id_)`

Gets a summoner by ID

Parameters `id` (*int*) – the ID of the summoner

Returns the summoner

Return type *Summoner*

`cassiopeia.core.summonerapi.get_summoner_by_name(name)`

Gets a summoner by name

Parameters `name` (*str*) – the name of the summoner

Returns the summoner

Return type *Summoner*

`cassiopeia.core.summonerapi.get_summoner_name(id_)`

Gets the name of a summoner by ID

Parameters `id` (*id*) – the summoner's ID

Returns the summoner's name

Return type *str*

`cassiopeia.core.summonerapi.get_summoner_names(ids)`

Gets the names of a bunch of summoners by ID

Parameters `ids` (*list<id>*) – the summoners' IDs

Returns the summoners' names

Return type *list<str>*

`cassiopeia.core.summonerapi.get_summoners_by_id(ids)`

Gets a bunch of summoners by ID

Parameters `ids` (*list<int>*) – the IDs of the summoners

Returns the summoners

Return type *list<Summoner>*

`cassiopeia.core.summonerapi.get_summoners_by_name(names)`

Gets a bunch of summoners by name

Parameters `names` (*list<str>*) – the names of the summoners

Returns the summoners

Return type *list<Summoner>*

`cassiopeia.core.teamapi.get_team(id_)`

Gets a team by ID

id_ *str* the ID of the team

return *Team* the team

`cassiopeia.core.teamapi.get_teams(ids)`

Gets teams by ID

`ids` *list<str>* the IDs of the teams

return *list<Team>* the teams

`cassiopeia.core.teamapi.get_teams_by_summoner(summoners)`

Gets (a) summoner(s)' teams

`summoners` *Summoner | list<Summoner>* the summoner(s) to get teams for

return list<Team> | list<list<Team>> the summoner(s)' teams

`cassiopeia.core.tournamentapi.create_tournament(provider_id, name='')`

Creates a tournament

Parameters

- **provider_id** (*int*) – the provider ID to specify the regional registered provider data to associate this tournament
- **name** (*str*) – the optional name of the tournament (default “”)

Returns the tournament ID

Return type *int*

`cassiopeia.core.tournamentapi.create_tournament_codes(tournament_id, team_size, spectator_type, pick_type, map_type, allowed_summoners=[], meta_data='', count=1)`

Creates tournament codes for a tournament

Parameters

- **tournament_id** (*int*) – the tournament ID to generate codes for
- **team_size** (*int*) – the team size for the tournament (1-5)
- **spectator_type** (*str* | *SpectatorType*) – the spectator availability for the tournament
- **pick_type** (*str* | *PickType*) – the pick type for the tournament
- **map_type** (*str* | *MapType*) – the map the tournament is played on
- **allowed_summoners** (*list*<*Summoner*>) – the summoners who are allowed to participate in the tournament (default [])
- **meta_data** (*str* | *object*) – meta data to be included with the tournament. Any non-string value will be cast to a string. (default “”)
- **count** (*int*) – the number of codes to generate (max 1000) (default 1)

Returns the created tournament codes

Return type *list*<*str*>

`cassiopeia.core.tournamentapi.create_tournament_provider(region, url)`

Creates a tournament provider

Parameters

- **region** (*str* | *TournamentRegion*) – the region in which the provider will be running tournaments
- **url** (*str*) – the provider's callback URL to which tournament game results in this region should be posted. The URL must be well-formed, use the http or https protocol, and use the default port for the protocol (http URLs must use port 80, https URLs must use port 443).

Returns the tournament provider ID

Return type *int*

`cassiopeia.core.tournamentapi.get_lobby_events(tournament_code)`

Gets the lobby events that have occurred for the tournament code

Parameters `tournament_code` (*str* | *TournamentCode*) – the tournament code to get lobby events for

Returns the lobby events for that tournament code

Return type `list<LobbyEvent>`

`cassiopeia.core.tournamentapi.get_tournament_code(tournament_code)`

Gets information about the tournament code

Parameters `tournament_code` (*str*) – the tournament code

Returns the tournament code information

Return type *TournamentCode*

`cassiopeia.core.tournamentapi.update_tournament_code(tournament_code, allowed_summoners=[], spectator_type=None, pick_type=None, map_type=None)`

Updates a tournament code

Parameters

- **tournament_code** (*str* | *TournamentCode*) – the tournament code to update
- **allowed_summoners** (*list<Summoner>*) – the summoners who are allowed to participate in the tournament (default [])
- **spectator_type** (*str* | *SpectatorType*) – the spectator availability for the tournament (default None)
- **pick_type** (*str* | *PickType*) – the pick type for the tournament (default None)
- **map_type** (*str* | *MapType*) – the map the tournament is played on (default None)

3.1.2 cassiopeia.dto

`cassiopeia.dto.championapi.get_champion_status(id_)`
<https://developer.riotgames.com/api/methods#!/1015/3443>

Parameters `id` (*int*) – the ID of the champion to look up

Returns the champion

Return type *Champion*

`cassiopeia.dto.championapi.get_champion_statuses(freeToPlay=False)`
<https://developer.riotgames.com/api/methods#!/1015/3444>

Parameters `freeToPlay` (*bool*) – whether to only get free to play champions (default False)

Returns all the champions

Return type `list<Champion>`

`cassiopeia.dto.championmasteryapi.get_champion_masteries(summoner_id)`
<https://developer.riotgames.com/api/methods#!/1034/3544>

Parameters `summoner_id` (*int*) – the summoner ID to get champion masteries for

Returns the summoner's champion masteries

Return type list<ChampionMastery>

`cassiopeia.dto.championmasteryapi.get_champion_mastery` (*summoner_id*, *champion_id*)
<https://developer.riotgames.com/api/methods#!/1034/3545>

Parameters

- **summoner_id** (*int*) – the summoner ID to get champion mastery for
- **champion_id** (*int*) – the champion ID for the desired champion

Returns the summoner’s champion mastery value for the specified champion

Return type list<ChampionMastery>

`cassiopeia.dto.championmasteryapi.get_champion_mastery_score` (*summoner_id*)
<https://developer.riotgames.com/api/methods#!/1034/3546>

Parameters **summoner_id** (*int*) – the summoner ID to get champion masteries for

Returns the summoner’s total champion mastery score

Return type *int*

`cassiopeia.dto.championmasteryapi.get_top_champion_masteries` (*summoner_id*, *count=3*)
<https://developer.riotgames.com/api/methods#!/1034/3540>

Parameters

- **summoner_id** (*int*) – the summoner ID to get champion masteries for
- **count** (*int*) – the maximum number of entires to retrieve (default 3)

Returns the summoner’s top champion masteries

Return type list<ChampionMastery>

`cassiopeia.dto.currentgameapi.get_current_game` (*summoner_id*)
<https://developer.riotgames.com/api/methods#!/976/3336>

Parameters **summoner_id** (*int*) – the ID of the summoner to find an active game for

Returns the summoner’s current game (or None if they aren’t in one)

Return type *CurrentGameInfo*

`cassiopeia.dto.featuredgamesapi.get_featured_games` ()
<https://developer.riotgames.com/api/methods#!/977/3337>

Returns the current featured game list

Return type *FeaturedGames*

`cassiopeia.dto.gameapi.get_recent_games` (*summoner_id*)
<https://developer.riotgames.com/api/methods#!/1016/3445>

Parameters **summoner_id** (*int*) – the ID of the summoner to find recent games for

Returns the summoner’s recent games

Return type *RecentGames*

`cassiopeia.dto.leagueapi.get_challenger` (*queue_type*)
<https://developer.riotgames.com/api/methods#!/985/3353>

Parameters **queue_type** (*str*) – the queue type to get the challenger league for (“RANKED_SOLO_5x5”, “RANKED_TEAM_3x3”, “RANKED_TEAM_5x5”)

Returns the challenger league

Return type *League*

`cassiopeia.dto.leagueapi.get_league_entries_by_summoner(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/985/3356>

Parameters `summoner_ids` (*int* | *list*<*int*>) – the summoner ID(s) to get league entries for

Returns the summoner(s)' league entries

Return type `dict`<*str*, *list*<*League*>>

`cassiopeia.dto.leagueapi.get_league_entries_by_team(team_ids)`
<https://developer.riotgames.com/api/methods#!/985/3355>

Parameters `team_ids` (*str* | *list*<*str*>) – the team ID(s) to get league entries for

Returns the team(s)' league entries

Return type `dict`<*str*, *list*<*League*>>

`cassiopeia.dto.leagueapi.get_leagues_by_summoner(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/985/3351>

Parameters `summoner_ids` (*int* | *list*<*int*>) – the summoner ID(s) to get leagues for

Returns the summoner(s)' leagues

Return type `dict`<*str*, *list*<*League*>>

`cassiopeia.dto.leagueapi.get_leagues_by_team(team_ids)`
<https://developer.riotgames.com/api/methods#!/985/3352>

Parameters `team_ids` (*str* | *list*<*str*>) – the team ID(s) to get leagues for

Returns the team(s)' leagues

Return type `dict`<*str*, *list*<*League*>>

`cassiopeia.dto.leagueapi.get_master(queue_type)`
<https://developer.riotgames.com/api/methods#!/985/3354>

Parameters `queue_type` (*str*) – the queue type to get the master league for (“RANKED_SOLO_5x5”, “RANKED_TEAM_3x3”, “RANKED_TEAM_5x5”)

Returns the master league

Return type *League*

`cassiopeia.dto.matchapi.get_match(id_, include_timeline=True, tournament_code='')`
<https://developer.riotgames.com/api/methods#!/1014/3442>

Parameters

- `id` (*int*) – the ID of the match to get
- `include_timeline` (*bool*) – whether to include timeline data in the returned match
- `tournament_code` (*str*) – the tournament code if the match to be retrieved is from a tournament

Returns the match

Return type *MatchDetail*

`cassiopeia.dto.matchapi.get_tournament_match_ids(tournament_code)`
<https://developer.riotgames.com/api/methods#!/1058/3656>

Parameters `tournament_code` (*str*) – the tournament code

Returns the match ids for the tournament

Return type `list<int>`

`cassiopeia.dto.matchlistapi.get_match_list(summoner_id, num_matches=0, begin_index=0, begin_time=0, end_time=0, champion_ids=None, ranked_queues=None, seasons=None)`
<https://developer.riotgames.com/api/methods#!/1013/3439>

Parameters

- **summoner_id** (*int*) – the ID of the summoner to get the match history for
- **num_matches** (*int*) – the maximum number of matches to retrieve. 0 will get as many as possible. (default 0)
- **begin_index** (*int*) – the game index to start from (default 0)
- **begin_time** (*int*) – the begin time to use for fetching games specified as epoch milliseconds (default 0)
- **end_time** (*int*) – the end time to use for fetching games specified as epoch milliseconds (default 0)
- **champion_ids** (*int* | *list<int>*) – the champion ID(s) to limit the results to (default None)
- **ranked_queues** (*str* | *list<str>*) – the ranked queue(s) to limit the results to (“RANKED_SOLO_5x5”, “RANKED_TEAM_3x3”, “RANKED_TEAM_5x5”) (default None)
- **seasons** (*str* | *list<str>*) – the season(s) to limit the results to (“PRESEASON3”, “SEASON3”, “PRESEASON2014”, “SEASON2014”, “PRESEASON2015”, “SEASON2015”, “PRESEASON2016”, “SEASON2016”) (default None)

Returns the summoner’s match history

Return type *MatchList*

Handles making HTTP requests to the REST API and converting the results into a usable format

`cassiopeia.dto.requests.execute_request(url, method, payload='')`

Executes an HTTP request and returns the result in a string

Parameters

- **url** (*str*) – the full URL to send a request to
- **method** (*str*) – the HTTP method to use
- **payload** (*str*) – the json payload to send if appropriate for HTTP method (default “”)

Returns the content returned by the server

Return type *str*

`cassiopeia.dto.requests.get(request, params={}, static=False, include_base=True, tournament=False)`

```
cassiopeia.dto.requests.make_request(request, method, params={}, payload=None,
                                     static=False, include_base=True, tournament=False)
```

Makes a rate-limited HTTP request to the Riot API and returns the result

Parameters

- **request** (*str*) – the request string
- **method** (*str*) – the HTTP method to use
- **params** (*dict*<*str*, *any*>) – the path parameters to send with the request (default `{}`)
- **payload** (*CassiopeiaDto* | *CassiopeiaObject*) – the payload to send with the POST or PUT request (default `None`)
- **static** (*bool*) – whether this is a call to a static (non-rate-limited) API (default `False`)
- **include_base** (*bool*) – whether to prepend `https://{server}.api.pvp.net/api/lol/{region}/` to the request (default `True`)
- **tournament** (*bool*) – whether to use the tournament API rate limit (default `False`)

Returns the JSON response from the Riot API as a dict

Return type *dict*

```
cassiopeia.dto.requests.post(request, payload, params={}, include_base=True, tournament=False)
```

```
cassiopeia.dto.requests.put(request, payload, params={}, include_base=True, tournament=False)
```

```
cassiopeia.dto.staticdataapi.get_champion(id_)
https://developer.riotgames.com/api/methods#!/968/3322
```

Parameters **id** (*int*) – the ID of the champion to get

Returns the champion

Return type *Champion*

```
cassiopeia.dto.staticdataapi.get_champions()
https://developer.riotgames.com/api/methods#!/968/3326
```

Returns all the champions

Return type *ChampionList*

```
cassiopeia.dto.staticdataapi.get_item(id_)
https://developer.riotgames.com/api/methods#!/968/3319
```

Parameters **id** (*int*) – the ID of the item to get

Returns the item

Return type *Item*

```
cassiopeia.dto.staticdataapi.get_items()
https://developer.riotgames.com/api/methods#!/968/3314
```

Returns all the items

Return type *ItemList*

```
cassiopeia.dto.staticdataapi.get_language_strings()
https://developer.riotgames.com/api/methods#!/968/3316
```

Returns the locale-based string replacements for various game constants

Return type *LanguageStrings*

```
cassiopeia.dto.staticdataapi.get_languages()
https://developer.riotgames.com/api/methods#!/968/3324
```

Returns the valid locale strings for the API

Return type `list<str>`

```
cassiopeia.dto.staticdataapi.get_maps()
https://developer.riotgames.com/api/methods#!/968/3328
```

Returns specific information about each map

Return type *MapData*

```
cassiopeia.dto.staticdataapi.get_masteries()
https://developer.riotgames.com/api/methods#!/968/3317
```

Returns all the masteries

Return type *MasteryList*

```
cassiopeia.dto.staticdataapi.get_mastery(id_)
https://developer.riotgames.com/api/methods#!/968/3318
```

Parameters `id (int)` – the ID of the mastery to get

Returns the mastery

Return type *Mastery*

```
cassiopeia.dto.staticdataapi.get_realm()
https://developer.riotgames.com/api/methods#!/968/3325
```

Returns the realm for the current region

Return type *Realm*

```
cassiopeia.dto.staticdataapi.get_rune(id_)
https://developer.riotgames.com/api/methods#!/968/3321
```

Parameters `id (int)` – the ID of the rune to get

Returns the rune

Return type *Rune*

```
cassiopeia.dto.staticdataapi.get_runes()
https://developer.riotgames.com/api/methods#!/968/3315
```

Returns all the runes

Return type *RuneList*

```
cassiopeia.dto.staticdataapi.get_summoner_spell(id_)
https://developer.riotgames.com/api/methods#!/968/3320
```

Parameters `id (int)` – the ID of the summoner spell to get

Returns the summoner spell

Return type *SummonerSpell*

```
cassiopeia.dto.staticdataapi.get_summoner_spells()
https://developer.riotgames.com/api/methods#!/968/3327
```

Returns all the summoner spells

Return type *SummonerSpellList*

`cassiopeia.dto.staticdataapi.get_versions()`
<https://developer.riotgames.com/api/methods#!/968/3323>

Returns the valid API versions

Return type `list<str>`

`cassiopeia.dto.statsapi.get_ranked_stats(summoner_id, season=None)`
<https://developer.riotgames.com/api/methods#!/1018/3452>

Parameters

- **summoner_id** (*int*) – the summoner to get ranked stats for
- **season** (*str*) – the season to get ranked stats for (“SEASON2015”, “SEASON2014”, “SEASON3”) (default None)

Returns the ranked stats for the summoner and season specified

Return type *RankedStats*

`cassiopeia.dto.statsapi.get_stats(summoner_id, season=None)`
<https://developer.riotgames.com/api/methods#!/1018/3453>

Parameters

- **summoner_id** (*int*) – the summoner to get ranked stats for
- **season** (*str*) – the season to get ranked stats for (“SEASON2015”, “SEASON2014”, “SEASON3”) (default None)

Returns the ranked stats for the summoner and season specified

Return type *PlayerStatsSummaryList*

`cassiopeia.dto.statusapi.get_shard()`
<https://developer.riotgames.com/api/methods#!/908/3142>

Returns the status of the current region’s shard

Return type *ShardStatus*

`cassiopeia.dto.statusapi.get_shards()`
<https://developer.riotgames.com/api/methods#!/908/3143>

Returns the shards (unfortunately neither Crystal nor Kirby’s)

Return type `list<Shard>`

`cassiopeia.dto.summonerapi.get_summoner_masteryes(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3450>

Parameters **summoner_ids** (*int* | *list<int>*) – the summoner ID(s) to get mastery pages for

Returns the requested summoners’ mastery pages

Return type `dict<str, MasteryPages>`

`cassiopeia.dto.summonerapi.get_summoner_names(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3451>

Parameters **summoner_ids** (*int* | *list<int>*) – the summoner ID(s) to get names for

Returns the requested summoners’ names

Return type dict<str, str>

`cassiopeia.dto.summonerapi.get_summoner_runes(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3449>

Parameters `summoner_ids` (*int* | *list*<*int*>) – the summoner ID(s) to get rune pages for

Returns the requested summoners' rune pages

Return type dict<str, RunePages>

`cassiopeia.dto.summonerapi.get_summoners_by_id(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/1017/3447>

Parameters `summoner_ids` (*int* | *list*<*int*>) – the summoner ID(s) to look up

Returns the requested summoners

Return type dict<str, Summoner>

`cassiopeia.dto.summonerapi.get_summoners_by_name(summoner_names)`
<https://developer.riotgames.com/api/methods#!/1017/3446>

Parameters `summoner_names` (*str* | *list*<*str*>) – the summoner name(s) to look up

Returns the requested summoners

Return type dict<str, Summoner>

`cassiopeia.dto.teamapi.get_teams_by_id(team_ids)`
<https://developer.riotgames.com/api/methods#!/986/3358>

Parameters `team_ids` (*str* | *list*<*str*>) – the team ID(s) to look up

Returns the requested teams

Return type dict<str, Team>

`cassiopeia.dto.teamapi.get_teams_by_summoner_id(summoner_ids)`
<https://developer.riotgames.com/api/methods#!/986/3358>

Parameters `summoner_ids` (*int* | *list*<*int*>) – the summoner ID(s) to look up teams for

Returns the requested summoners' teams

Return type dict<str, list<Team>>

`cassiopeia.dto.tournamentapi.create_tournament(parameters)`
<https://developer.riotgames.com/api/methods#!/1057/3649>

Parameters `parameters` (`TournamentRegistrationParameters`) – the parameters for the tournament

Returns the tournament ID

Return type `int`

`cassiopeia.dto.tournamentapi.create_tournament_codes(tournament_id, parameters, count=1)`
<https://developer.riotgames.com/api/methods#!/1063>

Parameters

- `tournament_id` (*int*) – the tournament ID to generate codes for
- `parameters` (`TournamentCodeParameters`) – the parameters for the tournament codes
- `count` (*int*) – the number of codes to generate (max 1000) (default 1)

Returns the created tournament codes

Return type `list<str>`

`cassiopeia.dto.tournamentapi.create_tournament_provider(parameters)`
<https://developer.riotgames.com/api/methods#!/1057/3646>

Parameters `parameters` (`ProviderRegistrationParameters`) – the parameters for the provider

Returns the provider ID

Return type `int`

`cassiopeia.dto.tournamentapi.get_lobby_events(tournament_code)`
<https://developer.riotgames.com/api/methods#!/1057/3653>

Parameters `tournament_code` (`str`) – the tournament code to get lobby events for

Returns the lobby events for that tournament code

Return type `LobbyEventWrapper`

`cassiopeia.dto.tournamentapi.get_tournament_code(tournament_code)`
<https://developer.riotgames.com/api/methods#!/1057/3643>

Parameters `tournament_code` (`str`) – the tournament code to get information about

Returns information about the tournament code

Return type `TournamentCode`

`cassiopeia.dto.tournamentapi.update_tournament_code(tournament_code, parameters)`
<https://developer.riotgames.com/api/methods#!/1057/3647>

Parameters

- `tournament_code` (`str`) – the tournament code to update
- `parameters` (`TournamentCodeUpdateParameters`) – the new parameters for the tournament code

3.2 API Class Definitions

3.2.1 cassiopeia.type.api

exception `cassiopeia.type.api.exception.APIError` (`message`, `error_code`)
Bases: `Exception`

Parameters

- `message` (`str`) – the error message
- `error_code` (`int`) – the HTTP error code that was received

args

with_traceback (`tb`)

Exception.with_traceback(`tb`) – set self.__traceback__ to `tb` and return self.

exception `cassiopeia.type.api.exception.CassiopeiaException`
Bases: `Exception`

Generic exception for a failure within Cassiopeia

args

with_traceback()
Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

class cassiopeia.type.api.rates.**MultiRateLimiter**(*limits)
Bases: `object`

Resets the rate limit

call(method=None, *args)
Waits until a call becomes available

calls
Returns the number of successful calls (no exceptions in the call) and total calls served by this limiter

Returns tuple A (successful calls, total calls) tuple

Return type return

reset_in(seconds)
Resets the rate limiter after waiting

Parameters **seconds** (`int`) – the number of seconds to wait before resetting

wait()
Resets the rate limiter after waiting

Parameters **seconds** (`int`) – the number of seconds to wait before resetting

class cassiopeia.type.api.rates.**SingleRateLimiter**(calls_per_epoch, seconds_per_epoch)
Bases: `object`

Handles a single rate limit, ensuring that calls don't exceed it

call(method=None, *args)

Parameters

- **calls_per_epoch** (`int`) – the number of calls allowed in each epoch
- **seconds_per_epoch** (`int`) – the number of seconds per epoch

calls
Drains all remaining calls

reset_in(seconds)
Calls a function when the rate limit allows (first come first serve)

Parameters

- **method** (`function`) – the function which will be called when the rate limit allows
- ***args** (`any...`) – the arguments to be passed to the functions when it is called

Returns the result of the function once it has been called

Return type `any`

wait()
Calls a function when the rate limit allows (first come first serve)

Parameters

- **method** (`function`) – the function which will be called when the rate limit allows
- ***args** (`any...`) – the arguments to be passed to the functions when it is called

Returns the result of the function once it has been called

Return type `any`

class `cassiopeia.type.api.store.Cache`

Bases: `cassiopeia.type.api.store.DataStore`

Stores objects in the data store

Parameters

- **objs** (`any`) – the objects to store
- **keys** (`any` | `list<any>`) – the keys to store those values with
- **complete_sets** (`list<type>`) – include any types for which it should be marked that all possible values are stored

get (`class_`, `keys`, `key_field`)

Gets an iterator over all currently stored values for a type

Parameters **class** (`type`) – the class to get values for

Returns and iterator over all stored values for the type

Return type `iterator<class_>`

get_all (`class_`)

Checks if the data store has all the values for a type there can be (as reported by the user)

Parameters **class** (`type`) – the class to do the check for

Returns whether all the values are stored

Return type `bool`

has_all (`class_`)

A place to store data. Used for caching/storing data from API calls

iterate (`class_`)

Gets all currently stored values for a type

Parameters **class** (`type`) – the class to get values for

Returns all stored values for the type

Return type `list<class_>`

store (`objs`, `keys`, `complete_sets=[]`)

Gets objects from the data store

Parameters

- **class** (`type`) – the class to get values for
- **keys** (`any` | `list<any>`) – the keys that should be used to find the desired values
- **key_field** (`str`) – the name of the attribute that the key(s) reference

Returns the values from storage - None will replace any value that couldn't be found

Return type `class_` | `list<class_>`

class `cassiopeia.type.api.store.DataStore`

Bases: `object`

A place to store data. Used for caching/storing data from API calls

get (*class_*, *keys*, *key_field*)
 Gets an iterator over all currently stored values for a type

Parameters **class** (*type*) – the class to get values for

Returns and iterator over all stored values for the type

Return type iterator<**class_**>

get_all (*class_*)
 Checks if the data store has all the values for a type there can be (as reported by the user)

Parameters **class** (*type*) – the class to do the check for

Returns whether all the values are stored

Return type **bool**

has_all (*class_*)
 A place to store data. Used for caching/storing data from API calls

iterate (*class_*)
 Gets all currently stored values for a type

Parameters **class** (*type*) – the class to get values for

Returns all stored values for the type

Return type list<**class_**>

store (*objs*, *keys*, *complete_sets*=[])
 Gets objects from the data store

Parameters

- **class** (*type*) – the class to get values for
- **keys** (*any* | list<*any*>) – the keys that should be used to find the desired values
- **key_field** (*str*) – the name of the attribute that the key(s) reference

Returns the values from storage - None will replace any value that couldn't be found

Return type **class_** | list<**class_**>

class cassiopeia.type.api.store.**HasAllStatus** (*class_*, *have_all*=True)
 Bases: sqlalchemy.ext.declarative.api.Base

class_

static get_name (*class_*)

have_all

metadata = **MetaData**(bind=None)

class cassiopeia.type.api.store.**SQLAlchemyDB** (*flavor*, *host*, *database*, *username*, *password*)
 Bases: *cassiopeia.type.api.store.DataStore*

class **Iterator** (*class_*, *result*)
 Bases: *object*

SQLAlchemyDB.close ()
 A mock cache that doesn't actually store anything

SQLAlchemyDB.get (*class_*, *keys*, *key_field*)
 Gets an iterator over all currently stored values for a type

Parameters `class (type)` – the class to get values for

Returns and iterator over all stored values for the type

Return type `iterator<class_>`

`SQLAlchemyDB.get_all (class_)`

Checks if the data store has all the values for a type there can be (as reported by the user)

Parameters `class (type)` – the class to do the check for

Returns whether all the values are stored

Return type `bool`

`SQLAlchemyDB.has_all (class_)`

A place to store data. Used for caching/storing data from API calls

`SQLAlchemyDB.iterate (class_)`

Gets all currently stored values for a type

Parameters `class (type)` – the class to get values for

Returns all stored values for the type

Return type `list<class_>`

`SQLAlchemyDB.store (objs, keys=None, complete_sets=[])`

Gets objects from the data store

Parameters

- **class (type)** – the class to get values for
- **keys (any | list<any>)** – the keys that should be used to find the desired values
- **key_field (str)** – the name of the attribute that the key(s) reference

Returns the values from storage - None will replace any value that couldn't be found

Return type `class_ | list<class_>`

`class cassiopeia.type.api.store.VoidDataStore`

Bases: `cassiopeia.type.api.store.DataStore`

Stores objects in the data store

Parameters

- **objs (any)** – the objects to store
- **keys (any | list<any>)** – the keys to store those values with
- **complete_sets (list<type>)** – include any types for which it should be marked that all possible values are stored

`get (class_, keys, key_field)`

Gets an iterator over all currently stored values for a type

Parameters `class (type)` – the class to get values for

Returns and iterator over all stored values for the type

Return type `iterator<class_>`

`get_all (class_)`

Checks if the data store has all the values for a type there can be (as reported by the user)

Parameters `class (type)` – the class to do the check for

Returns whether all the values are stored

Return type `bool`

has_all (*class_*)

A place to store data. Used for caching/storing data from API calls

iterate (*class_*)

Gets all currently stored values for a type

Parameters **class** (*type*) – the class to get values for

Returns all stored values for the type

Return type `list<class_>`

store (*objs*, *keys*, *complete_sets*=[])

Gets objects from the data store

Parameters

- **class** (*type*) – the class to get values for
- **keys** (*any* | *list<any>*) – the keys that should be used to find the desired values
- **key_field** (*str*) – the name of the attribute that the key(s) reference

Returns the values from storage - None will replace any value that couldn't be found

Return type `class_` | `list<class_>`

3.2.2 cassiopeia.type.core

class `cassiopeia.type.core.champion.ChampionStatus` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

champion

Returns the Champion this status is for

Return type `Champion`

coop_ai_enabled

Returns whether the champion is currently enabled for coop vs ai games

Return type `bool`

custom_enabled

Returns whether the champion is currently enabled for custom games

Return type `bool`

dto_type

alias of `Champion`

enabled

Returns whether the champion is currently enabled

Return type `bool`

free

Returns whether the champion is currently free this week

Return type `bool`

ranked_enabled

Returns whether the champion is currently enabled for ranked games

Return type `bool`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.championmastery.ChampionMastery` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

champion

Returns champion for this entry

Return type `Champion`

chest_granted

Returns is chest granted for this champion or not in current season

Return type `bool`

dto_type

alias of `ChampionMastery`

last_played

Returns last time this champion was played by this player

Return type `datetime`

level

Returns champion level for specified player and champion combination

Return type `int`

points

Returns total number of champion points for this player and champion combination - they are used to determine `champion_level`

Return type `int`

points_since_last_level

Returns number of points earned since current level has been achieved. Zero if player reached maximum champion level for this champion.

Return type `int`

points_until_next_level

Returns number of points needed to achieve next level. Zero if player reached maximum champion level for this champion.

Return type `int`

summoner

Returns the player this mastery information is for

Return type `Summoner`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

tokens_earned

Returns number of tokens earned to next level mastery

Return type `int`

class `cassiopeia.type.core.common.Ascended`

Bases: `enum.Enum`

death = `<Ascended.death: 'CLEAR_ASCENDED'>`

npc = `<Ascended.npc: 'MINION_ASCENDED'>`

player = `<Ascended.player: 'CHAMPION_ASCENDED'>`

class `cassiopeia.type.core.common.Building`

Bases: `enum.Enum`

inhibitor = `<Building.inhibitor: 'INHIBITOR_BUILDING'>`

turret = `<Building.turret: 'TOWER_BUILDING'>`

class `cassiopeia.type.core.common.CassiopeiaObject` (*data*)

Bases: `object`

An object storing data from the API, with various helpful utilities and shortcuts

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.common.Division`

Bases: `enum.Enum`

five = `<Division.five: 'V'>`

four = `<Division.four: 'IV'>`

one = `<Division.one: 'I'>`

three = `<Division.three: 'III'>`

two = `<Division.two: 'II'>`

class `cassiopeia.type.core.common.EventType`

Bases: `enum.Enum`

ascension = `<EventType.ascension: 'ASCENDED_EVENT'>`

building_kill = `<EventType.building_kill: 'BUILDING_KILL'>`

elite_monster_kill = `<EventType.elite_monster_kill: 'ELITE_MONSTER_KILL'>`

item_destruction = `<EventType.item_destruction: 'ITEM_DESTROYED'>`

item_purchase = `<EventType.item_purchase: 'ITEM_PURCHASED'>`

item_sale = `<EventType.item_sale: 'ITEM_SOLD'>`

item_undo = `<EventType.item_undo: 'ITEM_UNDO'>`

kill = `<EventType.kill: 'CHAMPION_KILL'>`

point_capture = `<EventType.point_capture: 'CAPTURE_POINT'>`

skill_up = `<EventType.skill_up: 'SKILL_LEVEL_UP'>`

summoning = `<EventType.summoning: 'PORO_KING_SUMMON'>`

ward_kill = `<EventType.ward_kill: 'WARD_KILL'>`

```
ward_placement = <EventType.ward_placement: 'WARD_PLACED'>

class cassiopeia.type.core.common.GameMode
    Bases: enum.Enum

    aram = <GameMode.aram: 'ARAM'>
    ascension = <GameMode.ascension: 'ASCENSION'>
    classic = <GameMode.classic: 'CLASSIC'>
    dominion = <GameMode.dominion: 'ODIN'>
    nexus_siege = <GameMode.nexus_siege: 'SIEGE'>
    one_for_all = <GameMode.one_for_all: 'ONEFORALL'>
    poro_king = <GameMode.poro_king: 'KINGPORO'>
    showdown = <GameMode.showdown: 'FIRSTBLOOD'>
    tutorial = <GameMode.tutorial: 'TUTORIAL'>

class cassiopeia.type.core.common.GameType
    Bases: enum.Enum

    custom = <GameType.custom: 'CUSTOM_GAME'>
    matched = <GameType.matched: 'MATCHED_GAME'>
    tutorial = <GameType.tutorial: 'TUTORIAL_GAME'>

class cassiopeia.type.core.common.Lane
    Bases: enum.Enum

    bot_lane = <Lane.bot_lane: 'BOTTOM'>
    jungle = <Lane.jungle: 'JUNGLE'>
    mid_lane = <Lane.mid_lane: 'MIDDLE'>
    top_lane = <Lane.top_lane: 'TOP'>

class cassiopeia.type.core.common.LaneType
    Bases: enum.Enum

    bot_lane = <LaneType.bot_lane: 'BOT_LANE'>
    mid_lane = <LaneType.mid_lane: 'MID_LANE'>
    top_lane = <LaneType.top_lane: 'TOP_LANE'>

class cassiopeia.type.core.common.LazyProperty (method)
    Bases: object

class cassiopeia.type.core.common.LevelUp
    Bases: enum.Enum

    evolution = <LevelUp.evolution: 'EVOLVE'>
    normal = <LevelUp.normal: 'NORMAL'>

class cassiopeia.type.core.common.LoadPolicy
    Bases: enum.Enum

    eager = <LoadPolicy.eager: 'EAGER'>
    lazy = <LoadPolicy.lazy: 'LAZY'>
```

```

class cassiopeia.type.core.common.Map
    Bases: enum.Enum

    butchers_bridge = <Map.butchers_bridge: 14>
    howling_abyss = <Map.howling_abyss: 12>
    summoners_rift = <Map.summoners_rift: 11>
    summoners_rift_autumn = <Map.summoners_rift_autumn: 2>
    summoners_rift_summer = <Map.summoners_rift_summer: 1>
    the_crystal_scar = <Map.the_crystal_scar: 8>
    the_proving_grounds = <Map.the_proving_grounds: 3>
    twisted_treeline = <Map.twisted_treeline: 10>
    twisted_treeline_original = <Map.twisted_treeline_original: 4>

class cassiopeia.type.core.common.MasteryType
    Bases: enum.Enum

    cunning = <MasteryType.cunning: 'Cunning'>
    ferocity = <MasteryType.ferocity: 'Ferocity'>
    resolve = <MasteryType.resolve: 'Resolve'>

class cassiopeia.type.core.common.Monster
    Bases: enum.Enum

    baron = <Monster.baron: 'BARON_NASHOR'>
    blue = <Monster.blue: 'BLUE_GOLEM'>
    dragon = <Monster.dragon: 'DRAGON'>
    red = <Monster.red: 'RED_LIZARD'>
    rift_herald = <Monster.rift_herald: 'RIFTHERALD'>
    spider = <Monster.spider: 'VILEMAW'>

class cassiopeia.type.core.common.Platform
    Bases: enum.Enum

    brazil = <Platform.brazil: 'BR1'>
    europe_north_east = <Platform.europe_north_east: 'EUN1'>
    europe_west = <Platform.europe_west: 'EUW1'>
    korea = <Platform.korea: 'KR'>
    latin_america_north = <Platform.latin_america_north: 'LA1'>
    latin_america_south = <Platform.latin_america_south: 'LA2'>
    north_america = <Platform.north_america: 'NA1'>
    oceania = <Platform.oceania: 'OC1'>
    russia = <Platform.russia: 'RU'>
    turkey = <Platform.turkey: 'TR1'>

class cassiopeia.type.core.common.Point
    Bases: enum.Enum

```

```
boneyard = <Point.boneyard: 'POINT_E'>
drill = <Point.drill: 'POINT_D'>
quarry = <Point.quarry: 'POINT_A'>
refinery = <Point.refinery: 'POINT_B'>
windmill = <Point.windmill: 'POINT_C'>

class cassiopeia.type.core.common.Queue
    Bases: enum.Enum

    aram = <Queue.aram: 'ARAM_5x5'>
    ascension = <Queue.ascension: 'ASCENSION_5x5'>
    black_market = <Queue.black_market: 'BILGEWATER_5x5'>
    bot_beginner_fives = <Queue.bot_beginner_fives: 'BOT_5x5_BEGINNER'>
    bot_dominion = <Queue.bot_dominion: 'BOT_ODIN_5x5'>
    bot_fives = <Queue.bot_fives: 'BOT_5x5'>
    bot_intermediate_fives = <Queue.bot_intermediate_fives: 'BOT_5x5_INTERMEDIATE'>
    bot_intro_fives = <Queue.bot_intro_fives: 'BOT_5x5_INTRO'>
    bot_threes = <Queue.bot_threes: 'BOT_TT_3x3'>
    bot_urf = <Queue.bot_urf: 'BOT_URF_5x5'>
    butchers_bridge = <Queue.butchers_bridge: 'BILGEWATER_ARAM_5x5'>
    custom = <Queue.custom: 'CUSTOM'>
    definitely_not_dominion = <Queue.definitely_not_dominion: 'DEFINITELY_NOT_DOMINION_5x5'>
    dominion_blind = <Queue.dominion_blind: 'ODIN_5x5_BLIND'>
    dominion_draft = <Queue.dominion_draft: 'ODIN_5x5_DRAFT'>
    doom_bots_1 = <Queue.doom_bots_1: 'NIGHTMARE_BOT_5x5_RANK1'>
    doom_bots_2 = <Queue.doom_bots_2: 'NIGHTMARE_BOT_5x5_RANK2'>
    doom_bots_5 = <Queue.doom_bots_5: 'NIGHTMARE_BOT_5x5_RANK5'>
    dynamic_queue = <Queue.dynamic_queue: 'TEAM_BUILDER_DRAFT_UNRANKED_5x5'>
    flex = <Queue.flex: 'RANKED_FLEX_SR'>
    hexakill_summoners_rift = <Queue.hexakill_summoners_rift: 'SR_6x6'>
    hexakill_twisted_treeline = <Queue.hexakill_twisted_treeline: 'HEXAKILL'>
    nemesis_draft = <Queue.nemesis_draft: 'COUNTER_PICK'>
    nexus_siege = <Queue.nexus_siege: 'SIEGE'>
    normal_blind_fives = <Queue.normal_blind_fives: 'NORMAL_5x5_BLIND'>
    normal_blind_threes = <Queue.normal_blind_threes: 'NORMAL_3x3'>
    normal_draft_fives = <Queue.normal_draft_fives: 'NORMAL_5x5_DRAFT'>
    one_for_all = <Queue.one_for_all: 'ONEFORALL_5x5'>
    one_for_all_mirror = <Queue.one_for_all_mirror: 'ONEFORALL_MIRRORMODE_5x5'>
```

```

poro_king = <Queue.poro_king: 'KING_PORO_5x5'>
random_urf = <Queue.random_urf: 'ARURF_5X5'>
ranked_dynamic_queue = <Queue.ranked_dynamic_queue: 'TEAM_BUILDER_DRAFT_RANKED_5x5'>
ranked_fives = <Queue.ranked_fives: 'RANKED_TEAM_5x5'>
ranked_premade_fives = <Queue.ranked_premade_fives: 'RANKED_PREMADE_5x5'>
ranked_premade_threes = <Queue.ranked_premade_threes: 'RANKED_PREMADE_3x3'>
ranked_solo = <Queue.ranked_solo: 'RANKED_SOLO_5x5'>
ranked_threes = <Queue.ranked_threes: 'RANKED_TEAM_3x3'>
showdown_duo = <Queue.showdown_duo: 'FIRSTBLOOD_2x2'>
showdown_solo = <Queue.showdown_solo: 'FIRSTBLOOD_1x1'>
team_builder = <Queue.team_builder: 'GROUP_FINDER_5x5'>
urf = <Queue.urf: 'URF_5x5'>

class cassiopeia.type.core.common.Region
    Bases: enum.Enum

    brazil = <Region.brazil: 'br'>
    europe_north_east = <Region.europe_north_east: 'eune'>
    europe_west = <Region.europe_west: 'euw'>
    japan = <Region.japan: 'jp'>
    korea = <Region.korea: 'kr'>
    latin_america_north = <Region.latin_america_north: 'lan'>
    latin_america_south = <Region.latin_america_south: 'las'>
    north_america = <Region.north_america: 'na'>
    oceania = <Region.oceania: 'oce'>
    pbe = <Region.pbe: 'pbe'>
    russia = <Region.russia: 'ru'>
    turkey = <Region.turkey: 'tr'>

class cassiopeia.type.core.common.Role
    Bases: enum.Enum

    carry = <Role.carry: 'DUO_CARRY'>
    duo = <Role.duo: 'DUO'>
    none = <Role.none: 'NONE'>
    solo = <Role.solo: 'SOLO'>
    support = <Role.support: 'DUO_SUPPORT'>

class cassiopeia.type.core.common.Season
    Bases: enum.Enum

    preseason_3 = <Season.preseason_3: 'PRESEASON3'>
    preseason_4 = <Season.preseason_4: 'PRESEASON2014'>

```

```
preseason_5 = <Season.preseason_5: 'PRESEASON2015'>
preseason_6 = <Season.preseason_6: 'PRESEASON2016'>
preseason_7 = <Season.preseason_7: 'PRESEASON2017'>
season_3 = <Season.season_3: 'SEASON3'>
season_4 = <Season.season_4: 'SEASON2014'>
season_5 = <Season.season_5: 'SEASON2015'>
season_6 = <Season.season_6: 'SEASON2016'>
season_7 = <Season.season_7: 'SEASON2017'>

class cassiopeia.type.core.common.Side
    Bases: enum.Enum

    blue = <Side.blue: 100>
    red = <Side.red: 200>

class cassiopeia.type.core.common.StatSummaryType
    Bases: enum.Enum

    aram = <StatSummaryType.aram: 'AramUnranked5x5'>
    ascension = <StatSummaryType.ascension: 'Ascension'>
    black_market = <StatSummaryType.black_market: 'Bilgewater'>
    bot_fives = <StatSummaryType.bot_fives: 'CoopVsAI'>
    bot_threes = <StatSummaryType.bot_threes: 'CoopVsAI3x3'>
    bot_urf = <StatSummaryType.bot_urf: 'URFBots'>
    dominion = <StatSummaryType.dominion: 'OdinUnranked'>
    doom_bots = <StatSummaryType.doom_bots: 'NightmareBot'>
    flex_summoners_rift = <StatSummaryType.flex_summoners_rift: 'RankedFlexSR'>
    flex_twisted_treeline = <StatSummaryType.flex_twisted_treeline: 'RankedFlexTT'>
    hexakill_summoners_rift = <StatSummaryType.hexakill_summoners_rift: 'SummonersRift6x6'>
    hexakill_twisted_treeline = <StatSummaryType.hexakill_twisted_treeline: 'Hexakill'>
    nemesis_draft = <StatSummaryType.nemesis_draft: 'CounterPick'>
    nexus_siege = <StatSummaryType.nexus_siege: 'Siege'>
    normal_fives = <StatSummaryType.normal_fives: 'Unranked'>
    normal_threes = <StatSummaryType.normal_threes: 'Unranked3x3'>
    one_for_all = <StatSummaryType.one_for_all: 'OneForAll5x5'>
    poro_king = <StatSummaryType.poro_king: 'KingPoro'>
    ranked_fives = <StatSummaryType.ranked_fives: 'RankedTeam5x5'>
    ranked_premade_fives = <StatSummaryType.ranked_premade_fives: 'RankedPremade5x5'>
    ranked_premade_threes = <StatSummaryType.ranked_premade_threes: 'RankedPremade3x3'>
    ranked_solo = <StatSummaryType.ranked_solo: 'RankedSolo5x5'>
    ranked_threes = <StatSummaryType.ranked_threes: 'RankedTeam3x3'>
```

```

    showdown_duo = <StatSummaryType.showdown_duo: 'FirstBlood2x2'>
    showdown_solo = <StatSummaryType.showdown_solo: 'FirstBlood1x1'>
    team_builder = <StatSummaryType.team_builder: 'CAP5x5'>
    urf = <StatSummaryType.urf: 'URF'>
class cassiopeia.type.core.common.SubType
    Bases: enum.Enum

    aram = <SubType.aram: 'ARAM_UNRANKED_5x5'>
    ascension = <SubType.ascension: 'ASCENSION'>
    black_market = <SubType.black_market: 'BILGEWATER'>
    bot_fives = <SubType.bot_fives: 'BOT'>
    bot_threes = <SubType.bot_threes: 'BOT_3x3'>
    bot_urf = <SubType.bot_urf: 'URF_BOT'>
    custom = <SubType.custom: 'NONE'>
    dominion = <SubType.dominion: 'ODIN_UNRANKED'>
    doom_bots = <SubType.doom_bots: 'NIGHTMARE_BOT'>
    flex = <SubType.flex: 'RANKED_FLEX_SR'>
    hexakill_summoners_rift = <SubType.hexakill_summoners_rift: 'SR_6x6'>
    hexakill_twisted_treeline = <SubType.hexakill_twisted_treeline: 'HEXAKILL'>
    nemesis_draft = <SubType.nemesis_draft: 'COUNTER_PICK'>
    nexus_siege = <SubType.nexus_siege: 'SIEGE'>
    normal_fives = <SubType.normal_fives: 'NORMAL'>
    normal_threes = <SubType.normal_threes: 'NORMAL_3x3'>
    one_for_all = <SubType.one_for_all: 'ONEFORALL_5x5'>
    poro_king = <SubType.poro_king: 'KING_PORO'>
    ranked_fives = <SubType.ranked_fives: 'RANKED_TEAM_5x5'>
    ranked_solo = <SubType.ranked_solo: 'RANKED_SOLO_5x5'>
    ranked_threes = <SubType.ranked_threes: 'RANKED_TEAM_3x3'>
    showdown_duo = <SubType.showdown_duo: 'FIRSTBLOOD_2x2'>
    showdown_solo = <SubType.showdown_solo: 'FIRSTBLOOD_1x1'>
    team_builder = <SubType.team_builder: 'CAP_5x5'>
    urf = <SubType.urf: 'URF'>
class cassiopeia.type.core.common.Tier
    Bases: enum.Enum

    bronze = <Tier.bronze: 'BRONZE'>
    challenger = <Tier.challenger: 'CHALLENGER'>
    diamond = <Tier.diamond: 'DIAMOND'>
    gold = <Tier.gold: 'GOLD'>

```

```
master = <Tier.master: 'MASTER'>
platinum = <Tier.platinum: 'PLATINUM'>
silver = <Tier.silver: 'SILVER'>
unranked = <Tier.unranked: 'UNRANKED'>

class cassiopeia.type.core.common.Turret
    Bases: enum.Enum

    fountain = <Turret.fountain: 'FOUNTAIN_TURRET'>
    inhibitor = <Turret.inhibitor: 'BASE_TURRET'>
    inner = <Turret.inner: 'INNER_TURRET'>
    nexus = <Turret.nexus: 'NEXUS_TURRET'>
    outer = <Turret.outer: 'OUTER_TURRET'>
    undefined = <Turret.undefined: 'UNDEFINED_TURRET'>

class cassiopeia.type.core.common.Ward
    Bases: enum.Enum

    blue_trinket = <Ward.blue_trinket: 'BLUE_TRINKET'>
    mushroom = <Ward.mushroom: 'TEEMO_MUSHROOM'>
    sight = <Ward.sight: 'SIGHT_WARD'>
    undefined = <Ward.undefined: 'UNDEFINED'>
    upgraded_yellow_trinket = <Ward.upgraded_yellow_trinket: 'YELLOW_TRINKET_UPGRADE'>
    vision = <Ward.vision: 'VISION_WARD'>
    yellow_trinket = <Ward.yellow_trinket: 'YELLOW_TRINKET'>

class cassiopeia.type.core.common.immutablemethod(method)
    Bases: object

    Makes a property load only once and store the result value to be returned to all later calls

    Parameters method (function) – the method to turn into a lazy property

    Returns the method as a lazy property

    Return type function

cassiopeia.type.core.common.inheritdocs(class_)

    Parameters method (function) – the method to make immutable

cassiopeia.type.core.common.lazyproperty(method)

    Parameters method (function) – the method to turn into a lazy property

class cassiopeia.type.core.currentgame.Ban(data)
    Bases: cassiopeia.type.core.common.CassiopeiaObject

    champion

        Returns which side banned this champion

        Return type Side

    dto_type
        alias of BannedChampion
```


pick_turn

Returns which pick turn this ban was on

Return type `int`

side

Returns the team that made this ban

Return type *Side*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.currentgame.Game* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

bans

Returns the bans for this game

Return type `list<Ban>`

creation

Returns the creation timestamp for this game

Return type `datetime`

dto_type

alias of `CurrentGameInfo`

duration

Returns current duration of the game

Return type `timedelta`

id

Returns the game id

Return type `int`

map

Returns the map for this game

Return type *Map*

mode

Returns what game mode is being played in this game

Return type *GameMode*

observer_token

Returns the token associated with the observer for this game

Return type `str`

participants

Returns the game's participants

Return type `list<Participant>`

platform

Returns which platform (ie server) the game is being played on

Return type *Platform*

queue

Returns the queue type for this game

Return type *Queue*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

type

Returns the game type

Return type *GameType*

class *cassiopeia.type.core.currentgame.Participant* (*data*)
Bases: *cassiopeia.type.core.common.CassiopeiaObject*

bot

Returns whether the participant is a bot

Return type *bool*

champion

Returns the champion this participant is playing

Return type *Champion*

dto_type

alias of *CurrentGameParticipant*

masteries

Returns the participant's masteries

Return type *list<Mastery>*

profile_icon_id

Returns the participant's profile icon's id

Return type *int*

runes

Returns the participant's rune

Return type *list<Rune>*

side

Returns which side of the map the participant is on

Return type *Side*

summoner

Returns the summoner associated with this participant

Return type *Summoner*

summoner_name

Returns the participant's summoner name

Return type `str`

summoner_spell_d

Returns the participant's first summoner spell

Return type *SummonerSpell*

summoner_spell_f

Returns the participant's second summoner spell

Return type *SummonerSpell*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.featuredgames.Ban* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

champion

Returns the champion that was banned

Return type *Champion*

dto_type

alias of *BannedChampion*

pick_turn

Returns which pick turn this ban was on

Return type `int`

side

Returns which side banned this champion

Return type *Side*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.featuredgames.Game* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

bans

Returns the bans for this game

Return type `list<Ban>`

creation

Returns the creation timestamp for this game

Return type `datetime`

dto_type

alias of *FeaturedGameInfo*

duration

Returns current duration of the game

Return type `timedelta`

id

Returns the game id

Return type `int`

map

Returns the map for this game

Return type *Map*

mode

Returns what game mode is being played in this game

Return type *GameMode*

observer_token

Returns the token associated with the observer for this game

Return type `str`

participants

Returns the game's participants

Return type `list<Participant>`

platform

Returns which platform (ie server) the game is being played on

Return type *Platform*

queue

Returns the queue type for this game

Return type *Queue*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

type

Returns the game type

Return type *GameType*

class `cassiopeia.type.core.featuredgames.Participant` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

bot

Returns whether the participant is a bot

Return type `bool`

champion

Returns the champion this participant is playing

Return type *Champion*

dto_type

alias of *Participant*

profile_icon_id

Returns the participant's profile icon's id

Return type `int`

side

Returns which side of the map the participant is on

Return type `Side`

summoner_name

Returns the participant's summoner name

Return type `str`

summoner_spell_d

Returns the participant's first summonerspell

Return type `SummonerSpell`

summoner_spell_f

Returns the participant's second summonerspell

Return type `SummonerSpell`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.game.Game` (*data, summoner_id*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

champion

Returns the champion for this participant

Return type `Champion`

creation

Returns the time when this game was created

Return type `datetime`

dto_type

alias of `Game`

id

Returns the match ID

Return type `int`

invalid

Returns well, we don't know what this one is. let us know if you figure it out.

Return type `bool`

ip

Returns the amount of IP the participant gained for this game (the one that this game was pulled using)

Return type `int`

level

Returns the participant's champion level

Return type `int`

map

Returns the map this game was played on

Return type *Map*

mode

Returns the game mode

Return type *GameMode*

participants

Returns the participants in this game

Return type `list<Participant>`

side

Returns the side the participant was on

Return type *Side*

stats

Returns the participant's stats (the one that this game was pulled using)

Return type *Stats*

sub_type

Returns the game's sub-type

Return type *SubType*

summoner

Returns the summoner for this participant

Return type *Summoner*

summoner_spell_d

Returns the participant's first summoner spell (the one that this game was pulled using)

Return type *SummonerSpell*

summoner_spell_f

Returns the participant's second summoner spell (the one that this game was pulled using)

Return type *SummonerSpell*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

type

Returns the game type

Return type *GameType*

class `cassiopeia.type.core.game.Participant` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

champion

Returns the champion for this participant

Return type *Champion*

dto_type

alias of `Player`

side

Returns the side the participant was on

Return type *Side*

summoner

Returns the summoner for this participant

Return type *Summoner*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.game.Stats` (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

ally_monster_kills

Returns the number of neutral ally minions this participant killed

Return type `int`

assists

Returns the total number of assists this participant had

Return type `int`

combat_score

Returns dominion only. the part of the participant's score that came from combat-related activities

Return type `int`

consumables_bought

Returns the consumables that the participant bought (careful, they might have just sold them back or hit undo?)

Return type `list<Item>`

crowd_control_dealt

Returns the total amount of crowd control this participant dealt (in seconds)

Return type `int`

d_casts

Returns the number of times this participant cast his D summoner spell

Return type `int`

damage_dealt

Returns the total damage this participant dealt

Return type `int`

damage_dealt_player

Returns well, we don't know what this one is. let us know if you figure it out.

Return type `int`

damage_dealt_to_champions

Returns the total damage this participant dealt to champions

Return type `int`

damage_taken

Returns the total damage this participant received

Return type `int`

deaths

Returns the number of deaths this participant had

Return type `int`

double_kills

Returns the number of double kills this participant had

Return type `int`

dto_type

alias of `RawStats`

e_casts

Returns the number of times this participant cast his E

Return type `int`

elite_monsters_kills

Returns the number of elite monsters this participant killed

Return type `int`

enemy_monster_kills

Returns the number of neutral enemy minions this participant killed

Return type `int`

f_casts

Returns the number of times this participant cast his F summoner spell

Return type `int`

first_blood

Returns whether this participant got first blood

Return type `bool`

gold

Returns the participant's current gold

Return type `int`

gold_earned

Returns the participant's total gold

Return type *int*

gold_spent

Returns the participant's spent gold

Return type *int*

healing_done

Returns the amount of healing this participant did

Return type *int*

inhibitor_kills

Returns the total number of inhibitors this participant killed

Return type *int*

item0

Returns the participant's first item

Return type *Item*

item1

Returns the participant's second item

Return type *Item*

item2

Returns the participant's third item

Return type *Item*

item3

Returns the participant's fourth item

Return type *Item*

item4

Returns the participant's fifth item

Return type *Item*

item5

Returns the participant's sixth item

Return type *Item*

item6

Returns the participant's seventh item (i.e. their ward)

Return type *Item*

items

Returns the participant's items

Return type *list<Item>*

items_bought

Returns the number of items this participant bought

Return type `int`

kda

Returns the participant's kda

Return type `float`

killings_sprees

Returns the number of killing spree this participant had

Return type `int`

kills

Returns the total number of kills this participant had

Return type `int`

lane

Returns the lane this participant was in

Return type `Lane`

largest_critical_strike

Returns the largest critical strike this participant had

Return type `int`

largest_killing_sprees

Returns the largest killing spree this participant had

Return type `int`

largest_multi_kill

Returns the largest multikill this participant had

Return type `int`

level

Returns the participant's champion level

Return type `int`

magic_damage_dealt

Returns the total magic damage this participant dealt

Return type `int`

magic_damage_dealt_to_champions

Returns the total magic damage this participant dealt to champions

Return type `int`

magic_damage_taken

Returns the total magic damage this participant received

Return type `int`

minion_denies

Returns the number of minions this participant denied to the enemy. let us know if you figure out what this actually is

Return type `int`

minion_kills

Returns the number of minions this participant killed

Return type `int`

monster_kills

Returns the number of neutral minions this participant killed

Return type `int`

nexus_killed

Returns the number of nexuses this participant killed

Return type `int`

node_capture_assists

Returns dominion only. the number of nodes this participant assisted in capturing

Return type `int`

node_captured

Returns dominion only. the number of nodes this participant captured

Return type `int`

node_neutralization_assists

Returns dominion only. the number of nodes this participant assisted in neutralizing

Return type `int`

node_neutralizations

Returns dominion only. the number of nodes this participant neutralized

Return type `int`

objective_score

Returns dominion only. the part of the participant's score that came from objective-related activities

Return type `int`

objectives

Returns well, we don't know what this one is. let us know if you figure it out.

Return type `int`

penta_kills

Returns the number of penta kills this participant had

Return type `int`

physical_damage_dealt

Returns the total physical damage this participant dealt

Return type `int`

physical_damage_dealt_to_champions

Returns the total physical damage this participant dealt to champions

Return type `int`

physical_damage_taken

Returns the total physical damage this participant received

Return type `int`

q_casts

Returns the number of times this participant cast his Q

Return type `int`

quadra_kills

Returns the number of quadra kills this participant had

Return type `int`

r_casts

Returns the number of times this participant cast his R

Return type `int`

role

Returns the role of this participant

Return type *Role*

score

Returns the score for this participant

Return type `int`

score_rank

Returns if game was a dominion game, team rank of the player's total score (e.g., 1-5)

Return type `int`

side

Returns the side the participant was on

Return type *Side*

sight_wards_bought

Returns the number of sight wards this participant bought

Return type `int`

tier_3_items_bought

Returns the number of tier 3 items built

Return type `int`

time_played

Returns the amount of time this participant played

Return type `int`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

triple_kills

Returns the number of triple kills this participant had

Return type `int`

true_damage_dealt

Returns the total true damage this participant dealt

Return type `int`

true_damage_dealt_to_champions

Returns the total damage this participant dealt to champions

Return type `int`

true_damage_taken

Returns the total true damage this participant received

Return type `int`

turret_kills

Returns the number of turret kills this participant had

Return type `int`

units_healed

Returns the number of units this participant healed

Return type `int`

unreal_kills

Returns the number of unreal kills this participant had

Return type `int`

victory_points

Returns the number of victory points this participant gained from winning or losing this game

Return type `int`

vision_wards_bought

Returns the number of vision wards this participant bought

Return type `int`

w_casts

Returns the number of times this participant cast his W

Return type `int`

ward_kills

Returns the number of wards this participant killed

Return type `int`

wards_placed

Returns the number of wards this participant placed

Return type `int`

win

Returns whether the participant won the game or not

Return type `bool`

class `cassiopeia.type.core.league.Entry`(*data*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

division

Returns the league division of the participant

Return type `Division`

dto_type

alias of `LeagueEntry`

fresh_blood

Returns specifies if the participant is fresh blood (ie if they have just joined the league)

Return type `bool`

hot_streak

Returns specifies if the participant is on a hot streak

Return type `bool`

inactive

Returns specifies if the participant is inactive

Return type `bool`

league_points

Returns the league points of the participant

Return type `int`

losses

Returns number of current losses for the participant

Return type `int`

series

Returns series data for the participant. Only present if the participant is currently in a mini series

Return type `Series`

summoner

Returns the summoner represented by this entry. None if this entry is for a team

Return type `Summoner`

summoner_name

Returns the name of the summoner represented by this entry. An empty string if this entry is for a team

Return type `str`

team

Returns the team represented by this entry. None if this entry is for a summoner

Return type `Team`

team_name

Returns the name of the team represented by this entry. An empty string if this entry is for a summoner

Return type `str`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

veteran

Returns specifies if the participant is a veteran (ie they have been in this league for a long time)

Return type `bool`

wins

Returns the number of wins for the participant

Return type `int`

class `cassiopeia.type.core.league.League` (*data*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

dto_type
 alias of `League`

entries

Returns a list of the requested league entries, sorted by LP

Return type `list<Entry>`

name

Returns the name of the league

Return type `str`

participant_entry

Returns the entry for the relevant team or summoner that is a member of this league. Only present when full league is requested so that participant's entry can be identified. None when individual entry is requested

Return type `Entry`

queue

Returns the league's queue type

Return type `Queue`

summoner

Returns the relevant summoner that is a member of this league. Only present when full league is requested so that participant's entry can be identified. None when individual entry is requested or the participant is a team.

Return type *Summoner*

team

Returns the relevant team that is a member of this league. Only present when full league is requested so that participant's entry can be identified. None when individual entry is requested or the participant is a summoner.

Return type *Team*

tier

Returns the league's tier

Return type *Tier*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.league.Series* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

dto_type

alias of *MiniSeries*

losses

Returns number of current losses in the mini series

Return type *int*

progress

Returns string showing the current, sequential mini series progress where 'W' represents a win, 'L' represents a loss, and 'N' represents a game that hasn't been played yet

Return type *str*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

wins

Returns number of current wins in the mini series

Return type *int*

wins_required

Returns number of wins required for promotion

Return type *int*

class *cassiopeia.type.core.match.Ban* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

champion

Returns the champion that was banned

Return type *Champion*

dto_type
alias of `BannedChampion`

pick_turn
Returns which pick turn this ban was on
Return type `int`

to_json (***kwargs*)
Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.match.CombinedParticipant` (*participant, identity*)
Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

to_json (***kwargs*)
Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.core.match.Event` (*data, participants*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

ascended
Returns what died in the event
Return type `Ascended`

assists
Returns the participants who assisted in the event
Return type `list<Participant>`

building
Returns the building type associated with the event, if any
Return type `Building`

creator
Returns the participant who created the event
Return type `Participant`

dto_type
alias of `Event`

item
Returns the item involved in the event
Return type `Item`

item_after
Returns the item involved before the event happened
Return type `Item`

item_before
Returns the item involved after the event happened
Return type `Item`

killer

Returns the participant who did the killing

Return type *Participant*

lane

Returns the lane this event happened in

Return type *Lane*

level_up

Returns the level up type of the event

Return type *LevelUp*

monster

Returns the monster that was involved in the event

Return type *Monster*

participant

Returns the primary participant that event happened to or who was involved in the event

Return type *Participant*

point_captured

Returns dominion only, which point was captured

Return type *Point*

position

Returns the position where the event occurred

Return type *Position*

side

Returns the side this participant was on

Return type *Side*

skill_slot

Returns the skill slot of the event

Return type *int*

timestamp

Returns the timestamp for this event

Return type *datetime*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

tower

Returns which tower was involved in the event

Return type *Tower*

type

Returns the event type

Return type *EventType*

victim

Returns the victim!

Return type *Participant*

ward

Returns the ward type associated with this event

Return type *Ward*

class `cassiopeia.type.core.match.Frame` (*data*, *participants*)
 Bases: `cassiopeia.type.core.common.CassiopeiaObject`

dto_type
 alias of *Frame*

events

Returns the events in this frame

Return type `list<Event>`

participant_frames

Returns the frames in for each participant

Return type `dict<participantID, ParticipantFrame>`

timestamp

Returns the timestamp for this frame

Return type `datetime`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.match.Match` (*data*)
 Bases: `cassiopeia.type.core.common.CassiopeiaObject`

blue_team

Returns the blue team

Return type *Team*

creation

Returns when the match was created

Return type `datetime`

dto_type
 alias of `MatchDetail`

duration

Returns duration of the match

Return type `datetime`

frames

Returns the frames in this match

Return type `list<Frame>`

id

Returns the match ID

Return type `int`

map

Returns the map the match was played on

Return type *Map*

mode

Returns the game mode

Return type *GameMode*

participants

Returns the participants in this match

Return type `list<Participant>`

platform

Returns the platform (ie server) for this match

Return type *Platform*

queue

Returns the queue type for this match

Return type *Queue*

red_team

Returns the red team

Return type *Team*

region

Returns the region the match was played in

Return type *Region*

season

Returns the season this match was played in

Return type *Season*

timeline

Returns the match timeline

Return type *Timeline*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

type

Returns the game type

Return type *GameType*

version

Returns the patch this match was played in

Return type `str`

class `cassiopeia.type.core.match.Participant` (*data*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

champion

Returns the champion this participant played

Return type `Champion`

dto_type

alias of `CombinedParticipant`

id

Returns the participant ID

Return type `int`

masteries

Returns the participant's masteries

Return type `list<Mastery>`

match_history_uri

Returns the the URI to access this player's match history online

Return type `str`

previous_season_tier

Returns the participant's tier last season

Return type `Tier`

runes

Returns the participant's current runes

Return type `list<Rune>`

side

Returns the side this participant was on

Return type `Side`

stats

Returns the participant's stats

Return type `ParticipantStats`

summoner

Returns the summoner associated with this participant

Return type `Summoner`

summoner_id

Returns the participant's summoner id

Return type `str`

summoner_name

Returns the participant's summoner name

Return type `str`

summoner_spell_d

Returns the participant's first summoner spell

Return type *SummonerSpell*

summoner_spell_f

Returns the participant's second summoner spell

Return type *SummonerSpell*

timeline

Returns the participant's timeline

Return type *ParticipantTimeline*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.match.ParticipantFrame` (*data, participants*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

current_gold

Returns the participant's current gold

Return type `int`

dto_type

alias of *ParticipantFrame*

gold

Returns the participant's total gold

Return type `int`

jungle_monsters_killed

Returns the number of neutral jungle monsters killed

Return type `int`

level

Returns the participant's champion level

Return type `int`

minion_kills

Returns the number of minions killed

Return type `int`

participant

Returns the participant whose frames you are looking at

Return type *Participant*

position

Returns the position of the participant

Return type *Position*

score

Returns dominion only. the score for this participant

Return type *int*

team_score

Returns the team score for the participant

Return type *int*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

xp

Returns the amount of XP the participant has

Return type *int*

class *cassiopeia.type.core.match.ParticipantStats* (*data*)
Bases: *cassiopeia.type.core.common.CassiopeiaObject*

ally_monster_kills

Returns the number of neutral jungle minions killed in your team's jungle

Return type *int*

assists

Returns the total number of assists this participant had

Return type *int*

champion_level

Returns the champion level of the participant when the game ended

Return type *int*

combat_score

Returns dominion only. the part of the participant's score that came from combat-related activities

Return type *int*

crowd_control_dealt

Returns the total amount of crowd control this participant dealt (in seconds)

Return type *int*

cs

damage_dealt

Returns the total damage this participant dealt

Return type *int*

damage_dealt_to_champions

Returns the total damage this participant dealt to champions

Return type `int`

damage_taken

Returns the total damage this participant received

Return type `int`

deaths

Returns the number of deaths this participant had

Return type `int`

double_kills

Returns the number of double kills this participant had

Return type `int`

dto_type

alias of *ParticipantStats*

enemy_monster_kills

Returns the number of neutral jungle minions killed in the enemy team's jungle

Return type `int`

first_blood

Returns whether participant team got first blood

Return type `bool`

first_blood_assist

Returns flag indicating if participant got an assist on first blood

Return type `bool`

first_inhibitor

Returns flag indicating if this participant destroyed the first inhibitor

Return type `bool`

first_inhibitor_assist

Returns flag indicating if participant got an assist on the first inhibitor

Return type `bool`

first_turret

Returns flag indicating if this team destroyed the first tower

Return type `bool`

first_turret_assist

Returns flag indicating if participant got an assist on the first tower

Return type `bool`

gold_earned

Returns the participant's total gold

Return type `int`

gold_spent

Returns the participant's spent gold

Return type `int`

healing_done

Returns the amount of healing this participant did

Return type `int`

inhibitor_kills

Returns the number of inhibitors this team killed

Return type `int`

item0

Returns the participant's first item

Return type *Item*

item1

Returns the participant's second item

Return type *Item*

item2

Returns the participant's third item

Return type *Item*

item3

Returns the participant's fourth item

Return type *Item*

item4

Returns the participant's fifth item

Return type *Item*

item5**item6**

Returns the participant's seventh item (i.e. their ward)

Return type *Item*

items

Returns the participant's items

Return type `list<Item>`

kda

Returns the participant's kda

Return type `float`

killing_sprees

Returns the number of killing spree this participant had

Return type `int`

kills

Returns the total number of kills this participant had

Return type `int`

largest_critical_strike

Returns the largest critical strike this participant had

Return type `int`

largest_killing_spree

Returns the largest killing spree this participant had

Return type `int`

largest_multi_kill

Returns the largest multikill this participant had

Return type `int`

magic_damage_dealt

Returns the total magic damage this participant dealt

Return type `int`

magic_damage_dealt_to_champions

Returns the total magic damage this participant dealt to champions

Return type `int`

magic_damage_taken

Returns the total magic damage this participant received

Return type `int`

minion_kills

Returns the number of minions killed

Return type `int`

monster_kills

Returns the number of neutral minions this participant killed

Return type `int`

node_capture_assists

Returns dominion only. the number of nodes this participant assisted in capturing

Return type `int`

node_neutralization_assists

Returns dominion only. the number of nodes this participant assisted in neutralizing

Return type `int`

node_neutralizations

Returns dominion only. the number of nodes this participant neutralized

Return type `int`

nodes_captured

Returns dominion only. the number of nodes this participant captured

Return type `int`

objective_score

Returns dominion only. the part of the participant's score that came from objective-related activities

Return type `int`

penta_kills

Returns the number of penta kills this participant had

Return type `int`

physical_damage_dealt

Returns the total physical damage this participant dealt

Return type `int`

physical_damage_dealt_to_champions

Returns the total physical damage this participant dealt to champions

Return type `int`

physical_damage_taken

Returns the total physical damage this participant received

Return type `int`

quadra_kills

Returns the number of quadra kills this participant had

Return type `int`

score

Returns dominion only. the score for this participant

Return type `int`

score_rank

Returns if game was a dominion game, team rank of the player's total score (e.g., 1-5)

Return type `int`

sight_wards_bought

Returns the number of sight wards this participant bought

Return type `int`

team_objectives

Returns if game was a dominion game, number of completed team objectives (i.e., quests)

Return type `int`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

triple_kills

Returns the number of triple kills this participant had

Return type *int*

true_damage_dealt

Returns the total true damage this participant dealt

Return type *int*

true_damage_dealt_to_champions

Returns the total damage this participant dealt to champions

Return type *int*

true_damage_taken

Returns the total true damage this participant received

Return type *int*

turret_kills

Returns the number of turret kills this participant had

Return type *int*

units_healed

Returns the number of units this participant healed

Return type *int*

unreal_kills

Returns the number of unreal kills this participant had

Return type *int*

vision_wards_bought

Returns the number of vision wards this participant bought

Return type *int*

ward_kills

Returns the number of wards this participant killed

Return type *int*

wards_placed

Returns the number of wards this participant placed

Return type *int*

win

Returns whether or not the participant won the game

Return type *bool*

class *cassiopeia.type.core.match.ParticipantTimeline* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

ancient_golem_assists_per_min_counts

Returns ancient golem assists per minute timeline counts

Return type *ParticipantTimelineData*

ancient_golem_kills_per_min_counts

Returns ancient golem kills per minute timeline counts

Return type *ParticipantTimelineData*

assisted_lane_deaths_per_min_deltas

Returns assisted lane deaths per minute timeline data

Return type *ParticipantTimelineData*

assisted_lane_kills_per_min_deltas

Returns assisted lane kills per minute timeline data

Return type *ParticipantTimelineData*

baron_assists_per_min_counts

Returns baron assists per minute timeline counts

Return type *ParticipantTimelineData*

baron_kills_per_min_counts

Returns baron kills per minute timeline counts

Return type *ParticipantTimelineData*

creeps_per_min_deltas

Returns creeps per minute timeline data

Return type *ParticipantTimelineData*

cs_diff_per_min_deltas

Returns creep score difference per minute timeline data

Return type *ParticipantTimelineData*

damage_taken_diff_per_min_deltas

Returns damage taken difference per minute timeline data

Return type *ParticipantTimelineData*

damage_taken_per_min_deltas

Returns damage taken per minute timeline data

Return type *ParticipantTimelineData*

dragon_assists_per_min_counts

Returns dragon assists per minute timeline counts

Return type *ParticipantTimelineData*

dragon_kills_per_min_counts

Returns dragon kills per minute timeline counts

Return type *ParticipantTimelineData*

dto_type

alias of *ParticipantTimeline*

elder_lizard_assists_per_min_counts

Returns elder lizard assists per minute timeline counts

Return type *ParticipantTimelineData*

elder_lizard_kills_per_min_counts

Returns elder lizard kills per minute timeline counts

Return type *ParticipantTimelineData*

gold_per_min_deltas

Returns gold per minute timeline data

Return type *ParticipantTimelineData*

inhibitor_assists_per_min_counts

Returns inhibitor assists per minute timeline counts

Return type *ParticipantTimelineData*

inhibitor_kills_per_min_counts

Returns inhibitor kills per minute timeline counts

Return type *ParticipantTimelineData*

lane

Returns the lane this participant was in

Return type *Lane*

role

Returns the role of this participant

Return type *Role*

spider_assists_per_min_counts

Returns vilemaw assists per minute timeline counts

Return type *ParticipantTimelineData*

spider_kills_per_min_counts

Returns vilemaw kills per minute timeline counts

Return type *ParticipantTimelineData*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

turret_kills_per_min_deltas

Returns tower kills per minute timeline data

Return type *ParticipantTimelineData*

turret_assists_per_min_counts

Returns tower assists per minute timeline counts

Return type *ParticipantTimelineData*

turret_kills_per_min_counts

Returns tower kills per minute timeline counts

Return type *ParticipantTimelineData*

wards_per_min_deltas

Returns wards placed per minute timeline data

Return type *ParticipantTimelineData*

xp_diff_per_min_deltas

Returns experience difference per minute timeline data

Return type *ParticipantTimelineData*

xp_per_min_deltas

Returns experience per minute timeline data

Return type *ParticipantTimelineData*

class `cassiopeia.type.core.match.ParticipantTimelineData` (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

dto_type

alias of *ParticipantTimelineData*

ten_to_twenty

Returns value per minute from 10 min to 20 min

Return type *float*

thirty_to_end

Returns value per minute from 30 min to the end of the game

Return type *float*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

twenty_to_thirty

Returns value per minute from 20 min to 30 min

Return type *float*

zero_to_ten

Returns value per minute from the beginning of the game to 10 min

Return type *float*

class `cassiopeia.type.core.match.Position` (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

dto_type

alias of *Position*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

x**Returns** the x-position of the pixel**Return type** `int`**y****Returns** the y-position of the pixel**Return type** `int`

class `cassiopeia.type.core.match.Team`(*data, participants*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

bans**Returns** the bans for this game**Return type** `list<Ban>`**baron_kills****Returns** the number of times the team killed Baron**Return type** `int`**dragon_kills****Returns** the number of times the team killed Dragon**Return type** `int`**dto_type**alias of `Team`**first_baron****Returns** whether or not the team killed the first baron**Return type** `bool`**first_blood****Returns** whether this team got first blood**Return type** `bool`**first_dragon****Returns** whether or not this team killed the first dragon**Return type** `bool`**first_inhibitor****Returns** flag indicating if this team destroyed the first inhibitor**Return type** `bool`**first_rift_herald****Returns** flag indicating if this team killed the first rift herald**Return type** `bool`**first_turret****Returns** flag indicating if this team destroyed the first tower**Return type** `bool`

inhibitor_kills

Returns the number of inhibitors this team killed

Return type `int`

participants

Returns the participants on this team

Return type `list<Participant>`

rift_herald_kills

Returns the number of rift heralds this team killed

Return type `int`

side

Returns the side this team was on

Return type `Side`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

turret_kills

Returns the number of turret kills this team had

Return type `int`

victory_score

Returns dominion only, the points the team had at game end

Return type `int`

vilemaw_kills

Returns the number of times the team has killed Vilemaw

Return type `int`

win

Returns whether or not the team won the game

Return type `bool`

class `cassiopeia.type.core.match.Timeline` (*data, participants*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

dto_type

alias of `Timeline`

frame_interval

Returns the number of milliseconds between frames

Return type `timedelta`

frames

Returns the frames in this match

Return type `list<Frame>`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.matchlist.MatchReference` (*data*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

champion

Returns the champion that the summoner played for the summoner that was used to pull this match reference

Return type *Champion*

dto_type
alias of *MatchReference*

id

Returns the match id for this match

Return type *int*

lane

Returns the lane that the summoner was in for the summoner that was used to pull this match reference

Return type *Lane*

match (*include_timeline=True*)
Gets the full information for this match

Parameters **include_timeline** (*bool*) – whether to include timeline data in the returned match

Returns the match

Return type *Match*

platform

Returns the platform (ie server) for this match

Return type *Platform*

queue

Returns the queue type for this match

Return type *Queue*

role

Returns the role that the summoner was in for the summoner that was used to pull this match reference

Return type *Role*

season

Returns the season that this match was played in

Return type *Season*

timestamp

Returns the timestamp for this match

Return type *datetime*

```

to_json (**kwargs)
    Parameters data (CassiopeiaDto) – the underlying DTO object with the data for this type

class cassiopeia.type.core.staticdata.Champion (data)
    Bases: cassiopeia.type.core.common.CassiopeiaObject

    ally_tips
        Returns the set of tips for allies of this champion
        Return type list<str>

    blurb
        Returns the items in this item set
        Return type list<SetItem>

    dto_type
        alias of Champion

    enemy_tips
        Returns the set of tips for enemies of this champion
        Return type list<str>

    id
        Returns the ID of this champion
        Return type int

    image
        Returns the image of this champion
        Return type Image

    info
        Returns ratings of what this champion is good/bad at
        Return type ChampionInfo

    key
        Returns this champion's identifying key
        Return type str

    lore
        Returns this champion's lore
        Return type str

    mastery_level (summoner)
        Gets the ChampionMastery object for the specified summoner :param summoner: the summoner to get
        champion mastery for :type summoner: Summoner
        Returns well, we don't know what this one is. let us know if you figure it out.
        Return type bool

    name
        Returns the name of this champion

```

Return type `str`

passive

Returns this champion's passive

Return type *Passive*

recommended_items

Returns item recommendations for this champion

Return type `list<RecommendedItems>`

resource

Returns the items in this item set

Return type `list<SetItem>`

skins

Returns this champion's skins

Return type `list<Skin>`

spells

Returns this champion's spells

Return type `list<Spell>`

stats

Returns the stats for this champion

Return type *ChampionStats*

status ()

Returns the items in this item set

Return type `list<SetItem>`

tags

Returns this champions's tags for sorting champions

Return type `list<str>`

title

Returns this champion's title

Return type `str`

to_json (kwargs)**

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.staticdata.ChampionInfo (data)`
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

defense

Returns defensive rating (out of 10)

Return type `int`

difficulty

Returns difficulty rating (out of 10)

Return type `int`

dto_type

alias of `Info`

magic

Returns magic damage output rating (out of 10)

Return type `int`

physical

Returns physical damage output rating (out of 10)

Return type `int`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.staticdata.ChampionStats` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

armor

Returns armor

Return type `float`

armor_per_level

Returns armor per level

Return type `float`

attack_damage

Returns attack damage

Return type `float`

attack_damage_per_level

Returns attack damage per level

Return type `float`

attack_range

Returns attack range

Return type `float`

attack_speed

Returns attack speed

Return type `float`

critical_strike_chance

Returns critical strike chance

Return type `float`

critical_strike_chance_per_level

Returns critical strike chance per level

Return type float

dto_type

alias of Stats

health

Returns health

Return type float

health_per_level

Returns health per level

Return type float

health_regen

Returns health regen

Return type float

health_regen_per_level

Returns health regen per level

Return type float

magic_resist

Returns magic resist

Return type float

magic_resist_per_level

Returns magic resist per level

Return type float

mana

Returns mana

Return type float

mana_per_level

Returns mana per level

Return type float

mana_regen

Returns mana regen

Return type float

mana_regen_per_level

Returns mana regen per level

Return type float

movespeed

Returns movespeed

Return type float

percent_attack_speed_per_level

Returns attack speed per level

Return type `float`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.staticdata.Gold` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

base

Returns the base price of the item

Return type `int`

dto_type

alias of `Gold`

purchasable

Returns whether the item can be bought

Return type `bool`

sell

Returns the sell price of the item

Return type `int`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

total

Returns the total price of the item

Return type `int`

class `cassiopeia.type.core.staticdata.Image` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

dto_type

alias of `Image`

group

Returns how many of this item are in the block

Return type `int`

height

Returns the height of the image

Return type `int`

link

Returns the link to the image. See <https://developer.riotgames.com/docs/static-data> for more information.

Return type `str`

sprite

Returns the sprite image link. See <https://developer.riotgames.com/docs/static-data> for more information.

Return type `str`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

width

Returns the width of the image

Return type `int`

x

Returns the x offset of the image

Return type `int`

y

Returns the y offset of the image

Return type `int`

class `cassiopeia.type.core.staticdata.Item` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

blurb

Returns the blurb for this item

Return type `str`

categories

Returns the shop categories that this item belongs to

Return type `list<str>`

component_of

Returns the items this one is a component of

Return type `list<Item>`

components

Returns the components for this item

Return type `list<Item>`

consumable

Returns whether the item is a consumable

Return type `bool`

consume_on_full

Returns well, we don't know what this one is. let us know if you figure it out.

Return type `bool`

description

Returns the description of the item

Return type `str`

dto_typealias of *Item***effect****Returns** the item's effects**Return type** dict<str, bool>**gold****Returns** price information for this item**Return type** *Gold***group****Returns** the group for this item**Return type** str**hide****Returns** well, we don't know what this one is. let us know if you figure it out.**Return type** bool**id****Returns** the ID of this item**Return type** int**image****Returns** the image of this item**Return type** *Image***in_store****Returns** well, we don't know what this one is. let us know if you figure it out.**Return type** bool**keywords****Returns** a string formatted list of search keywords for this item in the shop**Return type** str**maps****Returns** a listing of whether this item is available on each map**Return type** dict<Map, bool>**meta_data****Returns** meta data about this item**Return type** *MetaData***name****Returns** the name of this item**Return type** str**required_champion**

Returns the required champion for this item

Return type *Champion*

sanitized_description

Returns the sanitized description of this item

Return type *str*

special_recipe

Returns well, we don't know what this one is. let us know if you figure it out.

Return type *int*

stacks

Returns the number of stacks this item can have

Return type *int*

stats

Returns the stats for this item

Return type *ItemStats*

tags

Returns this item's tags for sorting items

Return type *list<str>*

tier

Returns what tier the item is

Return type *str*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.staticdata.ItemSet* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

dto_type

alias of *Block*

items

Returns the items in this item set

Return type *list<SetItem>*

rec_math

Returns well, we don't know what this one is. let us know if you figure it out.

Return type *bool*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

type

Returns what type the item set is (e.g. starting items)

Return type *str*

class cassiopeia.type.core.staticdata.**ItemStats** (*data*, *scraped_stats*={})

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

ability_power

Returns ability power

Return type *float*

ability_power_per_level

Returns ability power per level

Return type *float*

armor

Returns armor

Return type *float*

armor_penetration

Returns armor penetration

Return type *float*

armor_penetration_per_level

Returns armor penetration per level

Return type *float*

armor_per_level

Returns armor per level

Return type *float*

attack_damage

Returns attack damage

Return type *float*

attack_damage_per_level

Returns float attack damage per level

attack_speed

Returns attack speed

Return type *float*

block

Returns blocked damage per attack

Return type *float*

cooldown_reduction

Returns cooldown reduction

Return type *float*

cooldown_reduction_per_level

Returns cooldown reduction per level

Return type `float`

critical_strike_chance

Returns critical strike chance

Return type `float`

critical_strike_chance_per_level

Returns critical strike chance per level

Return type `float`

critical_strike_damage

Returns critical strike damage modification

Return type `float`

critical_strike_damage_per_level

Returns critical strike damage per level

Return type `float`

dodge_chance

Returns dodge chance

Return type `float`

dodge_chance_per_level

Returns dodge change per level

Return type `float`

dto_type

alias of `BasicDataStats`

energy

Returns energy

Return type `float`

energy_per_level

Returns energy per level

Return type `float`

energy_regen

Returns energy regen

Return type `float`

energy_regen_per_level

Returns energy regen per level

Return type `float`

gold_per_ten

Returns gold per 10 seconds

Return type `float`

health

Returns health

Return type float

health_per_level

Returns health per level

Return type float

health_regen

Returns health regen

Return type float

health_regen_per_level

Returns health regen per level

Return type float

life_steal

Returns life steal

Return type float

magic_penetration

Returns magic penetration

Return type float

magic_penetration_per_level

Returns magic penetration per level

Return type float

magic_resist

Returns magic resist

Return type float

magic_resist_per_level

Returns magic resist per level

Return type float

mana

Returns mana

Return type float

mana_per_level

Returns armor per level

Return type float

mana_regen

Returns float mana regen

mana_regen_per_level

Returns mana regen per level

Return type float

movespeed

Returns movespeed

Return type float

movespeed_per_level

Returns movespeed per level

Return type float

percent_ability_power

Returns percent ability power

Return type float

percent_armor

Returns percent armor

Return type float

percent_armor_penetration

Returns percent armor penetration

Return type float

percent_armor_penetration_per_level

Returns percent armor penetration per level

Return type float

percent_attack_damage

Returns percent attack damage

Return type float

percent_attack_speed

Returns percent attack speed

Return type float

percent_attack_speed_per_level

Returns percent attack speed per level

Return type float

percent_base_attack_damage

Returns percent attack damage

Return type float

percent_base_health_regen

Returns percent base health regen

Return type float

percent_base_mana_regen

Returns percent base mana regen

Return type float

percent_block

Returns percent blocked damage per attack

Return type float

percent_bonus_armor_penetration

Returns percent armor penetration

Return type float

percent_bonus_health

Returns percent bonus health

Return type float

percent_critical_strike_damage

Returns percent critical strike damage modification

Return type float

percent_health

Returns percent health

Return type float

percent_health_regen

Returns percent health regen

Return type float

percent_magic_pen_per_level

Returns percent magic penetration per level

Return type float

percent_magic_penetration

Returns percent magic penetration

Return type float

percent_magic_resist

Returns percent magic resist

Return type float

percent_mana

Returns percent mana

Return type float

percent_mana_regen

Returns percent mana regen

Return type float

percent_movespeed

Returns percent movespeed

Return type `float`

percent_movespeed_per_level

Returns percent movespeed per level

Return type `float`

percent_time_dead

Returns percent time dead modification

Return type `float`

percent_time_dead_per_level

Returns percent time dead modification per level

Return type `float`

percent_xp_bonus

Returns percent experience bonus

Return type `float`

spell_vamp

Returns spell vamp

Return type `float`

stat = 'spell_vamp'

stats = ['spell_vamp']

tag = 'Spell Vamp'

tags = {'Other': [], 'Mana Regen': ['base_mana_regen', 'bonus_mana_regen', 'mana_regen_per_level', 'percent_mana_regen']}

tenacity

Returns tenacity

Return type `float`

time_dead

Returns time dead modification

Return type `float`

time_dead_per_level

Returns time dead modification per level

Return type `float`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

xp_bonus

Returns experience bonus

Return type `float`


```

class cassiopeia.type.core.staticdata.LevelTip(data)
    Bases: cassiopeia.type.core.common.CassiopeiaObject

    dto_type
        alias of LevelTip

    effects

        Returns the level-by-level replacements for {{ e# }} tags in other values

        Return type list<list<float>>

    labels

        Returns the labels for the changes in effects

        Return type list<str>

    to_json (**kwargs)

        Parameters data (CassiopeiaDto) – the underlying DTO object with the data for this type

class cassiopeia.type.core.staticdata.MapDetails(data)
    Bases: cassiopeia.type.core.common.CassiopeiaObject

    dto_type
        alias of MapDetails

    image

        Returns the image of this map

        Return type Image

    map

        Returns the type of this map

        Return type Map

    map_id

        Returns the ID of this map

        Return type int

    to_json (**kwargs)

        Parameters data (CassiopeiaDto) – the underlying DTO object with the data for this type

    unpurchasable_items

        Returns the items that can't be bought on this map

        Return type list<Item>

class cassiopeia.type.core.staticdata.Mastery(data)
    Bases: cassiopeia.type.core.common.CassiopeiaObject

    descriptions

        Returns descriptions of this mastery by rank

        Return type list<str>

    dto_type
        alias of Mastery

    id

```

Returns the ID of this mastery

Return type `int`

image

Returns the image of this mastery

Return type *Image*

max_rank

Returns the max rank for this mastery

Return type `int`

name

Returns the name of this mastery

Return type `str`

prerequisite

Returns the prerequisite for this mastery

Return type *Mastery*

sanitized_descriptions

Returns sanitized descriptions of this mastery by rank

Return type `list<str>`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

tree

Returns which mastery tree this mastery belongs to

Return type `str`

class *cassiopeia.type.core.staticdata.MetaData* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

dto_type

alias of *MetaData*

rune

Returns whether the item is a rune

Return type `bool`

tier

Returns what tier the item is

Return type `str`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

type

Returns the type of meta data

Return type `str`

```

class cassiopeia.type.core.staticdata.Passive(data)
    Bases: cassiopeia.type.core.common.CassiopeiaObject

    description
        Returns the description of the passive
        Return type str

    dto_type
        alias of Passive

    image
        Returns the image of this passive
        Return type Image

    name
        Returns the name of this passive
        Return type str

    sanitized_description
        Returns the sanitized description of this passive
        Return type str

    to_json(**kwargs)
        Parameters data (CassiopeiaDto) – the underlying DTO object with the data for this type

class cassiopeia.type.core.staticdata.Realm(data)
    Bases: cassiopeia.type.core.common.CassiopeiaObject

    cdn
        Returns the base CDN url
        Return type str

    css
        Returns the latest version of the Dragon Magic's css file
        Return type str

    data_type_versions
        Returns the latest versions for listed data types
        Return type dict<str, str>

    dragon_magic
        Returns the latest version of Dragon Magic
        Return type str

    dto_type
        alias of Realm

    language
        Returns the default locale for this realm
        Return type str

```

legacy

Returns the legacy script mode for IE6 or older

Return type `str`

profile_icon_id_max

Returns the largest profileicon id that can be used under 500.0 Any profileicon that is requested between this number and 500 should be mapped to 0.0.

Return type `int`

store

Returns additional api data drawn from other sources that may be related to data dragon functionality

Return type `str`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

version

Returns the current version of this file

Return type `str`

class `cassiopeia.type.core.staticdata.RecommendedItems` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

champion

Returns the champion these recommendations are for

Return type `Champion`

dto_type

alias of `Recommended`

item_sets

Returns the sets of items that make up this reommended page

Return type `list<ItemSet>`

map

Returns the name of the map these recommendations are for

Return type `str`

mode

Returns the game mode these recommendations are for

Return type `GameMode`

name

Returns the name of this item

Return type `str`

priority

Returns whether this is a priority recommendation

Return type `bool`

`to_json` (***kwargs*)

Parameters `data` (`CassiopeiaDto`) – the underlying DTO object with the data for this type

type

Returns the type of recommended items

Return type `str`

`class cassiopeia.type.core.staticdata.Rune(data)`
 Bases: `cassiopeia.type.core.common.CassiopeiaObject`

description

Returns the description of the rune

Return type `str`

dto_type

alias of `Rune`

id

Returns the ID of this rune

Return type `int`

image

Returns the image of this rune

Return type `Image`

meta_data

Returns meta data about this rune

Return type `MetaData`

name

Returns the name of this rune

Return type `str`

rune_type

Returns what type of rune this is

Return type `str`

sanitized_description

Returns the sanitized description of this rune

Return type `str`

stats

Returns the stats for this rune

Return type `ItemStats`

tags

Returns this rune's tags for sorting runes

Return type `list<str>`

`to_json (**kwargs)`

Parameters `data` (`CassiopeiaDto`) – the underlying DTO object with the data for this type

`class cassiopeia.type.core.staticdata.SetItem(data)`

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

count

Returns how many of this item are in the block

Return type `int`

dto_type

alias of `BlockItem`

item

Returns the item itself

Return type `Item`

`to_json (**kwargs)`

Parameters `data` (`CassiopeiaDto`) – the underlying DTO object with the data for this type

`class cassiopeia.type.core.staticdata.Skin(data, key)`

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

dto_type

alias of `Skin`

id

Returns the ID of this skin

Return type `int`

loading

Returns the link to the loading art for this skin

Return type `str`

name

Returns the name of this skin

Return type `str`

number

Returns where in the skin order this skin comes

Return type `int`

splash

Returns the link to the splash art for this skin

Return type `str`

`to_json (**kwargs)`

Parameters `data` (`CassiopeiaDto`) – the underlying DTO object with the data for this type

`class cassiopeia.type.core.staticdata.Spell(data)`

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

alternate_images

Returns the alternate images for this spell

Return type list<Image>

cooldown_burn

Returns a string formatted list of the spell's cooldowns by level

Return type str

cooldowns

Returns the cooldowns of this spell level-by-level

Return type list<float>

cost_burn

Returns a string formatted list of the spell's cost by level

Return type str

cost_type

Returns what the spell costs to use (e.g. mana or energy)

Return type str

costs

Returns the cost of the spell level-by-level

Return type list<int>

description

Returns the description of the spell

Return type str

dto_type

alias of ChampionSpell

effect_burn

Returns the string formatted replacements for {{ e# }} tags in other values by level

Return type list<str>

effects

Returns the level-by-level replacements for {{ e# }} tags in other values

Return type list<list<float>>

image

Returns the image of this spell

Return type Image

key

Returns this spell's identifying key

Return type str

level_tip

Returns the level-up tips for this spell

Return type *LevelTip*

max_rank

Returns the maximum level for this spell

Return type *int*

name

Returns the name of this spell

Return type *str*

range

Returns the level-by-level range of this spell

Return type *self | list<int>*

range_burn

Returns the string formatted range of this spell by level

Return type *str*

resource

Returns what resource this spell uses

Return type *str*

sanitized_description

Returns the sanitized description of this spell

Return type *str*

sanitized_tooltip

Returns the sanitized tooltip for this spell

Return type *str*

sanitized_tooltip_for_level (*level*, *rank*)

Gets the sanitized tooltip for this spell for a specific level/rank

Parameters

- **level** (*int*) – the level of the champion
- **rank** (*int*) – the rank of this spell

Returns the sanitized tooltip for this spell for the specified level and rank

Return type *str*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

tooltip

Returns the tooltip for this spell

Return type *str*

tooltip_for_level (*level*, *rank*)

Gets the tooltip for this spell for a specific level/rank

Parameters

- **level** (*int*) – the level of the champion
- **rank** (*int*) – the rank of this spell

Returns how many of this item are in the block

Return type *int*

variables

Returns the variables that determine this spell's damage

Return type *SpellVariables*

class `cassiopeia.type.core.staticdata.SpellVariables` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

coefficients

Returns the coefficients for determining spell scaling

Return type *list<float>*

dto_type

alias of *SpellVars*

dynamic

Returns whether the spell variables are dynamic

Return type *str*

key

Returns this spell's identifying key

Return type *str*

link

Returns the link to the image. See <https://developer.riotgames.com/docs/static-data> for more information.

Return type *str*

ranks_with

Returns what these variables rank with

Return type *str*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.staticdata.SummonerSpell` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

cooldown_burn

Returns formatted list of the spell's cooldowns by level

Return type *str*

cooldowns

Returns the cooldowns of this spell level-by-level

Return type *list<float>*

cost_burn

Returns formatted list of the spell's cost by level

Return type `str`

cost_type

Returns what the spell costs to use (e.g. mana or energy)

Return type `str`

costs

Returns the cost of the spell level-by-level

Return type `list<int>`

description

Returns the description of the summoner spell

Return type `str`

dto_type

alias of *SummonerSpell*

effect_burn

Returns the string formatted replacements for `{{ e# }}` tags in other values by level

Return type `list<str>`

effects

Returns the level-by-level replacements for `{{ e# }}` tags in other values

Return type `list<list<float>>`

id

Returns the ID of this summoner spell

Return type `int`

image

Returns the image of this summoner spell

Return type *Image*

key

Returns this summoner spell's identifying key

Return type `str`

leveltip

Returns the level-up tips for this spell

Return type *LevelTip*

max_rank

Returns the maximum level for this spell

modes

Returns the game modes that this spell is allowed on

Return type `list<GameMode>`

name

Returns the name of this spell

Return type `str`

range

Returns the level-by-level range of this spell

Return type `list<int>`

range_burn

Returns formatted range of this spell by level

Return type `str`

resource

Returns what resource this spell uses

Return type `str`

sanitized_description

Returns the sanitized description of this summoner spell

Return type `str`

sanitized_tooltip

Returns the sanitized tooltip for this summoner spell

Return type `str`

sanitized_tooltip_for_level (*level*)

Gets the sanitized tooltip for this spell for a specific level

Parameters **level** (*int*) – the level of the champion

Returns the sanitized tooltip for this summoner spell for the provided level

Return type `str`

summoner_level

Returns the summoner level required to use this spell

Return type `int`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

tooltip

Returns the tooltip for this spell

Return type `str`

tooltip_for_level (*level*)

Gets the tooltip for this spell for a specific level

Parameters **level** (*int*) – the level of the champion

Returns the tooltip for that rank/level

Return type `str`

variables

Returns the variables that determine this spell's effects

Return type *SpellVariables*

class `cassiopeia.type.core.stats.AggregatedStats` (*data*)
Bases: `cassiopeia.type.core.common.CassiopeiaObject`

assists

Returns the total number of assists this participant has had

Return type `int`

average_assists

Returns dominion only

Return type `float`

average_combat_score

Returns dominion only. the part of your score in dominion that comes from combat-related activities

Return type `float`

average_deaths

Returns dominion only

Return type `float`

average_kills

Returns dominion only

Return type `float`

average_node_capture_assists

Returns dominion only

Return type `float`

average_node_captures

Returns dominion only

Return type `float`

average_node_neutralization_assists

Returns dominion only

Return type `float`

average_node_neutralizations

Returns dominion only

Return type `float`

average_objective_score

Returns dominion only. the part of your score in dominion that comes from object-based activities

Return type `float`

average_score

Returns dominion only

Return type `float`

average_team_score

Returns dominion only

Return type `float`

bot_games

Returns the number of bot games the participant has played

Return type `int`

damage_dealt

Returns the total amount of damage this participant has dealt

Return type `int`

damage_taken

Returns the total amount of damage this participant has taken

Return type `int`

deaths

Returns the total number of deaths this participant has had

Return type `int`

double_kills

Returns the total number of double kills this participant has had

Return type `int`

dto_type

alias of *AggregatedStats*

first_bloods

Returns the total number of first bloods this participant has had

Return type `int`

games_played

Returns the total number of games this participant has played

Return type `int`

gold_earned

Returns the total amount of gold earned this participant has had

Return type `int`

healing_done

Returns the total amount of healing this participant has done

Return type `int`

kda

Returns the participant's kda

Return type `float`

killings_sprees

Returns how many killing spree the participant has had

Return type `int`

kills

Returns the total number of champion kills this participant has had

Return type `int`

losses

Returns how many loses this participant has had

Return type `int`

magic_damage_dealt

Returns the total amount of magic damage this participant has dealt

Return type `int`

max_assists

Returns dominion only. the most assists the participant has ever had

Return type `int`

max_combat_score

Returns dominion only. the highest combat score the participant has ever had

Return type `int`

max_crit

Returns the highest damage crit the participant has ever had

Return type `int`

max_deaths

Returns only returned for ranked statistics. the most deaths the participant has ever had

Return type `int`

max_game_time

Returns the longest a participant has ever been in a game

Return type `int`

max_killing_spree

Returns the largest killing spree the participant has ever had

Return type `int`

max_kills

Returns the most kills the participant has ever had

Return type `int`

max_kills_per_session

Returns well, we don't know what this one is. let us know if you figure it out.

Return type `int`

max_node_capture_assists

Returns dominion only. the most node capture assists the participant has ever had

Return type `int`

max_node_captures

Returns dominion only. the most node captures the participant has ever had

Return type `int`

max_node_neutralizations

Returns dominion only. the most node neutralizations the participant has ever had

Return type `int`

max_node_neutralize_assist

Returns dominion only. the most node neutralization assists the participant has ever had

Return type `int`

max_objective_score

Returns dominion only. the highest object score the participant has ever had

Return type `int`

max_score

Returns dominion only. the highest dominion score the participant has ever obtained

Return type `int`

max_spells_cast

Returns the most spell casts the participant has ever done in a game

Return type `int`

max_team_score

Returns dominion only. the highest team score the participant has ever had

Return type `int`

max_time_alive

Returns the longest a participant has ever been alive

Return type `int`

minions_killed

Returns the total number of minion kills this participant has had

Return type `int`

neutral_monster_killed

Returns the total number of neutral monster kills this participant has had

Return type `int`

node_captures

Returns dominion only. the total number of nodes this participant has captured

Return type `int`

node_neutralizations

Returns dominion only. the total number of nodes this participant has neutralized

Return type `int`

normal_games

Returns the number of normal games the participant has played

Return type `int`

penta_kills

Returns the total number of penta kills this participant has gotten

Return type `int`

physical_damage_dealt

Returns the total amount of physical damage this participant has dealt

Return type `int`

quadra_kills

Returns the total number of quadra kills this participant has gotten

Return type `int`

ranked_premade_games

Returns how many premade, ranked games the participant has played

Return type `int`

ranked_solo_games

Returns how many premade, solo games the participant has played

Return type `int`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

triple_kills

Returns the total number of triple kills this participant has gotten

Return type `int`

turrets_killed

Returns the total number of turrets this participant has killed

Return type `int`

unreal_kills

Returns the total number of unreal kills this participant has gotten

Return type `int`

wins

Returns how many wins this participant has had

Return type `int`

class `cassiopeia.type.core.stats.StatsSummary` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

dto_type

alias of `PlayerStatsSummary`

losses

Returns how many loses this participant has

Return type `int`

modify_date

Returns the date when the stats were last updated (in epoch milliseconds)

Return type `datetime`

stats

Returns the aggregated stats (contains pretty much every stat you probably want to access)

Return type `AggregatedStats`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

type

Returns the identifier for what queue this stat summary is for

Return type `StatSummaryType`

wins

Returns how many wins this participant has

Return type `int`

class `cassiopeia.type.core.status.Incident` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

active

Returns whether or not this incident is active

Return type `boolean`

created

Returns when this message was created

Return type `datetime.datetime`

dto_type

alias of `Incident`

id

Returns the id of this message

Return type `int`

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

updates

Returns the updates associated with this incident

Return type list<Message>

class `cassiopeia.type.core.status.Message` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

author

Returns who wrote this message

Return type str

content

Returns the content of this message

Return type str

created

Returns when this incident was created

Return type datetime.datetime

dto_type

alias of `Message`

id

Returns the id of this incident

Return type int

severity

Returns the severity of this message

Return type str

to_json (***kwargs*)

Parameters **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type

translations

Returns Translation>: the translated text of this message

Return type dict<translation.locale

updated

Returns when this message was last updated

Return type datetime.datetime

class `cassiopeia.type.core.status.Service` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

dto_type

alias of `Service`

incidents

Returns the incidents associated with this server

Return type list<Incident>

name**Returns** the name of this service**Return type** `str`**slug****Returns** the name of the service in lowercase**Return type** `str`**status****Returns** the status of the service**Return type** `str`**to_json** (***kwargs*)**Parameters** **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type**class** `cassiopeia.type.core.status.Shard` (*data*)Bases: `cassiopeia.type.core.common.CassiopeiaObject`**dto_type**alias of `Shard`**host_name****Returns** the domain name of the server**Return type** `str`**locales****Returns** the languages that you can have api results in**Return type** `list<str>`**name****Returns** the name of this service**Return type** `str`**platform****Returns** the platform (i.e. server) for this match**Return type** `Platform`**region****Returns** the region of the server is located in**Return type** `Region`**to_json** (***kwargs*)**Parameters** **data** (`CassiopeiaDto`) – the underlying DTO object with the data for this type**class** `cassiopeia.type.core.status.ShardStatus` (*data*)Bases: `cassiopeia.type.core.common.CassiopeiaObject`**dto_type**alias of `ShardStatus`**host_name**

Returns the domain name of the server

Return type `str`

locales

Returns the languages that you can have api results in

Return type `list<str>`

name

Returns the full name of the region the server is located in

Return type `str`

platform

Returns the platform (i.e. server) for this match

Return type *Platform*

region

Returns the region of the server is located in

Return type *Region*

services

Returns the services that this region offers

Return type `list<Service>`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class `cassiopeia.type.core.status.Translation` (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

content

Returns the content of this translation

Return type `str`

dto_type

alias of *Translation*

locale

Returns the content of this translation

Return type `str`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

updated

Returns when this translation was last updated

Return type `datetime.datetime`

class `cassiopeia.type.core.summoner.MasteryPage` (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

current

Returns whether or not this mastery page is active

Return type `bool`

dto_type

alias of *MasteryPage*

id

Returns the mastery page's id

Return type `int`

masteries

Returns this mastery page's masteries

Return type `list<Mastery>`

name

Returns the mastery page's name

Return type `str`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.summoner.RunePage* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

current

Returns whether or not this rune page is active

Return type `bool`

dto_type

alias of *RunePage*

id

Returns the rune page's id

Return type `int`

name

Returns the rune page's name

Return type `str`

runes

Returns the runes in this rune page

Return type `list<Rune>`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.summoner.Summoner* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

champion_masteries ()

Returns the date this summoner was last modified specified as epoch milliseconds. The following events will update this timestamp: profile icon change, playing the tutorial or advanced tutorial, finishing a game, summoner name change

Return type `datetime`

champion_mastery (*champion*)

Returns the ID of the summoner icon associated with the summoner

Return type `int`

champion_mastery_score ()

Returns the Summoner's level

Return type `int`

current_game ()

Gets the game the summoner is currently in, if they're in one

Returns the game they're in (or None if they aren't in one)

Return type `Game`

dto_type

alias of `Summoner`

id

Returns the summoner's id

Return type `int`

league_entries ()

Returns the id of the mastery page for this summoner

Return type `int`

leagues ()

Returns whether or not this mastery page is active

Return type `bool`

level

Returns the Summoner's level

Return type `int`

mastery_pages ()

Returns the runes in this rune page

Return type `list<Rune>`

match_list (*num_matches=0, begin_index=0, begin_time=0, end_time=0, champions=None, ranked_queues=None, seasons=None*)

Parameters

- **num_matches** (*int*) – the maximum number of matches to retrieve. 0 will get as many as possible. (default 0)
- **begin_index** (*int*) – the game index to start from (default 0)
- **begin_time** (*int | datetime*) – the begin time to use for fetching games (default 0)

- **end_time** (*int* | *datetime*) – the end time to use for fetching games (default 0)
- **champions** (*Champion* | *list<Champion>*) – the champion(s) to limit the results to (default None)
- **Queue** | **list<Queue>** (*ranked_queues*) – the ranked queue(s) to limit the results to (default None)
- **seasons** (*Season* | *list<Season>*) – the season(s) to limit the results to (default None)

Returns the name of this summoner’s mastery page

Return type `str`

modify_date

Returns the date this summoner was last modified specified as epoch milliseconds. The following events will update this timestamp: profile icon change, playing the tutorial or advanced tutorial, finishing a game, summoner name change

Return type `datetime`

name

Returns the summoner’s name

Return type `str`

profile_icon_id

Returns the ID of the summoner icon associated with the summoner

Return type `int`

ranked_stats (*season=None*)

Returns the summoner’s id

Return type `int`

recent_games ()

Returns the id of this summoner’s rune page

Return type `int`

rune_pages ()

Returns the name of this summoner’s rune page

Return type `str`

stats (*season=None*)

Returns the summoner’s name

Return type `str`

teams ()

Returns this mastery page’s masteries

Return type `list<Mastery>`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

top_champion_masterships (*max_entries=3*)

Gets the game the summoner is currently in, if they're in one

Returns the game they're in (or None if they aren't in one)

Return type *Game*

class `cassiopeia.type.core.team.MatchSummary` (*data*)

Bases: `cassiopeia.type.core.common.CassiopeiaObject`

assists

Returns the number of assists the team had

Return type `int`

date

Returns the date that match was completed specified as epoch milliseconds

Return type `datetime`

deaths

Returns the number of deaths the team had

Return type `int`

dto_type

alias of `MatchHistorySummary`

id

Returns the date and time for the team's last game in epoch milliseconds

Return type `datetime`

invalid

Returns whether or not the data is valid?

Return type `bool`

kda

Returns the participant's kda

Return type `float`

kills

Returns the number of kills the team had

Return type `int`

map

Returns the map that the game was played on

Return type *Map*

match ()

Returns the date and time for the team's last game in epoch milliseconds

Return type `datetime`

mode

Returns the game mode of the match

Return type *GameMode*

opponent

Returns the name of the opposing team

Return type *str*

opponent_kills

Returns the number of kills that the opponent had

Return type *int*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

win

Returns whether or not the team won this match

Return type *bool*

class *cassiopeia.type.core.team.Stats* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

average_games_played

Returns the average number of games played

Return type *float*

dto_type

alias of *TeamStatDetail*

losses

Returns the number of times this team has lost

Return type *int*

queue

Returns the queue type that these stats were aggregated for

Return type *Queue*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

wins

Returns the number of times this team has won

Return type *int*

class *cassiopeia.type.core.team.Team* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

captain

Returns the captain of the team (returns a summoner)

Return type *Summoner*

creation

Returns the creation date of the team

Return type `datetime`

dto_type

alias of `Team`

id

Returns the team's id

Return type `int`

last_game

Returns the date and time for the team's last game in epoch milliseconds

Return type `datetime`

last_join

Returns the date and time for when the most recent team member joined in epoch milliseconds

Return type `datetime`

last_queue

Returns the date the team last joined the ranked team queue in epoch milliseconds

Return type `datetime`

league_entries()

Returns the team's id

Return type `int`

leagues()

Returns the creation date of the team

Return type `datetime`

match_history

Returns the match history of the team

Return type `list<MatchSummary>`

modify

Returns the date that team was last modified specified as epoch milliseconds

Return type `datetime`

name

Returns the name of the team

Return type `str`

roster

Returns the team members

Return type `list<TeamMember>`

second_to_last_join

Returns the date the second to last member joined specified as epoch milliseconds

Return type `datetime`

stats

Returns the team's stats

Return type *Stats*

status

Returns the status of the team

Return type *str*

tag

Returns the team's tag

Return type *str*

third_to_last_join

Returns the date the third to last member joined specified as epoch milliseconds

Return type *datetime*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.team.TeamMember* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

dto_type

alias of *TeamMemberInfo*

invite

Returns the date this team member was invited to team specified as epoch milliseconds

Return type *datetime*

join

Returns the date this team member joined the team specified as epoch milliseconds

Return type *datetime*

status

Returns the date and time for when the most recent team member joined in epoch milliseconds

Return type *datetime*

summoner

Returns the summoner associated with this team member

Return type *Summoner*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

class *cassiopeia.type.core.tournament.LobbyEvent* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

dto_type

alias of *LobbyEvent*

summoner

Returns the summoner that triggered the event

Return type *Summoner*

timestamp

Returns the time that the event occurred

Return type *str*

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

type

Returns the type of the event

Return type *str*

class *cassiopeia.type.core.tournament.MapType*

Bases: *enum.Enum*

crystal_scar = <*MapType.crystal_scar*: 'CRYSTAL_SCAR'>

howling_abyss = <*MapType.howling_abyss*: 'HOWLING_ABYSS'>

summoners_rift = <*MapType.summoners_rift*: 'SUMMONERS_RIFT'>

twisted_treeline = <*MapType.twisted_treeline*: 'TWISTED_TREELINE'>

class *cassiopeia.type.core.tournament.PickType*

Bases: *enum.Enum*

blind = <*PickType.blind*: 'BLIND_PICK'>

draft = <*PickType.draft*: 'DRAFT_MODE'>

random = <*PickType.random*: 'ALL_RANDOM'>

tournament_draft = <*PickType.tournament_draft*: 'TOURNAMENT_DRAFT'>

class *cassiopeia.type.core.tournament.SpectatorType*

Bases: *enum.Enum*

all = <*SpectatorType.all*: 'ALL'>

lobby = <*SpectatorType.lobby*: 'LOBBYONLY'>

none = <*SpectatorType.none*: 'NONE'>

class *cassiopeia.type.core.tournament.TournamentCode* (*data*)

Bases: *cassiopeia.type.core.common.CassiopeiaObject*

code

Returns the tournament code

Return type *str*

dto_type

alias of *TournamentCode*

id

Returns the tournament code's ID

Return type *int*

map

Returns the map for the game

Return type `str`

meta_data

Returns the metadata for the game

Return type `str`

name

Returns the lobby name

Return type `str`

participants

Returns the summoners participating in the tournament

Return type `list<Summoner>`

password

Returns the password for the lobby

Return type `str`

pick_type

Returns the pick mode for the game

Return type *PickType*

provider_id

Returns the provider's ID

Return type `int`

region

Returns the tournament's region

Return type *TournamentRegion*

spectator_type

Returns the spectator mode for the game

Return type *SpectatorType*

team_size

Returns the team size for the game

Return type `int`

to_json (***kwargs*)

Parameters **data** (*CassiopeiaDto*) – the underlying DTO object with the data for this type

tournament_id

Returns the tournament's ID

Return type `int`

class `cassiopeia.type.core.tournament.TournamentRegion`

Bases: `enum.Enum`

```
brazil = <TournamentRegion.brazil: 'BR'>
europe_north_east = <TournamentRegion.europe_north_east: 'EUNE'>
europe_west = <TournamentRegion.europe_west: 'EUW'>
japan = <TournamentRegion.japan: 'JP'>
korea = <TournamentRegion.korea: 'KR'>
latin_america_north = <TournamentRegion.latin_america_north: 'LAN'>
latin_america_south = <TournamentRegion.latin_america_south: 'LAS'>
north_america = <TournamentRegion.north_america: 'NA'>
oceania = <TournamentRegion.oceania: 'OCE'>
pbe = <TournamentRegion.pbe: 'PBE'>
russia = <TournamentRegion.russia: 'RU'>
turkey = <TournamentRegion.turkey: 'TR'>
```

3.2.3 cassiopeia.type.dto

```
class cassiopeia.type.dto.champion.Champion (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **active** (*bool*) – indicates if the champion is active
- **botEnabled** (*bool*) – bot enabled flag (for custom games)
- **botMmEnabled** (*bool*) – bot Match Made enabled flag (for Co-op vs. AI games)
- **freeToPlay** (*bool*) – indicates if the champion is free to play. Free to play champions are rotated periodically
- **id** (*int*) – champion ID. For static information correlating to champion IDs, please refer to the LoL Static Data API.
- **rankedPlayEnabled** (*bool*) – ranked play enabled flag

```
to_json (**kwargs)
```

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.champion.ChampionList (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters **champions** (*list<Champion>*) – the collection of champion information

champion_ids

Gets all champion IDs contained in this object

```
to_json (**kwargs)
```

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.championmastery.ChampionMastery (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **championId** (*int*) – champion ID for this entry

- **championLevel** (*int*) – champion level for specified player and champion combination
- **championPoints** (*int*) – total number of champion points for this player and champion combination - they are used to determine championLevel
- **championPointsSinceLastLevel** (*int*) – number of points earned since current level has been achieved. Zero if player reached maximum champion level for this champion.
- **championPointsUntilNextLevel** (*int*) – number of points needed to achieve next level. Zero if player reached maximum champion level for this champion.
- **chestGranted** (*bool*) – is chest granted for this champion or not in current season
- **lastPlayTime** (*int*) – last time this champion was played by this player - in Unix milliseconds time format
- **playerId** (*int*) – player ID for this entry
- **tokensEarned** (*int*) – number of token earned for next level mastery

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

`cassiopeia.type.dto.common.BaseDB`
alias of Base

class `cassiopeia.type.dto.common.CassiopeiaDto` (*dictionary*)
Bases: `object`

A Python representation of an object returned by the RiotAPI

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.common.CassiopeiaParametersDto` (*dictionary*)
Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets a JSON representation of the object

Returns a JSON representation of the object

Return type `str`

to_json (***kwargs*)

Gets a JSON representation of the object

Returns a JSON representation of the object

Return type `str`

class `cassiopeia.type.dto.common.JSONEncoded` (**args, **kwargs*)
Bases: `sqlalchemy.sql.type_api.TypeDecorator`

Parameters for a POST or PUT request to the Riot API

class `Comparator` (*expr*)
Bases: `sqlalchemy.sql.type_api.Comparator`

BOOLEANTYPE = `Boolean()`

__add__ (*other*)

Implement the + operator.

In a column context, produces the clause `a + b` if the parent object has non-string affinity. If the parent object has a string affinity, produces the concatenation operator, `a || b` - see `ColumnOperators.concat()`.

__and__ (*other*)

Implement the `&` operator.

When used with SQL expressions, results in an AND operation, equivalent to `and_()`, that is:

```
a & b
```

is equivalent to:

```
from sqlalchemy import and_
and_(a, b)
```

Care should be taken when using `&` regarding operator precedence; the `&` operator has the highest precedence. The operands should be enclosed in parenthesis if they contain further sub expressions:

```
(a == 2) & (b == 4)
```

__div__ (*other*)

Implement the `/` operator.

In a column context, produces the clause `a / b`.

__eq__ (*other*)

Implement the `==` operator.

In a column context, produces the clause `a = b`. If the target is `None`, produces `a IS NULL`.

__ge__ (*other*)

Implement the `>=` operator.

In a column context, produces the clause `a >= b`.

__getitem__ (*index*)

Implement the `[]` operator.

This can be used by some database-specific types such as PostgreSQL `ARRAY` and `HSTORE`.

__gt__ (*other*)

Implement the `>` operator.

In a column context, produces the clause `a > b`.

__invert__ ()

Implement the `~` operator.

When used with SQL expressions, results in a NOT operation, equivalent to `not_()`, that is:

```
~a
```

is equivalent to:

```
from sqlalchemy import not_
not_(a)
```

__le__ (*other*)

Implement the `<=` operator.

In a column context, produces the clause `a <= b`.

`__lshift__(other)`

implement the << operator.

Not used by SQLAlchemy core, this is provided for custom operator systems which want to use << as an extension point.

`__lt__(other)`

Implement the < operator.

In a column context, produces the clause `a < b`.

`__mod__(other)`

Implement the % operator.

In a column context, produces the clause `a % b`.

`__mul__(other)`

Implement the * operator.

In a column context, produces the clause `a * b`.

`__ne__(other)`

Implement the != operator.

In a column context, produces the clause `a != b`. If the target is None, produces `a IS NOT NULL`.

`__neg__()`

Implement the - operator.

In a column context, produces the clause `-a`.

`__or__(other)`

Implement the | operator.

When used with SQL expressions, results in an OR operation, equivalent to `or_()`, that is:

```
a | b
```

is equivalent to:

```
from sqlalchemy import or_
or_(a, b)
```

Care should be taken when using | regarding operator precedence; the | operator has the highest precedence. The operands should be enclosed in parenthesis if they contain further sub expressions:

```
(a == 2) | (b == 4)
```

`__radd__(other)`

Implement the + operator in reverse.

See `ColumnOperators.__add__()`.

`__rdiv__(other)`

Implement the / operator in reverse.

See `ColumnOperators.__div__()`.

`__rmod__(other)`

Implement the % operator in reverse.

See `ColumnOperators.__mod__()`.

__rmul__ (*other*)

Implement the * operator in reverse.

See `ColumnOperators.__mul__()`.

__rshift__ (*other*)

implement the >> operator.

Not used by SQLAlchemy core, this is provided for custom operator systems which want to use >> as an extension point.

__rsub__ (*other*)

Implement the - operator in reverse.

See `ColumnOperators.__sub__()`.

__rtruediv__ (*other*)

Implement the // operator in reverse.

See `ColumnOperators.__truediv__()`.

__sub__ (*other*)

Implement the - operator.

In a column context, produces the clause `a - b`.

__truediv__ (*other*)

Implement the // operator.

In a column context, produces the clause `a / b`.

all_ ()

Produce a `all_()` clause against the parent object.

New in version 1.1.

any_ ()

Produce a `any_()` clause against the parent object.

New in version 1.1.

asc ()

Produce a `asc()` clause against the parent object.

between (*clef*, *crigh*, *symmetric=False*)

Produce a `between()` clause against the parent object, given the lower and upper range.

collate (*collation*)

Produce a `collate()` clause against the parent object, given the collation string.

concat (*other*)

Implement the 'concat' operator.

In a column context, produces the clause `a || b`, or uses the `concat()` operator on MySQL.

contains (*other*, ***kwargs*)

Implement the 'contains' operator.

In a column context, produces the clause `LIKE '%<other>%'`

default_comparator = `None`

desc ()

Produce a `desc()` clause against the parent object.

distinct()

Produce a `distinct()` clause against the parent object.

endswith(*other*, *kwargs*)**

Implement the ‘endswith’ operator.

In a column context, produces the clause `LIKE '%<other>'`

expr

ilike(*other*, *escape=None*)

Implement the `ilike` operator.

In a column context, produces the clause `a ILIKE other`.

E.g.:

```
select ([sometable]).where(sometable.c.column.ilike("%foobar%"))
```

Parameters

- **other** – expression to be compared
- **escape** – optional escape character, renders the `ESCAPE` keyword, e.g.:

```
somecolumn.ilike("foo/%bar", escape="/")
```

See also:

`ColumnOperators.like()`

in_(*other*)

Implement the `in` operator.

In a column context, produces the clause `a IN other`. “other” may be a tuple/list of column expressions, or a `select()` construct.

is_(*other*)

Implement the `IS` operator.

Normally, `IS` is generated automatically when comparing to a value of `None`, which resolves to `NULL`. However, explicit usage of `IS` may be desirable if comparing to boolean values on certain platforms.

New in version 0.7.9.

See also:

`ColumnOperators.isnot()`

is_distinct_from(*other*)

Implement the `IS DISTINCT FROM` operator.

Renders “a IS DISTINCT FROM b” on most platforms; on some such as SQLite may render “a IS NOT b”.

New in version 1.1.

isnot(*other*)

Implement the `IS NOT` operator.

Normally, `IS NOT` is generated automatically when comparing to a value of `None`, which resolves to `NULL`. However, explicit usage of `IS NOT` may be desirable if comparing to boolean values on certain platforms.

New in version 0.7.9.

See also:`ColumnOperators.is_()`**isnot_distinct_from**(*other*)

Implement the IS NOT DISTINCT FROM operator.

Renders “a IS NOT DISTINCT FROM b” on most platforms; on some such as SQLite may render “a IS b”.

New in version 1.1.

like(*other*, *escape=None*)

Implement the like operator.

In a column context, produces the clause a LIKE other.

E.g.:

```
select ([sometable]).where(sometable.c.column.like("%foobar%"))
```

Parameters

- **other** – expression to be compared
- **escape** – optional escape character, renders the ESCAPE keyword, e.g.:

```
somecolumn.like("foo/%bar", escape="/")
```

See also:`ColumnOperators.ilike()`**match**(*other*, ***kwargs*)

Implements a database-specific ‘match’ operator.

`match()` attempts to resolve to a MATCH-like function or operator provided by the backend. Examples include:

- PostgreSQL - renders `x @@ to_tsquery(y)`
- MySQL - renders `MATCH (x) AGAINST (y IN BOOLEAN MODE)`
- Oracle - renders `CONTAINS(x, y)`
- other backends may provide special implementations.
- Backends without any special implementation will emit the operator as “MATCH”. This is compatible with SQLite, for example.

notilike(*other*, *escape=None*)

implement the NOT ILIKE operator.

This is equivalent to using negation with `ColumnOperators.ilike()`, i.e. `~x.ilike(y)`.

New in version 0.8.

See also:`ColumnOperators.ilike()`**notin_**(*other*)

implement the NOT IN operator.

This is equivalent to using negation with `ColumnOperators.in_()`, i.e. `~x.in_(y)`.

New in version 0.8.

See also:

`ColumnOperators.in_()`

notlike (*other, escape=None*)
implement the NOT LIKE operator.

This is equivalent to using negation with `ColumnOperators.like()`, i.e. `~x.like(y)`.

New in version 0.8.

See also:

`ColumnOperators.like()`

nullsfirst ()
Produce a `nullsfirst()` clause against the parent object.

nullslast ()
Produce a `nullslast()` clause against the parent object.

op (*opstring, precedence=0, is_comparison=False*)
produce a generic operator function.

e.g.:

```
somecolumn.op("*")(5)
```

produces:

```
somecolumn * 5
```

This function can also be used to make bitwise operators explicit. For example:

```
somecolumn.op('&')(0xff)
```

is a bitwise AND of the value in `somecolumn`.

Parameters

- **operator** – a string which will be output as the infix operator between this element and the expression passed to the generated function.
- **precedence** – precedence to apply to the operator, when parenthesizing expressions. A lower number will cause the expression to be parenthesized when applied against another operator with higher precedence. The default value of 0 is lower than all operators except for the comma (,) and AS operators. A value of 100 will be higher or equal to all operators, and -100 will be lower than or equal to all operators.

New in version 0.8: - added the ‘precedence’ argument.

- **is_comparison** – if True, the operator will be considered as a “comparison” operator, that is which evaluates to a boolean true/false value, like `==`, `>`, etc. This flag should be set so that ORM relationships can establish that the operator is a comparison operator when used in a custom join condition.

New in version 0.9.2: - added the [:paramref:‘.Operators.op.is_comparison’](#) flag.

See also:

`types_operators`

`relationship_custom_operator`

operate (*op*, **other*, ***kwargs*)

reverse_operate (*op*, *other*, ***kwargs*)

startswith (*other*, ***kwargs*)

Implement the `startswith` operator.

In a column context, produces the clause `LIKE '<other>%'`

timetuple = `None`

type

`JSONEncoded.__getattr__` (*key*)

Proxy all other undefined accessors to the underlying implementation.

`JSONEncoded.adapt` (*cls*, ***kw*)

Produce an “adapted” form of this type, given an “impl” class to work with.

This method is used internally to associate generic types with “implementation” types that are specific to a particular dialect.

`JSONEncoded.bind_expression` (*bindvalue*)

“Given a bind value (i.e. a `BindParameter` instance), return a SQL expression in its place.

This is typically a SQL function that wraps the existing bound parameter within the statement. It is used for special data types that require literals being wrapped in some special database function in order to coerce an application-level value into a database-specific format. It is the SQL analogue of the `TypeEngine.bind_processor()` method.

The method is evaluated at statement compile time, as opposed to statement construction time.

Note that this method, when implemented, should always return the exact same structure, without any conditional logic, as it may be used in an `executemany()` call against an arbitrary number of bound parameter sets.

See also:

`types_sql_value_processing`

`JSONEncoded.bind_processor` (*dialect*)

Provide a bound value processing function for the given `Dialect`.

This is the method that fulfills the `TypeEngine` contract for bound value conversion. `TypeDecorator` will wrap a user-defined implementation of `process_bind_param()` here.

User-defined code can override this method directly, though its likely best to use `process_bind_param()` so that the processing provided by `self.impl` is maintained.

Parameters `dialect` – `Dialect` instance in use.

This method is the reverse counterpart to the `result_processor()` method of this class.

`JSONEncoded.coerce_compared_value` (*op*, *value*)

Suggest a type for a ‘coerced’ Python value in an expression.

By default, returns self. This method is called by the expression system when an object using this type is on the left or right side of an expression against a plain Python object which does not yet have a SQLAlchemy type assigned:

```
expr = table.c.somecolumn + 35
```

Where above, if `somecolumn` uses this type, this method will be called with the value `operator.add` and `35`. The return value is whatever SQLAlchemy type should be used for `35` for this particular operation.

`JSONEncoded.coerce_to_is_types = (<class 'NoneType'>,)`

`JSONEncoded.column_expression (colexpr)`

Given a SELECT column expression, return a wrapping SQL expression.

This is typically a SQL function that wraps a column expression as rendered in the columns clause of a SELECT statement. It is used for special data types that require columns to be wrapped in some special database function in order to coerce the value before being sent back to the application. It is the SQL analogue of the `TypeEngine.result_processor()` method.

The method is evaluated at statement compile time, as opposed to statement construction time.

See also:

`types_sql_value_processing`

`JSONEncoded.comparator_factory`

`JSONEncoded.compare_against_backend (dialect, conn_type)`

Compare this type against the given backend type.

This function is currently not implemented for SQLAlchemy types, and for all built in types will return `None`. However, it can be implemented by a user-defined type where it can be consumed by schema comparison tools such as Alembic autogenerate.

A future release of SQLAlchemy will potentially impement this method for builtin types as well.

The function should return `True` if this type is equivalent to the given type; the type is typically reflected from the database so should be database specific. The dialect in use is also passed. It can also return `False` to assert that the type is not equivalent.

Parameters

- **dialect** – a `Dialect` that is involved in the comparison.
- **conn_type** – the type object reflected from the backend.

New in version 1.0.3.

`JSONEncoded.compare_values (x, y)`

Given two values, compare them for equality.

By default this calls upon `TypeEngine.compare_values()` of the underlying “impl”, which in turn usually uses the Python equals operator `==`.

This function is used by the ORM to compare an original-loaded value with an intercepted “changed” value, to determine if a net change has occurred.

`JSONEncoded.compile (dialect=None)`

Produce a string-compiled form of this `TypeEngine`.

When called with no arguments, uses a “default” dialect to produce a string result.

Parameters **dialect** – a `Dialect` instance.

`JSONEncoded.copy (**kw)`

Produce a copy of this `TypeDecorator` instance.

This is a shallow copy and is provided to fulfill part of the `TypeEngine` contract. It usually does not need to be overridden unless the user-defined `TypeDecorator` has local state that should be deep-copied.

`JSONEncoded.copy_value (value)`

`JSONEncoded.dialect_impl (dialect)`

Return a dialect-specific implementation for this `TypeEngine`.

`JSONEncoded.dispatch = <sqlalchemy.event.base.DDLEventsDispatch object>`

`JSONEncoded.evaluates_none()`

Return a copy of this type which has the `should_evaluate_none` flag set to True.

E.g.:

```
Table(
    'some_table', metadata,
    Column(
        String(50).evaluates_none(),
        nullable=True,
        server_default='no value')
)
```

The ORM uses this flag to indicate that a positive value of None is passed to the column in an INSERT statement, rather than omitting the column from the INSERT statement which has the effect of firing off column-level defaults. It also allows for types which have special behavior associated with the Python None value to indicate that the value doesn't necessarily translate into SQL NULL; a prime example of this is a JSON type which may wish to persist the JSON value 'null'.

In all cases, the actual NULL SQL value can be always be persisted in any column by using the null SQL construct in an INSERT statement or associated with an ORM-mapped attribute.

Note: The “evaluates none” flag does **not** apply to a value of None passed to **:paramref:'.Column.default'** or **:paramref:'.Column.server_default'**; in these cases, None still means “no default”.

New in version 1.1.

See also:

`session_forcing_null` - in the ORM documentation

:paramref:'.postgresql.JSON.none_as_null' - PostgreSQL JSON interaction with this flag.

`TypeEngine.should_evaluate_none` - class-level flag

`JSONEncoded.get_dbapi_type(dbapi)`

Return the DBAPI type object represented by this TypeDecorator.

By default this calls upon `TypeEngine.get_dbapi_type()` of the underlying “impl”.

`JSONEncoded.hashable = True`

`JSONEncoded.impl`
alias of `Text`

`JSONEncoded.literal_processor(dialect)`

Provide a literal processing function for the given `Dialect`.

Subclasses here will typically override `TypeDecorator.process_literal_param()` instead of this method directly.

By default, this method makes use of `TypeDecorator.process_bind_param()` if that method is implemented, where `TypeDecorator.process_literal_param()` is not. The rationale here is that `TypeDecorator` typically deals with Python conversions of data that are above the layer of database presentation. With the value converted by `TypeDecorator.process_bind_param()`, the underlying type will then handle whether it needs to be presented to the DBAPI as a bound parameter or to the database as an inline SQL value.

New in version 0.9.0.

`JSONEncoded.load_dialect_impl(dialect)`

Return a `TypeEngine` object corresponding to a dialect.

This is an end-user override hook that can be used to provide differing types depending on the given dialect. It is used by the `TypeDecorator` implementation of `type_engine()` to help determine what type should ultimately be returned for a given `TypeDecorator`.

By default returns `self.impl`.

`JSONEncoded.process_bind_param(value, dialect)`

`JSONEncoded.process_literal_param(value, dialect)`

Receive a literal parameter value to be rendered inline within a statement.

This method is used when the compiler renders a literal value without using binds, typically within DDL such as in the “server default” of a column or an expression within a `CHECK` constraint.

The returned string will be rendered into the output string.

New in version 0.9.0.

`JSONEncoded.process_result_value(value, dialect)`

`JSONEncoded.python_type`

Return the Python type object expected to be returned by instances of this type, if known.

Basically, for those types which enforce a return type, or are known across the board to do such for all common DBAPIs (like `int` for example), will return that type.

If a return type is not defined, raises `NotImplementedError`.

Note that any type also accommodates `NULL` in SQL which means you can also get back `None` from any type in practice.

`JSONEncoded.result_processor(dialect, coltype)`

Provide a result value processing function for the given `Dialect`.

This is the method that fulfills the `TypeEngine` contract for result value conversion. `TypeDecorator` will wrap a user-defined implementation of `process_result_value()` here.

User-defined code can override this method directly, though its likely best to use `process_result_value()` so that the processing provided by `self.impl` is maintained.

Parameters

- **dialect** – Dialect instance in use.
- **coltype** – A `SQLAlchemy` data type

This method is the reverse counterpart to the `bind_processor()` method of this class.

`JSONEncoded.should_evaluate_none = False`

`JSONEncoded.type_engine(dialect)`

Return a dialect-specific `TypeEngine` instance for this `TypeDecorator`.

In most cases this returns a dialect-adapted form of the `TypeEngine` type represented by `self.impl`. Makes usage of `dialect_impl()` but also traverses into wrapped `TypeDecorator` instances. Behavior can be customized here by overriding `load_dialect_impl()`.

`JSONEncoded.with_variant(type_, dialect_name)`

Produce a new type object that will utilize the given type when applied to the dialect of the given name.

e.g.:

```
from sqlalchemy.types import String
from sqlalchemy.dialects import mysql

s = String()

s = s.with_variant(mysql.VARCHAR(collation='foo'), 'mysql')
```

The construction of `TypeEngine.with_variant()` is always from the “fallback” type to that which is dialect specific. The returned type is an instance of `Variant`, which itself provides a `Variant.with_variant()` that can be called repeatedly.

Parameters

- **type** – a `TypeEngine` that will be selected as a variant from the originating type, when a dialect of the given name is in use.
- **dialect_name** – base name of the dialect which uses this type. (i.e. 'postgresql', 'mysql', etc.)

New in version 0.7.2.

class `cassiopeia.type.dto.currentgame.BannedChampion` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **championId** (*int*) – the ID of the banned champion
- **pickTurn** (*int*) – the turn during which the champion was banned
- **teamId** (*int*) – the ID of the team that banned the champion

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.currentgame.CurrentGameInfo` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **bannedChampions** (*list<BannedChampion>*) – banned champion information
- **gameId** (*int*) – the ID of the game
- **gameLength** (*int*) – the amount of time in seconds that has passed since the game started
- **gameMode** (*str*) – the game mode (Legal values: CLASSIC, ODIN, ARAM, TUTORIAL, ONEFORALL, ASCENSION, FIRSTBLOOD, KINGPORO)
- **gameQueueConfigId** (*int*) – the queue type (queue types are documented on the Game Constants page)
- **gameStartTime** (*int*) – the game start time represented in epoch milliseconds
- **gameType** (*str*) – the game type (Legal values: CUSTOM_GAME, MATCHED_GAME, TUTORIAL_GAME)
- **mapId** (*int*) – the ID of the map
- **observers** (*Observer*) – the observer information
- **participants** (*list<CurrentGameParticipant>*) – the participant information

- **platformId** (*str*) – the ID of the platform on which the game is being played

champion_ids

Gets all champion IDs contained in this object

mastery_ids

Gets all champion IDs contained in this object

rune_ids

Gets all champion IDs contained in this object

summoner_ids

Gets all champion IDs contained in this object

summoner_spell_ids

Gets all champion IDs contained in this object

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.currentgame.CurrentGameParticipant` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **bot** (*bool*) – flag indicating whether or not this participant is a bot
- **championId** (*int*) – the ID of the champion played by this participant
- **masteries** (*list<Mastery>*) – the masteries used by this participant
- **profileIconId** (*int*) – the ID of the profile icon used by this participant
- **runes** (*list<Rune>*) – the runes used by this participant
- **spell1Id** (*int*) – the ID of the first summoner spell used by this participant
- **spell2Id** (*int*) – the ID of the second summoner spell used by this participant
- **summonerId** (*int*) – the summoner ID of this participant
- **summonerName** (*str*) – the summoner name of this participant
- **teamId** (*int*) – the team ID of this participant, indicating the participant's team

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.currentgame.Mastery` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **masteryId** (*int*) – the ID of the mastery
- **rank** (*int*) – the number of points put into this mastery by the user

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.currentgame.Observer` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters **encryptionKey** (*str*) – key used to decrypt the spectator grid game data for playback

`to_json (**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.currentgame.Rune` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **count** (*int*) – the count of this rune used by the participant
- **runeId** (*int*) – the ID of the rune

`to_json (**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.featuredgames.BannedChampion` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **championId** (*int*) – the ID of the banned champion
- **pickTurn** (*int*) – the turn during which the champion was banned
- **teamId** (*int*) – the ID of the team that banned the champion

`to_json (**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.featuredgames.FeaturedGameInfo` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **bannedChampions** (*list<BannedChampion>*) – banned champion information
- **gameId** (*int*) – the ID of the game
- **gameLength** (*int*) – the amount of time in seconds that has passed since the game started
- **gameMode** (*str*) – the game mode (Legal values: CLASSIC, ODIN, ARAM, TUTORIAL, ONEFORALL, ASCENSION, FIRSTBLOOD, KINGPORO)
- **gameQueueConfigId** (*int*) – the queue type (queue types are documented on the Game Constants page)
- **gameStartTime** (*int*) – the game start time represented in epoch milliseconds
- **gameType** (*str*) – the game type (Legal values: CUSTOM_GAME, MATCHED_GAME, TUTORIAL_GAME)
- **mapId** (*int*) – the ID of the map
- **observers** (*Observer*) – the observer information
- **participants** (*list<Participant>*) – the participant information
- **platformId** (*str*) – the ID of the platform on which the game is being played

champion_ids

Gets all champion IDs contained in this object

summoner_spell_ids

Gets all champion IDs contained in this object

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.featuredgames.FeaturedGames (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all champion IDs contained in this object

champion_ids

Gets all summoner spell IDs contained in this object

summoner_spell_ids

Gets all summoner spell IDs contained in this object

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.featuredgames.Observer (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters `encryptionKey (str)` – key used to decrypt the spectator grid game data for playback

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.featuredgames.Participant (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **bot** (`bool`) – flag indicating whether or not this participant is a bot
- **championId** (`int`) – the ID of the champion played by this participant
- **profileIconId** (`int`) – the ID of the profile icon used by this participant
- **spell1Id** (`int`) – the ID of the first summoner spell used by this participant
- **spell2Id** (`int`) – the ID of the second summoner spell used by this participant
- **summonerName** (`str`) – the summoner name of this participant
- **teamId** (`int`) – the team ID of this participant, indicating the participant's team
- **encryptionKey** (`str`) – key used to decrypt the spectator grid game data for playback
- **championId** – the ID of the banned champion
- **pickTurn** (`int`) – the turn during which the champion was banned
- **teamId** – the ID of the team that banned the champion

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.game.Game (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **championId** (`int`) – champion ID associated with game
- **createDate** (`int`) – date that end game data was recorded, specified as epoch milliseconds

- **fellowPlayers** (*list<Player>*) – other players associated with the game
- **gameId** (*int*) – game ID
- **gameMode** (*str*) – game mode (Legal values: CLASSIC, ODIN, ARAM, TUTORIAL, ONEFORALL, ASCENSION, FIRSTBLOOD, KINGPORO)
- **gameType** (*str*) – game type (Legal values: CUSTOM_GAME, MATCHED_GAME, TUTORIAL_GAME)
- **invalid** (*bool*) – invalid flag
- **ipEarned** (*int*) – IP Earned
- **level** (*int*) – level
- **mapId** (*int*) – map ID
- **spell11** (*int*) – ID of first summoner spell
- **spell12** (*int*) – ID of second summoner spell
- **stats** (*RawStats*) – statistics associated with the game for this summoner
- **subType** (*str*) – game sub-type (Legal values: NONE, NORMAL, BOT, RANKED_SOLO_5x5, RANKED_PREMADE_3x3, RANKED_PREMADE_5x5, ODIN_UNRANKED, RANKED_TEAM_3x3, RANKED_TEAM_5x5, NORMAL_3x3, BOT_3x3, CAP_5x5, ARAM_UNRANKED_5x5, ONEFORALL_5x5, FIRSTBLOOD_1x1, FIRSTBLOOD_2x2, SR_6x6, URF, URF_BOT, NIGHTMARE_BOT, ASCENSION, HEXAKILL, KING_PORO, COUNTER_PICK)
- **teamId** (*int*) – team ID associated with game. Team ID 100 is blue team. Team ID 300 is purple team.

champion_ids

Gets all champion IDs contained in this object

item_ids

Gets all champion IDs contained in this object

summoner_ids

Gets all champion IDs contained in this object

summoner_spell_ids

Gets all champion IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.game.Player` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **championId** (*int*) – champion id associated with player
- **summonerId** (*int*) – summoner id associated with player
- **teamId** (*int*) – team id associated with player

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.game.RawStats` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **assists** (*int*) – number of assists
- **barracksKilled** (*int*) – number of enemy inhibitors killed
- **championsKilled** (*int*) – number of champions killed
- **combatPlayerScore** (*int*) – the combat player score
- **consumablesPurchased** (*int*) – number of consumables purchased
- **damageDealtPlayer** (*int*) – total damage dealt
- **doubleKills** (*int*) – number of double kills
- **firstBlood** (*int*) – first blood
- **gold** (*int*) – amount of gold
- **goldEarned** (*int*) – total gold earned
- **goldSpent** (*int*) – total gold spent
- **item0** (*int*) – ID of item 0
- **item1** (*int*) – ID of item 1
- **item2** (*int*) – ID of item 2
- **item3** (*int*) – ID of item 3
- **item4** (*int*) – ID of item 4
- **item5** (*int*) – ID of item 5
- **item6** (*int*) – ID of item 6
- **itemsPurchased** (*int*) – number of items purchased
- **killingSprees** (*int*) – number of killing sprees
- **largestCriticalStrike** (*int*) – largest critical strike
- **largestKillingSpree** (*int*) – largest killing spree
- **largestMultiKill** (*int*) – largest multi kill
- **legendaryItemsCreated** (*int*) – number of tier 3 items built
- **level** (*int*) – level
- **magicDamageDealtPlayer** (*int*) – total magic damage dealt
- **magicDamageDealtToChampions** (*int*) – total magic damage dealt to champions
- **magicDamageTaken** (*int*) – total magic damage taken
- **minionsDenied** (*int*) – total minions denied
- **minionsKilled** (*int*) – total minions killed
- **neutralMinionsKilled** (*int*) – total neutral minions killed
- **neutralMinionsKilledEnemyJungle** (*int*) – neutral minions killed in enemy jungle
- **neutralMinionsKilledYourJungle** (*int*) – neutral minions killed in own jungle

- **nexusKilled** (*bool*) – flag specifying if the summoner got the killing blow on the nexus
- **nodeCapture** (*int*) – number of nodes captured
- **nodeCaptureAssist** (*int*) – number of node capture assists
- **nodeNeutralize** (*int*) – number of nodes neutralized
- **nodeNeutralizeAssist** (*int*) – number of node neutralization assists
- **numDeaths** (*int*) – number of deaths
- **numItemsBought** (*int*) – number of items bought
- **objectivePlayerScore** (*int*) – objective player score
- **pentaKills** (*int*) – number of penta kills
- **physicalDamageDealtPlayer** (*int*) – total physical damage dealt
- **physicalDamageDealtToChampions** (*int*) – total physical damage dealt to champions
- **physicalDamageTaken** (*int*) – total physical damage taken
- **playerPosition** (*int*) – player position
- **playerRole** (*int*) – player role
- **quadraKills** (*int*) – number of quadra kills
- **sightWardsBought** (*int*) – number of sight wards bought
- **spell1Cast** (*int*) – number of times first champion spell was cast
- **spell2Cast** (*int*) – number of times second champion spell was cast
- **spell3Cast** (*int*) – number of times third champion spell was cast
- **spell4Cast** (*int*) – number of times fourth champion spell was cast
- **summonSpell1Cast** (*int*) – number of times summoner spell 1 was cast
- **summonSpell2Cast** (*int*) – number of times summoner spell 2 was cast
- **superMonsterKilled** (*int*) – number of super monsters killed
- **team** (*int*) – team
- **teamObjective** (*int*) – team objectives
- **timePlayed** (*int*) – time played
- **totalDamageDealt** (*int*) – total damage dealt
- **totalDamageDealtToChampions** (*int*) – total damage dealt to champions
- **totalDamageTaken** (*int*) – total damage taken
- **totalHeal** (*int*) – total healing done
- **totalPlayerScore** (*int*) – total player score
- **totalScoreRank** (*int*) – total score rank
- **totalTimeCrowdControlDealt** (*int*) – total crowd control time dealt
- **totalUnitsHealed** (*int*) – number of units healed
- **tripleKills** (*int*) – number of triple kills

- **trueDamageDealtPlayer** (*int*) – total true damage dealt
- **trueDamageDealtToChampions** (*int*) – total true damage dealt to champions
- **trueDamageTaken** (*int*) – total true damage taken
- **turretsKilled** (*int*) – number of turrets killed
- **unrealKills** (*int*) – number of unreal kills
- **victoryPointTotal** (*int*) – total victory points
- **visionWardsBought** (*int*) – number of vision wards bought
- **wardKilled** (*int*) – number of wards killed
- **wardPlaced** (*int*) – number of wards placed
- **win** (*bool*) – flag specifying whether or not this game was won

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.game.RecentGames` (*dictionary*)
 Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all champion IDs contained in this object

champion_ids

Gets all summoner IDs contained in this object

item_ids

Gets all summoner IDs contained in this object

summoner_ids

Gets all summoner IDs contained in this object

summoner_spell_ids

Gets all summoner IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.league.League` (*dictionary*)
 Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **entries** (*list*<*LeagueEntry*>) – the requested league entries
- **name** (*str*) – this name is an internal place-holder name only. Display and localization of names in the game client are handled client-side.
- **participantId** (*str*) – specifies the relevant participant that is a member of this league (i.e., a requested summoner ID, a requested team ID, or the ID of a team to which one of the requested summoners belongs). Only present when full league is requested so that participant's entry can be identified. Not present when individual entry is requested.
- **queue** (*str*) – the league's queue type (Legal values: RANKED_SOLO_5x5, RANKED_TEAM_3x3, RANKED_TEAM_5x5)
- **tier** (*str*) – the league's tier (Legal values: CHALLENGER, MASTER, DIAMOND, PLATINUM, GOLD, SILVER, BRONZE)

summoner_ids

Gets all summoner IDs contained in this object

team_ids

Gets all summoner IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.league.LeagueEntry` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **division** (*str*) – the league division of the participant
- **isFreshBlood** (*bool*) – specifies if the participant is fresh blood
- **isHotStreak** (*bool*) – specifies if the participant is on a hot streak
- **isInactive** (*bool*) – specifies if the participant is inactive
- **isVeteran** (*bool*) – specifies if the participant is a veteran
- **leaguePoints** (*int*) – the league points of the participant
- **losses** (*int*) – the number of losses for the participant
- **miniSeries** (*MiniSeries*) – mini series data for the participant. Only present if the participant is currently in a mini series.
- **playerOrTeamId** (*str*) – the ID of the participant (i.e., summoner or team) represented by this entry
- **playerOrTeamName** (*str*) – the name of the the participant (i.e., summoner or team) represented by this entry
- **wins** (*int*) – the number of wins for the participant

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.league.MiniSeries` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **losses** (*int*) – number of current losses in the mini series
- **progress** (*str*) – string showing the current, sequential mini series progress where ‘W’ represents a win, ‘L’ represents a loss, and ‘N’ represents a game that hasn’t been played yet
- **target** (*int*) – number of wins required for promotion
- **wins** (*int*) – number of current wins in the mini series

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.BannedChampion` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **frameInterval** (*int*) – time between each returned frame in milliseconds
- **frames** (*list<Frame>*) – list of timeline frames for the game

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.match.Event* (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Parameters

- **ancientGolemAssistsPerMinCounts** (*ParticipantTimelineData*) – ancient golem assists per minute timeline counts
- **ancientGolemKillsPerMinCounts** (*ParticipantTimelineData*) – ancient golem kills per minute timeline counts
- **assistedLaneDeathsPerMinDeltas** (*ParticipantTimelineData*) – assisted lane deaths per minute timeline data
- **assistedLaneKillsPerMinDeltas** (*ParticipantTimelineData*) – assisted lane kills per minute timeline data
- **baronAssistsPerMinCounts** (*ParticipantTimelineData*) – baron assists per minute timeline counts
- **baronKillsPerMinCounts** (*ParticipantTimelineData*) – baron kills per minute timeline counts
- **creepsPerMinDeltas** (*ParticipantTimelineData*) – creeps per minute timeline data
- **csDiffPerMinDeltas** (*ParticipantTimelineData*) – creep score difference per minute timeline data
- **damageTakenDiffPerMinDeltas** (*ParticipantTimelineData*) – damage taken difference per minute timeline data
- **damageTakenPerMinDeltas** (*ParticipantTimelineData*) – damage taken per minute timeline data
- **dragonAssistsPerMinCounts** (*ParticipantTimelineData*) – dragon assists per minute timeline counts
- **dragonKillsPerMinCounts** (*ParticipantTimelineData*) – dragon kills per minute timeline counts
- **elderLizardAssistsPerMinCounts** (*ParticipantTimelineData*) – elder lizard assists per minute timeline counts
- **elderLizardKillsPerMinCounts** (*ParticipantTimelineData*) – elder lizard kills per minute timeline counts
- **goldPerMinDeltas** (*ParticipantTimelineData*) – gold per minute timeline data
- **inhibitorAssistsPerMinCounts** (*ParticipantTimelineData*) – inhibitor assists per minute timeline counts
- **inhibitorKillsPerMinCounts** (*ParticipantTimelineData*) – inhibitor kills per minute timeline counts

- **lane** (*str*) – participant’s lane (Legal values: MID, MIDDLE, TOP, JUNGLE, BOT, BOTTOM)
- **role** (*str*) – participant’s role (Legal values: DUO, NONE, SOLO, DUO_CARRY, DUO_SUPPORT)
- **towerAssistsPerMinCounts** (*ParticipantTimelineData*) – tower assists per minute timeline counts
- **towerKillsPerMinCounts** (*ParticipantTimelineData*) – tower kills per minute timeline counts
- **towerKillsPerMinDeltas** (*ParticipantTimelineData*) – tower kills per minute timeline data
- **vilemawAssistsPerMinCounts** (*ParticipantTimelineData*) – vilemaw assists per minute timeline counts
- **vilemawKillsPerMinCounts** (*ParticipantTimelineData*) – vilemaw kills per minute timeline counts
- **wardsPerMinDeltas** (*ParticipantTimelineData*) – wards placed per minute timeline data
- **xpDiffPerMinDeltas** (*ParticipantTimelineData*) – experience difference per minute timeline data
- **xpPerMinDeltas** (*ParticipantTimelineData*) – experience per minute timeline data

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.match.Frame* (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Parameters

- **masteryId** (*int*) – mastery ID
- **rank** (*int*) – mastery rank

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.match.Mastery* (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Gets all summoner IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.match.MatchDetail* (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Parameters

- **mapId** (*int*) – match map ID
- **matchCreation** (*int*) – match creation time. Designates when the team select lobby is created and/or the match is made through match making, not when the game actually starts.

- **matchDuration** (*int*) – match duration
- **matchId** (*int*) – ID of the match
- **matchMode** (*str*) – match mode (Legal values: CLASSIC, ODIN, ARAM, TUTORIAL, ONEFORALL, ASCENSION, FIRSTBLOOD, KINGPORO)
- **matchType** (*str*) – match type (Legal values: CUSTOM_GAME, MATCHED_GAME, TUTORIAL_GAME)
- **matchVersion** (*str*) – match version
- **participantIdentities** (*list*<*ParticipantIdentity*>) – participant identity information
- **participants** (*list*<*Participant*>) – participant information
- **platformId** (*str*) – platform ID of the match
- **queueType** (*str*) – match queue type (Legal values: CUSTOM, NORMAL_5x5_BLIND, RANKED_SOLO_5x5, RANKED_PREMADE_5x5, BOT_5x5, NORMAL_3x3, RANKED_PREMADE_3x3, NORMAL_5x5_DRAFT, ODIN_5x5_BLIND, ODIN_5x5_DRAFT, BOT_ODIN_5x5, BOT_5x5_INTRO, BOT_5x5_BEGINNER, BOT_5x5_INTERMEDIATE, RANKED_TEAM_3x3, RANKED_TEAM_5x5, BOT_TT_3x3, GROUP_FINDER_5x5, ARAM_5x5, ONEFORALL_5x5, FIRSTBLOOD_1x1, FIRSTBLOOD_2x2, SR_6x6, URF_5x5, ONEFORALL_MIRRORMODE_5x5, BOT_URF_5x5, NIGHTMARE_BOT_5x5_RANK1, NIGHTMARE_BOT_5x5_RANK2, NIGHTMARE_BOT_5x5_RANK5, ASCENSION_5x5, HEXAKILL, KING_PORO_5x5, COUNTER_PICK)
- **region** (*str*) – region where the match was played
- **season** (*str*) – season match was played (Legal values: PRESEASON3, SEASON3, PRESEASON2014, SEASON2014, PRESEASON2015, SEASON2015)
- **teams** (*list*<*Team*>) – team information
- **timeline** (*Timeline*) – match timeline data (not included by default)

champion_ids

Gets all item IDs contained in this object

item_ids

Gets all item IDs contained in this object

mastery_ids

Gets all item IDs contained in this object

rune_ids

Gets all item IDs contained in this object

summoner_ids

Gets all item IDs contained in this object

summoner_spell_ids

Gets all item IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.Participant` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all item IDs contained in this object

`to_json (**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.ParticipantFrame` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **rank** (*int*) – rune rank
- **runeId** (*int*) – rune ID

`to_json (**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.ParticipantIdentity` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all champion IDs contained in this object

`to_json (**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.ParticipantStats` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all summoner spell IDs contained in this object

`to_json (**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.ParticipantTimeline` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **championId** (*int*) – champion ID
- **highestAchievedSeasonTier** (*str*) – highest ranked tier achieved for the previous season, if any, otherwise null. Used to display border in game loading screen. (Legal values: CHALLENGER, MASTER, DIAMOND, PLATINUM, GOLD, SILVER, BRONZE, UNRANKED)
- **masteries** (*list<Mastery>*) – list of mastery information
- **participantId** (*int*) – participant ID
- **runes** (*list<Rune>*) – list of rune information
- **spell1Id** (*int*) – first summoner spell ID
- **spell2Id** (*int*) – second summoner spell ID
- **stats** (`ParticipantStats`) – participant statistics
- **teamId** (*int*) – team ID
- **timeline** (`ParticipantTimeline`) – timeline data. Delta fields refer to values for the specified period (e.g., the gold per minute over the first 10 minutes of the game versus the second 20 minutes of the game. Diffs fields refer to the deltas versus the calculated lane opponent(s).

`to_json (**kwargs)`

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.ParticipantTimelineData` (*dictionary*, *type_=None*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **assists** (*int*) – number of assists
- **champLevel** (*int*) – champion level achieved
- **combatPlayerScore** (*int*) – if game was a dominion game, player’s combat score, otherwise 0
- **deaths** (*int*) – number of deaths
- **doubleKills** (*int*) – number of double kills
- **firstBloodAssist** (*bool*) – flag indicating if participant got an assist on first blood
- **firstBloodKill** (*bool*) – flag indicating if participant got first blood
- **firstInhibitorAssist** (*bool*) – flag indicating if participant got an assist on the first inhibitor
- **firstInhibitorKill** (*bool*) – flag indicating if participant destroyed the first inhibitor
- **firstTowerAssist** (*bool*) – flag indicating if participant got an assist on the first tower
- **firstTowerKill** (*bool*) – flag indicating if participant destroyed the first tower
- **goldEarned** (*int*) – gold earned
- **goldSpent** (*int*) – gold spent
- **inhibitorKills** (*int*) – number of inhibitor kills
- **item0** (*int*) – first item ID
- **item1** (*int*) – second item ID
- **item2** (*int*) – third item ID
- **item3** (*int*) – fourth item ID
- **item4** (*int*) – fifth item ID
- **item5** (*int*) – sixth item ID
- **item6** (*int*) – seventh item ID
- **killingSprees** (*int*) – number of killing sprees
- **kills** (*int*) – number of kills
- **largestCriticalStrike** (*int*) – largest critical strike
- **largestKillingSpree** (*int*) – largest killing spree
- **largestMultiKill** (*int*) – largest multi kill
- **magicDamageDealt** (*int*) – magical damage dealt
- **magicDamageDealtToChampions** (*int*) – magical damage dealt to champions
- **magicDamageTaken** (*int*) – magic damage taken
- **minionsKilled** (*int*) – minions killed

- **neutralMinionsKilled** (*int*) – neutral minions killed
- **neutralMinionsKilledEnemyJungle** (*int*) – neutral jungle minions killed in the enemy team’s jungle
- **neutralMinionsKilledTeamJungle** (*int*) – neutral jungle minions killed in your team’s jungle
- **nodeCapture** (*int*) – if game was a dominion game, number of node captures
- **nodeCaptureAssist** (*int*) – if game was a dominion game, number of node capture assists
- **nodeNeutralize** (*int*) – if game was a dominion game, number of node neutralizations
- **nodeNeutralizeAssist** (*int*) – if game was a dominion game, number of node neutralization assists
- **objectivePlayerScore** (*int*) – if game was a dominion game, player’s objectives score, otherwise 0
- **pentaKills** (*int*) – number of penta kills
- **physicalDamageDealt** (*int*) – physical damage dealt
- **physicalDamageDealtToChampions** (*int*) – physical damage dealt to champions
- **physicalDamageTaken** (*int*) – physical damage taken
- **quadraKills** (*int*) – number of quadra kills
- **sightWardsBoughtInGame** (*int*) – sight wards purchased
- **teamObjective** (*int*) – if game was a dominion game, number of completed team objectives (i.e., quests)
- **totalDamageDealt** (*int*) – total damage dealt
- **totalDamageDealtToChampions** (*int*) – total damage dealt to champions
- **totalDamageTaken** (*int*) – total damage taken
- **totalHeal** (*int*) – total heal amount
- **totalPlayerScore** (*int*) – if game was a dominion game, player’s total score, otherwise 0
- **totalScoreRank** (*int*) – if game was a dominion game, team rank of the player’s total score (e.g., 1-5)
- **totalTimeCrowdControlDealt** (*int*) – total dealt crowd control time
- **totalUnitsHealed** (*int*) – total units healed
- **towerKills** (*int*) – number of tower kills
- **tripleKills** (*int*) – number of triple kills
- **trueDamageDealt** (*int*) – true damage dealt
- **trueDamageDealtToChampions** (*int*) – true damage dealt to champions
- **trueDamageTaken** (*int*) – true damage taken
- **unrealKills** (*int*) – number of unreal kills

- **visionWardsBoughtInGame** (*int*) – vision wards purchased
- **wardsKilled** (*int*) – number of wards killed
- **wardsPlaced** (*int*) – number of wards placed
- **winner** (*bool*) – flag indicating whether or not the participant won

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.Player` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **bans** (*list*<*BannedChampion*>) – if game was draft mode, contains banned champion data, otherwise null
- **baronKills** (*int*) – number of times the team killed baron
- **dominionVictoryScore** (*int*) – if game was a dominion game, specifies the points the team had at game end, otherwise null
- **dragonKills** (*int*) – number of times the team killed dragon
- **firstBaron** (*bool*) – flag indicating whether or not the team got the first baron kill
- **firstBlood** (*bool*) – flag indicating whether or not the team got first blood
- **firstDragon** (*bool*) – flag indicating whether or not the team got the first dragon kill
- **firstInhibitor** (*bool*) – flag indicating whether or not the team destroyed the first inhibitor
- **firstRiftHerald** (*bool*) – flag indicating whether or not the team got the first rift herald kill
- **firstTower** (*bool*) – flag indicating whether or not the team destroyed the first tower
- **inhibitorKills** (*int*) – number of inhibitors the team destroyed
- **riftHeraldKills** (*int*) – number of times the team killed rift herald
- **teamId** (*int*) – team ID
- **towerKills** (*int*) – number of towers the team destroyed
- **vilemawKills** (*int*) – number of times the team killed vilemaw
- **winner** (*bool*) – flag indicating whether or not the team won

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.Position` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **matchHistoryUri** (*str*) – match history URI
- **profileIcon** (*int*) – profile icon ID
- **summonerId** (*int*) – summoner ID
- **summonerName** (*str*) – summoner name

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.Rune (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **participantId** (`int`) – participant ID
- **player** (`Player`) – player information

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.Team (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all mastery IDs contained in this object

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.match.Timeline (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all rune IDs contained in this object

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.matchlist.MatchList (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **endIndex** (`int`) – the last match index from history returned
- **matches** (`list<MatchReference>`) – list of matches from the player's history
- **startIndex** (`int`) – the first match index from history returned
- **totalGames** (`int`) – the number of games provided

champion_ids

Gets all champion IDs contained in this object

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.matchlist.MatchReference (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all champion IDs contained in this object

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.BasicData (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **colloq** (*str*) – colloq
- **consumeOnFull** (*bool*) – consume on full
- **consumed** (*bool*) – consumed
- **depth** (*int*) – depth
- **description** (*str*) – description
- **effect** (*dict*<*str*, *str*>) – effect
- **from** (*list*<*str*>) – from
- **gold** (*Gold*) – data dragon includes the gold field for basic data, which is shared by both rune and item. However, only items have a gold field on them, representing their gold cost in the store. Since runes are not sold in the store, they have no gold cost.
- **group** (*str*) – group
- **hideFromAll** (*bool*) – hide from all
- **id** (*int*) – ID
- **image** (*Image*) – image
- **inStore** (*bool*) – in store
- **into** (*list*<*str*>) – into
- **maps** (*dict*<*str*, *bool*>) – maps
- **name** (*str*) – name
- **plaintext** (*str*) – plain text
- **requiredChampion** (*str*) – required champion
- **rune** (*MetaData*) – rune
- **sanitizedDescription** (*str*) – sanitized description
- **specialRecipe** (*int*) – special recipe
- **stacks** (*int*) – stacks
- **stats** (*BasicDataStats*) – stats
- **tags** (*list*<*str*>) – tags

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.BasicDataStats` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **isRune** (*bool*) – is a rune
- **tier** (*str*) – tier
- **type** (*str*) – type

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.staticdata.Block (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **items** (*list<BlockItem>*) – the items
- **recMath** (*bool*) – rec math
- **type** (*str*) – type

```
to_json (**kwargs)
```

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.staticdata.BlockItem (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **count** (*int*) – item count
- **id** (*int*) – item ID

```
to_json (**kwargs)
```

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.staticdata.Champion (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **allytips** (*list<str>*) – ally tips
- **blurb** (*str*) – blurb
- **enemytips** (*list<str>*) – enemy tips
- **id** (*int*) – ID
- **image** (*Image*) – image
- **info** (*Info*) – info
- **key** (*str*) – key
- **lore** (*str*) – lore
- **name** (*str*) – name
- **partype** (*str*) – partype
- **passive** (*Passive*) – passive
- **recommended** (*list<Recommended>*) – recommended
- **skins** (*list<Skin>*) – skins
- **spells** (*list<ChampionSpell>*) – spells
- **stats** (*Stats*) – stats
- **tags** (*list<str>*) – tags
- **title** (*str*) – title

```
item_ids
```

Gets all item IDs contained in this object

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.ChampionList (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all item IDs contained in this object

item_ids

Parameters

- **data** (`dict<str, Champion>`) – champion data
- **format** (`str`) – format
- **keys** (`dict<str, str>`) – keys
- **type** (`str`) – type
- **version** (`str`) – version

`to_json (**kwargs)`

Parameters `dictionary (dict)` – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.ChampionSpell (dictionary)`

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **altimages** (`list<Image>`) – alternate images
- **cooldown** (`list<float>`) – cooldown
- **cooldownBurn** (`str`) – cooldown burn
- **cost** (`list<int>`) – cost
- **costBurn** (`str`) – cost burn
- **costType** (`str`) – cost type
- **description** (`str`) – description
- **effect** (`list<list<float>>`) – effects
- **effectBurn** (`list<str>`) – effect burn
- **image** (`Image`) – image
- **key** (`str`) – key
- **leveltip** (`LevelTip`) – level tip
- **maxrank** (`int`) – max rank
- **name** (`str`) – name
- **range** (`list<int> or "self"`) – range
- **rangeBurn** (`str`) – range burn
- **resource** (`str`) – resource
- **sanitizedDescription** (`str`) – sanitized description
- **sanitizedTooltip** (`str`) – sanitized tooltip
- **tooltip** (`str`) – tooltip

- **vars** (*list<SpellVars>*) – vars

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Gold` (*dictionary*)
Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all item IDs contained in this object

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Group` (*dictionary*)
Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **header** (*str*) – the header
- **tags** (*list[str]*) – tags

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Image` (*dictionary, is_alt=False*)
Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **full** (*str*) – full link
- **group** (*str*) – group
- **h** (*int*) – h
- **sprite** (*str*) – sprite
- **w** (*int*) – w
- **x** (*int*) – x
- **y** (*int*) – y

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Info` (*dictionary*)
Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **attack** (*int*) – attack rating
- **defense** (*int*) – defense rating
- **difficulty** (*int*) – difficulty rating
- **magic** (*int*) – magic rating

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Item` (*dictionary*)
Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **FlatArmorMod** (*float*) – the FlatArmorMod
- **FlatAttackSpeedMod** (*float*) – the FlatAttackSpeedMod
- **FlatBlockMod** (*float*) – the FlatBlockMod
- **FlatCritChanceMod** (*float*) – the FlatCritChanceMod
- **FlatCritDamageMod** (*float*) – the FlatCritDamageMod
- **FlatEXPBonus** (*float*) – the FlatEXPBonus
- **FlatEnergyPoolMod** (*float*) – the FlatEnergyPoolMod
- **FlatEnergyRegenMod** (*float*) – the FlatEnergyRegenMod
- **FlatHPPoolMod** (*float*) – the FlatHPPoolMod
- **FlatHPRegenMod** (*float*) – the FlatHPRegenMod
- **FlatMPPoolMod** (*float*) – the FlatMPPoolMod
- **FlatMPRegenMod** (*float*) – the FlatMPRegenMod
- **FlatMagicDamageMod** (*float*) – the FlatMagicDamageMod
- **FlatMovementSpeedMod** (*float*) – the FlatMovementSpeedMod
- **FlatPhysicalDamageMod** (*float*) – the FlatPhysicalDamageMod
- **FlatSpellBlockMod** (*float*) – the FlatSpellBlockMod
- **PercentArmorMod** (*float*) – the PercentArmorMod
- **PercentAttackSpeedMod** (*float*) – the PercentAttackSpeedMod
- **PercentBlockMod** (*float*) – the PercentBlockMod
- **PercentCritChanceMod** (*float*) – the PercentCritChanceMod
- **PercentCritDamageMod** (*float*) – the PercentCritDamageMod
- **PercentDodgeMod** (*float*) – the PercentDodgeMod
- **PercentEXPBonus** (*float*) – the PercentEXPBonus
- **PercentHPPoolMod** (*float*) – the PercentHPPoolMod
- **PercentHPRegenMod** (*float*) – the PercentHPRegenMod
- **PercentLifeStealMod** (*float*) – the PercentLifeStealMod
- **PercentMPPoolMod** (*float*) – the PercentMPPoolMod
- **PercentMPRegenMod** (*float*) – the PercentMPRegenMod
- **PercentMagicDamageMod** (*float*) – the PercentMagicDamageMod
- **PercentMovementSpeedMod** (*float*) – the PercentMovementSpeedMod
- **PercentPhysicalDamageMod** (*float*) – the PercentPhysicalDamageMod
- **PercentSpellBlockMod** (*float*) – the PercentSpellBlockMod
- **PercentSpellVampMod** (*float*) – the PercentSpellVampMod
- **rFlatArmorModPerLevel** (*float*) – the rFlatArmorModPerLevel
- **rFlatArmorPenetrationMod** (*float*) – the rFlatArmorPenetrationMod

- **rFlatArmorPenetrationModPerLevel** (*float*) – the rFlatArmorPenetrationModPerLevel
- **rFlatCritChanceModPerLevel** (*float*) – the rFlatCritChanceModPerLevel
- **rFlatCritDamageModPerLevel** (*float*) – the rFlatCritDamageModPerLevel
- **rFlatDodgeMod** (*float*) – the rFlatDodgeMod
- **rFlatDodgeModPerLevel** (*float*) – the rFlatDodgeModPerLevel
- **rFlatEnergyModPerLevel** (*float*) – the rFlatEnergyModPerLevel
- **rFlatEnergyRegenModPerLevel** (*float*) – the rFlatEnergyRegenModPerLevel
- **rFlatGoldPer10Mod** (*float*) – the rFlatGoldPer10Mod
- **rFlatHPModPerLevel** (*float*) – the rFlatHPModPerLevel
- **rFlatHPRegenModPerLevel** (*float*) – the rFlatHPRegenModPerLevel
- **rFlatMPModPerLevel** (*float*) – the rFlatMPModPerLevel
- **rFlatMPRegenModPerLevel** (*float*) – the rFlatMPRegenModPerLevel
- **rFlatMagicDamageModPerLevel** (*float*) – the rFlatMagicDamageModPerLevel
- **rFlatMagicPenetrationMod** (*float*) – the rFlatMagicPenetrationMod
- **rFlatMagicPenetrationModPerLevel** (*float*) – the rFlatMagicPenetrationModPerLevel
- **rFlatMovementSpeedModPerLevel** (*float*) – the rFlatMovementSpeedModPerLevel
- **rFlatPhysicalDamageModPerLevel** (*float*) – the rFlatPhysicalDamageModPerLevel
- **rFlatSpellBlockModPerLevel** (*float*) – the rFlatSpellBlockModPerLevel
- **rFlatTimeDeadMod** (*float*) – the rFlatTimeDeadMod
- **rFlatTimeDeadModPerLevel** (*float*) – the rFlatTimeDeadModPerLevel
- **rPercentArmorPenetrationMod** (*float*) – the rPercentArmorPenetrationMod
- **rPercentArmorPenetrationModPerLevel** (*float*) – the rPercentArmorPenetrationModPerLevel
- **rPercentAttackSpeedModPerLevel** (*float*) – the rPercentAttackSpeedModPerLevel
- **rPercentCooldownMod** (*float*) – the rPercentCooldownMod
- **rPercentCooldownModPerLevel** (*float*) – the rPercentCooldownModPerLevel
- **rPercentMagicPenetrationMod** (*float*) – the rPercentMagicPenetrationMod
- **rPercentMagicPenetrationModPerLevel** (*float*) – the rPercentMagicPenetrationModPerLevel
- **rPercentMovementSpeedModPerLevel** (*float*) – the rPercentMovementSpeedModPerLevel
- **rPercentTimeDeadMod** (*float*) – the rPercentTimeDeadMod
- **rPercentTimeDeadModPerLevel** (*float*) – the rPercentTimeDeadModPerLevel

item_ids

Parameters

- **header** (*str*) – the header
- **tags** (*list[str]*) – tags

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.ItemList` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all other item IDs contained in this object

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.ItemTree` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **base** (*int*) – base price
- **purchasable** (*bool*) – is purchasable
- **sell** (*int*) – sell price
- **total** (*int*) – total price

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.LanguageStrings` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **MaxGroupOwnable** (*str*) – max ownable of group
- **key** (*str*) – key

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.LevelTip` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **effect** (*list<str>*) – effects
- **label** (*list<str>*) – labels

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.MapData` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **basic** (`BasicData`) – basic data

- **data** (*dict*<*str*, *Item*>) –
- **groups** (*list*<*Group*>) – groups
- **tree** (*list*<*ItemTree*>) – item tree
- **type** (*str*) – type
- **version** (*str*) – version

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.MapDetails` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **colloq** (*str*) – colloq
- **consumeOnFull** (*bool*) – consume on full
- **consumed** (*bool*) – consumed
- **depth** (*int*) – depth
- **description** (*str*) – description
- **from** (*list*<*str*>) – from
- **gold** (*Gold*) – data dragon includes the gold field for basic data, which is shared by both rune and item. However, only items have a gold field on them, representing their gold cost in the store. Since runes are not sold in the store, they have no gold cost.
- **group** (*str*) – group
- **hideFromAll** (*bool*) – hide from all
- **id** (*int*) – ID
- **image** (*Image*) – image
- **inStore** (*bool*) – in store
- **into** (*list*<*str*>) – into
- **maps** (*dict*<*str*, *bool*>) – maps
- **name** (*str*) – name
- **plaintext** (*str*) – plain text
- **requiredChampion** (*str*) – required champion
- **rune** (*MetaData*) – rune
- **sanitizedDescription** (*str*) – sanitized description
- **specialRecipe** (*int*) – special recipe
- **stacks** (*int*) – stacks
- **stats** (*BasicDataStats*) – stats
- **tags** (*list* [*str*]) – tags

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

```

class cassiopeia.type.dto.staticdata.Mastery (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto

    Parameters

        • masteryId (int) – mastery ID

        • prereq (str) – prerequisites

    mastery_ids

        Parameters masteryTreeItems (list<MasteryTreeItem>) – mastery tree items

    to_json (**kwargs)

        Parameters dictionary (dict) – the JSON data returned from the Riot API as a dict

class cassiopeia.type.dto.staticdata.MasteryList (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto

    Parameters masteryTreeItems (list<MasteryTreeItem>) – mastery tree items

    to_json (**kwargs)

        Parameters dictionary (dict) – the JSON data returned from the Riot API as a dict

class cassiopeia.type.dto.staticdata.MasteryTree (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto

    Parameters

        • data (dict<str, MapDetails>) – map data

        • type (str) – type

        • version (str) – version

    to_json (**kwargs)

        Parameters dictionary (dict) – the JSON data returned from the Riot API as a dict

class cassiopeia.type.dto.staticdata.MasteryTreeItem (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto

    Parameters

        • data (dict<str, str>) – language str data

        • type (str) – type

        • version (str) – version

    to_json (**kwargs)

        Parameters dictionary (dict) – the JSON data returned from the Riot API as a dict

class cassiopeia.type.dto.staticdata.MasteryTreeList (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto

    Parameters

        • image (Image) – image

        • mapId (int) – ID

        • mapName (str) – name

        • unpurchasableItemList (list<int>) – items that can't be purchased on this
            map (IDs)

```

`to_json(**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.MetaData` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **data** (*dict*<*str*, *Champion*>) – champion data
- **format** (*str*) – format
- **keys** (*dict*<*str*, *str*>) – keys
- **type** (*str*) – type
- **version** (*str*) – version

`to_json(**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Passive` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **description** (*str*) – description
- **image** (*Image*) – image
- **name** (*str*) – name
- **sanitizedDescription** (*str*) – sanitized description

`to_json(**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Realm` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **Defense** (*list*<*MasteryTreeList*>) – defense tree
- **Offense** (*list*<*MasteryTreeList*>) – offense tree
- **Utility** (*list*<*MasteryTreeList*>) – utility tree

`to_json(**kwargs)`

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Recommended` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **blocks** (*list*<*Block*>) – blocks
- **champion** (*str*) – champion
- **map** (*str*) – map
- **mode** (*str*) – mode
- **priority** (*bool*) – priority

- **title** (*str*) – title
- **type** (*str*) – type

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Rune` (*dictionary*)
 Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **description** (*list<str>*) – description
- **id** (*int*) – iD
- **image** (*Image*) – image
- **masteryTree** (*str*) – legal values: Defense, Offense, Utility
- **name** (*str*) – name
- **prereq** (*str*) – prerequisites
- **ranks** (*int*) – ranks
- **sanitizedDescription** (*list<str>*) – sanitized description

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.RuneList` (*dictionary*)
 Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all other mastery IDs contained in this object

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Skin` (*dictionary*)
 Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **id** (*int*) – ID
- **name** (*str*) – name
- **num** (*int*) – number

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.SpellVars` (*dictionary*)
 Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **coeff** (*list<float>*) – coefficients
- **dyn** (*str*) – dyn
- **key** (*str*) – key
- **link** (*str*) – link

- **ranksWith** (*str*) – ranks with

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.Stats` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **armor** (*float*) – armor
- **armorperlevel** (*float*) – armor per level
- **attackdamage** (*float*) – attack damage
- **attackdamageperlevel** (*float*) – attack damage per level
- **attackrange** (*float*) – attack range
- **attackspeedoffset** (*float*) – attack speed offset
- **attackspeedperlevel** (*float*) – attack speed per level
- **crit** (*float*) – crit chance
- **critperlevel** (*float*) – crit change per level
- **hp** (*float*) – health
- **hpperlevel** (*float*) – health per level
- **hpregen** (*float*) – health regen
- **hpregenperlevel** (*float*) – health regen per level
- **movespeed** (*float*) – movespeed
- **mp** (*float*) – mana
- **mpperlevel** (*float*) – mana per level
- **mpregen** (*float*) – mana regen
- **mpregenperlevel** (*float*) – mana regen per level
- **spellblock** (*float*) – magic resist
- **spellblockperlevel** (*float*) – magic resist per level

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.SummonerSpell` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **data** (*dict*<*str*, *Mastery*>) – mastery data
- **tree** (*MasteryTree*) – mastery tree
- **type** (*str*) – type
- **version** (*str*) – version

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.staticdata.SummonerSpellList` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **cdn** (*str*) – the base CDN url
- **css** (*str*) – latest changed version of Dragon Magic’s css file
- **dd** (*str*) – latest changed version of Dragon Magic
- **l** (*str*) – default language for this realm
- **lg** (*str*) – legacy script mode for IE6 or older
- **n** (*dict*<*str*, *str*>) – latest changed version for each data type listed
- **profileiconmax** (*int*) – special behavior number identifying the largest profileicon id that can be used under 500.0 Any profileicon that is requested between this number and 500 should be mapped to 0.0
- **store** (*str*) – additional api data drawn from other sources that may be related to data dragon functionality
- **v** (*str*) – current version of this file for this realm

to_json (***kwargs*)

Parameters **dictionary** (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.stats.AggregatedStats` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **averageAssists** (*int*) – dominion only
- **averageChampionsKilled** (*int*) – dominion only
- **averageCombatPlayerScore** (*int*) – dominion only
- **averageNodeCapture** (*int*) – dominion only
- **averageNodeCaptureAssist** (*int*) – dominion only
- **averageNodeNeutralize** (*int*) – dominion only
- **averageNodeNeutralizeAssist** (*int*) – dominion only
- **averageNumDeaths** (*int*) – dominion only
- **averageObjectivePlayerScore** (*int*) – dominion only
- **averageTeamObjective** (*int*) – dominion only
- **averageTotalPlayerScore** (*int*) – dominion only
- **botGamesPlayed** (*int*) – botGamesPlayed
- **killingSpree** (*int*) – killingSpree
- **maxAssists** (*int*) – dominion only
- **maxChampionsKilled** (*int*) – maxChampionsKilled
- **maxCombatPlayerScore** (*int*) – dominion only
- **maxLargestCriticalStrike** (*int*) – maxLargestCriticalStrike
- **maxLargestKillingSpree** (*int*) – maxLargestKillingSpree

- **maxNodeCapture** (*int*) – dominion only
- **maxNodeCaptureAssist** (*int*) – dominion only
- **maxNodeNeutralize** (*int*) – dominion only
- **maxNodeNeutralizeAssist** (*int*) – dominion only
- **maxNumDeaths** (*int*) – only returned for ranked statistics.
- **maxObjectivePlayerScore** (*int*) – dominion only
- **maxTeamObjective** (*int*) – dominion only
- **maxTimePlayed** (*int*) – maxTimePlayed
- **maxTimeSpentLiving** (*int*) – maxTimeSpentLiving
- **maxTotalPlayerScore** (*int*) – dominion only
- **mostChampionKillsPerSession** (*int*) – mostChampionKillsPerSession
- **mostSpellsCast** (*int*) – mostSpellsCast
- **normalGamesPlayed** (*int*) – normalGamesPlayed
- **rankedPremadeGamesPlayed** (*int*) – rankedPremadeGamesPlayed
- **rankedSoloGamesPlayed** (*int*) – rankedSoloGamesPlayed
- **totalAssists** (*int*) – totalAssists
- **totalChampionKills** (*int*) – totalChampionKills
- **totalDamageDealt** (*int*) – totalDamageDealt
- **totalDamageTaken** (*int*) – totalDamageTaken
- **totalDeathsPerSession** (*int*) – only returned for ranked statistics
- **totalDoubleKills** (*int*) – totalDoubleKills
- **totalFirstBlood** (*int*) – totalFirstBlood
- **totalGoldEarned** (*int*) – totalGoldEarned
- **totalHeal** (*int*) – totalHeal
- **totalMagicDamageDealt** (*int*) – totalMagicDamageDealt
- **totalMinionKills** (*int*) – totalMinionKills
- **totalNeutralMinionsKilled** (*int*) – totalNeutralMinionsKilled
- **totalNodeCapture** (*int*) – dominion only
- **totalNodeNeutralize** (*int*) – dominion only
- **totalPentaKills** (*int*) – totalPentaKills
- **totalPhysicalDamageDealt** (*int*) – totalPhysicalDamageDealt
- **totalQuadraKills** (*int*) – totalQuadraKills
- **totalSessionsLost** (*int*) – totalSessionsLost
- **totalSessionsPlayed** (*int*) – totalSessionsPlayed
- **totalSessionsWon** (*int*) – totalSessionsWon
- **totalTripleKills** (*int*) – totalTripleKills

- **totalTurretsKilled** (*int*) – totalTurretsKilled
- **totalUnrealKills** (*int*) – totalUnrealKills

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.stats.ChampionStats` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Gets all champion IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.stats.PlayerStatsSummary` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **losses** (*int*) – number of losses for this queue type. Returned for ranked queue types only
- **modifyDate** (*int*) – date stats were last modified specified as epoch milliseconds
- **playerStatSummaryType** (*str*) – player stats summary type (Legal values: AramUnranked5x5, Ascension, CAP5x5, CoopVsAI, CoopVsAI3x3, CounterPick, FirstBlood1x1, FirstBlood2x2, Hexakill, KingPoro, NightmareBot, OdinUnranked, OneForAll5x5, RankedPremade3x3, RankedPremade5x5, RankedSolo5x5, RankedTeam3x3, RankedTeam5x5, SummonersRift6x6, Unranked, Unranked3x3, URF, URFBots)
- **wins** (*int*) – number of wins for this queue type

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.stats.PlayerStatsSummaryList` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **playerStatSummaries** (*list*<*PlayerStatsSummary*>) – collection of player stats summaries associated with the summoner
- **summonerId** (*int*) – summoner ID

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.stats.RankedStats` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **champions** (*list*<*ChampionStats*>) – collection of aggregated stats summarized by champion
- **modifyDate** (*int*) – date stats were last modified specified as epoch milliseconds
- **summonerId** (*int*) – summoner ID

champion_ids

Gets all champion IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.status.Incident` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **active** (*bool*) – active
- **created_at** (*str*) – timestamp created
- **id** (*int*) – ID
- **updates** (*list*<*Message*>) – updates

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.status.Message` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **author** (*str*) – author
- **content** (*str*) – content
- **created_at** (*str*) – timestamp created
- **id** (*int*) – ID
- **severity** (*str*) – legal values: Info, Alert, Error
- **translations** (*list*<*Translation*>) – translations
- **updated_at** (*str*) – timestamp updated

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.status.Service` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **incidents** (*list*<*Incident*>) – incidents
- **name** (*str*) – name
- **slug** (*str*) – slug
- **status** (*str*) – legal values: Online, Alert, Offline, Deploying

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.status.Shard` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **hostname** (*str*) – hostname

- **locales** (*list<str>*) – locales
- **name** (*str*) – name
- **region_tag** (*str*) – region tag
- **slug** (*str*) – slug

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.status.ShardStatus` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **hostname** (*str*) – hostname
- **locales** (*list<string>*) – locales
- **name** (*str*) – name
- **region_tag** (*str*) – region tag
- **services** (*list<Service>*) – services
- **slug** (*str*) – slug

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.status.Translation` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **content** (*str*) – content
- **locale** (*str*) – locale
- **updated_at** (*str*) – timestamp

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.summoner.Mastery` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **pages** (*list<MasteryPage>*) – collection of mastery pages associated with the summoner
- **summonerId** (*int*) – summoner ID

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.summoner.MasteryPage` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **runeId** (*int*) – rune ID associated with the rune slot. For static information correlating to rune IDs, please refer to the LoL Static Data API.

- **runeSlotId** (*int*) – rune slot ID.

mastery_ids

Parameters

- **pages** (*list*<*MasteryPage*>) – collection of mastery pages associated with the summoner
- **summonerId** (*int*) – summoner ID

to_json (***kwargs*)

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.summoner.MasteryPages* (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Gets all rune IDs contained in this object

mastery_ids

Parameters

- **runeId** (*int*) – rune ID associated with the rune slot. For static information correlating to rune IDs, please refer to the LoL Static Data API.
- **runeSlotId** (*int*) – rune slot ID.

to_json (***kwargs*)

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.summoner.RunePage* (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Gets all rune IDs contained in this object

rune_ids

Parameters

- **current** (*bool*) – indicates if the page is the current page
- **id** (*int*) – rune page ID
- **name** (*str*) – rune page name
- **slots** (*list*<*RuneSlot*>) – collection of rune slots associated with the rune page

to_json (***kwargs*)

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.summoner.RunePages* (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Parameters

- **pages** (*list*<*RunePage*>) – collection of rune pages associated with the summoner
- **summonerId** (*int*) – summoner ID

rune_ids

Gets all rune IDs contained in this object

to_json (***kwargs*)

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.summoner.RuneSlot (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **current** (*bool*) – indicates if the page is the current page
- **id** (*int*) – rune page ID
- **name** (*str*) – rune page name
- **slots** (*list<RuneSlot>*) – collection of rune slots associated with the rune page

```
to_json (**kwargs)
```

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.summoner.Summoner (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Gets all mastery IDs contained in this object

```
to_json (**kwargs)
```

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.team.MatchHistorySummary (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Gets all summoner IDs contained in this object

```
to_json (**kwargs)
```

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.team.Roster (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **assists** (*int*) – assists
- **date** (*int*) – date that match was completed specified as epoch milliseconds
- **deaths** (*int*) – deaths
- **gameId** (*int*) – gameId
- **gameMode** (*str*) – gameMode
- **invalid** (*bool*) – invalid
- **kills** (*int*) – kills
- **mapId** (*int*) – mapId
- **opposingTeamKills** (*int*) – opposingTeamKills
- **opposingTeamName** (*str*) – opposingTeamName
- **win** (*bool*) – win

```
to_json (**kwargs)
```

Parameters dictionary (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.team.Team (dictionary)
    Bases: cassiopeia.type.dto.common.CassiopeiaDto
```

Parameters

- **createDate** (*int*) – date that team was created specified as epoch milliseconds
- **fullId** (*str*) – fullId
- **lastGameDate** (*int*) – date that last game played by team ended specified as epoch milliseconds
- **lastJoinDate** (*int*) – date that last member joined specified as epoch milliseconds
- **lastJoinedRankedTeamQueueDate** (*int*) – date that team last joined the ranked team queue specified as epoch milliseconds
- **matchHistory** (*list<MatchHistorySummary>*) – matchHistory
- **modifyDate** (*int*) – date that team was last modified specified as epoch milliseconds
- **name** (*str*) – name
- **roster** (*Roster*) – roster
- **secondLastJoinDate** (*int*) – date that second to last member joined specified as epoch milliseconds
- **status** (*str*) – status
- **tag** (*str*) – tag
- **teamStatDetails** (*list<TeamStatDetail>*) – stat details
- **thirdLastJoinDate** (*int*) – date that third to last member joined specified as epoch milliseconds

summoner_ids

Gets all summoner IDs contained in this object

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.team*.**TeamMemberInfo** (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Parameters

- **averageGamesPlayed** (*int*) – averageGamesPlayed
- **losses** (*int*) – losses
- **teamStatType** (*str*) – teamStatType
- **wins** (*int*) – wins

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class *cassiopeia.type.dto.team*.**TeamStatDetail** (*dictionary*)

Bases: *cassiopeia.type.dto.common.CassiopeiaDto*

Parameters

- **memberList** (*list<TeamMemberInfo>*) – memberList
- **ownerId** (*int*) – ownerId

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.tournament.LobbyEvent` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **eventType** (*str*) – the type of event that was triggered
- **summonerId** (*str*) – the summoner that triggered the event
- **timestamp** (*str*) – timestamp from the event

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.tournament.LobbyEventWrapper` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters *eventList* (*list<LobbyEvent>*) – the list of events

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

class `cassiopeia.type.dto.tournament.ProviderRegistrationParameters` (*region, url*)

Bases: `cassiopeia.type.dto.common.CassiopeiaParametersDto`

Parameters

- **region** (*str*) – the region in which the provider will be running tournaments (Legal values: BR, EUNE, EUW, JP, KR, LAN, LAS, NA, OCE, PBE, RU, TR)
- **url** (*str*) – the provider’s callback URL to which tournament game results in this region should be posted. The URL must be well-formed, use the http or https protocol, and use the default port for the protocol (http URLs must use port 80, https URLs must use port 443).

to_json (***kwargs*)

Gets a JSON representation of the object

Returns a JSON representation of the object

Return type *str*

class `cassiopeia.type.dto.tournament.SummonerIdParams` (*participants*)

Bases: `cassiopeia.type.dto.common.CassiopeiaParametersDto`

Returns *list<int>* the tournament participants

Return type *participants*

to_json (***kwargs*)

Gets a JSON representation of the object

Returns a JSON representation of the object

Return type *str*

class `cassiopeia.type.dto.tournament.TournamentCode` (*dictionary*)

Bases: `cassiopeia.type.dto.common.CassiopeiaDto`

Parameters

- **code** (*str*) – the tournament code
- **id** (*int*) – the tournament code’s ID
- **lobbyName** (*str*) – the lobby name for the tournament code game

- **map** (*str*) – the game map for the tournament code game
- **metaData** (*str*) – the metadata for tournament code
- **participants** (*list<int>*) – the IDs of the summoners participating in the tournament
- **password** (*str*) – the password for the tournament code game
- **pickType** (*str*) – the pick mode for tournament code game
- **providerId** (*int*) – the provider’s ID
- **region** (*str*) – the tournament code’s region (Legal values: BR, EUNE, EUW, JP, KR, LAN, LAS, NA, OCE, PBE, RU, TR)
- **spectators** (*str*) – the spectator mode for the tournament code game
- **teamSize** (*int*) – the team size for the tournament code game
- **tournamentId** (*int*) – the tournament’s ID

to_json (***kwargs*)

Parameters *dictionary* (*dict*) – the JSON data returned from the Riot API as a dict

```
class cassiopeia.type.dto.tournament.TournamentCodeParameters (teamSize,    specta-
                                                                torType,    pickType,
                                                                mapType,    allowed-
                                                                SummonerIds=None,
                                                                metadata='')
```

Bases: *cassiopeia.type.dto.common.CassiopeiaParametersDto*

Parameters

- **teamSize** (*int*) – the team size of the game. Valid values are 1-5.
- **spectatorType** (*str*) – the spectator type of the game. Valid values are NONE, LOBBYONLY, ALL.
- **pickType** (*str*) – the pick type of the game. Valid values are BLIND_PICK, DRAFT_MODE, ALL_RANDOM, TOURNAMENT_DRAFT.
- **mapType** (*str*) – the map type of the game. Valid values are SUMMONERS_RIFT, TWISTED_TREELINE, CRYSTAL_SCAR, and HOWLING_ABYSS.
- **allowedSummonerIds** (*SummonerIdParams*) – optional list of participants in order to validate the players eligible to join the lobby. NOTE: We currently do not enforce participants at the team level, but rather the aggregate of teamOne and teamTwo. We may add the ability to enforce at the team level in the future.
- **metadata** (*str*) – optional string that may contain any data in any format, if specified at all. Used to denote any custom information about the game.

to_json (***kwargs*)

Gets a JSON representation of the object

Returns a JSON representation of the object

Return type *str*


```
class cassiopeia.type.dto.tournament.TournamentCodeUpdateParameters (allowedParticipants='',
                                                                    specta-
                                                                    torType='',
                                                                    pick-
                                                                    Type='',
                                                                    map-
                                                                    Type='')
```

Bases: `cassiopeia.type.dto.common.CassiopeiaParametersDto`

Parameters

- **allowedParticipants** (*str*) – comma separated list of summoner Ids
- **spectatorType** (*str*) – the spectator type (Legal values: NONE, LOBBYONLY, ALL)
- **pickType** (*str*) – the pick type (Legal values: BLIND_PICK, DRAFT_MODE, ALL_RANDOM, TOURNAMENT_DRAFT)
- **mapType** (*str*) – the map type (Legal values: SUMMONERS_RIFT, CRYSTAL_SCAR, TWISTED_TREELINE, HOWLING_ABYSS)

to_json (**kwargs)

Gets a JSON representation of the object

Returns a JSON representation of the object

Return type *str*

```
class cassiopeia.type.dto.tournament.TournamentRegistrationParameters (providerId,
                                                                    name='')
```

Bases: `cassiopeia.type.dto.common.CassiopeiaParametersDto`

Parameters

- **providerId** (*int*) – the provider ID to specify the regional registered provider data to associate this tournament
- **name** (*str*) – the optional name of the tournament

to_json (**kwargs)

Gets a JSON representation of the object

Returns a JSON representation of the object

Return type *str*

Index and Search

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

C

`cassiopeia.baseriotapi`, 19
`cassiopeia.core.championapi`, 29
`cassiopeia.core.championmasteryapi`, 29
`cassiopeia.core.currentgameapi`, 30
`cassiopeia.core.featuredgamesapi`, 30
`cassiopeia.core.gameapi`, 30
`cassiopeia.core.leagueapi`, 30
`cassiopeia.core.matchapi`, 31
`cassiopeia.core.matchlistapi`, 32
`cassiopeia.core.requests`, 32
`cassiopeia.core.staticdataapi`, 32
`cassiopeia.core.statsapi`, 35
`cassiopeia.core.statusapi`, 35
`cassiopeia.core.summonerapi`, 35
`cassiopeia.core.teamapi`, 36
`cassiopeia.core.tournamentapi`, 37
`cassiopeia.dto.championapi`, 38
`cassiopeia.dto.championmasteryapi`, 38
`cassiopeia.dto.currentgameapi`, 39
`cassiopeia.dto.featuredgamesapi`, 39
`cassiopeia.dto.gameapi`, 39
`cassiopeia.dto.leagueapi`, 39
`cassiopeia.dto.matchapi`, 40
`cassiopeia.dto.matchlistapi`, 41
`cassiopeia.dto.requests`, 41
`cassiopeia.dto.staticdataapi`, 42
`cassiopeia.dto.statsapi`, 44
`cassiopeia.dto.statusapi`, 44
`cassiopeia.dto.summonerapi`, 44
`cassiopeia.dto.teamapi`, 45
`cassiopeia.dto.tournamentapi`, 45
`cassiopeia.riotapi`, 9
`cassiopeia.type.api.exception`, 46
`cassiopeia.type.api.rates`, 47
`cassiopeia.type.api.store`, 48
`cassiopeia.type.core.champion`, 51
`cassiopeia.type.core.championmastery`, 52
`cassiopeia.type.core.common`, 53
`cassiopeia.type.core.currentgame`, 60
`cassiopeia.type.core.featuredgames`, 63
`cassiopeia.type.core.game`, 65
`cassiopeia.type.core.league`, 74
`cassiopeia.type.core.match`, 76
`cassiopeia.type.core.matchlist`, 94
`cassiopeia.type.core.staticdata`, 95
`cassiopeia.type.core.stats`, 120
`cassiopeia.type.core.status`, 125
`cassiopeia.type.core.summoner`, 128
`cassiopeia.type.core.team`, 132
`cassiopeia.type.core.tournament`, 135
`cassiopeia.type.dto.champion`, 138
`cassiopeia.type.dto.championmastery`, 138
`cassiopeia.type.dto.common`, 139
`cassiopeia.type.dto.currentgame`, 150
`cassiopeia.type.dto.featuredgames`, 152
`cassiopeia.type.dto.game`, 153
`cassiopeia.type.dto.league`, 157
`cassiopeia.type.dto.match`, 158
`cassiopeia.type.dto.matchlist`, 166
`cassiopeia.type.dto.staticdata`, 166
`cassiopeia.type.dto.stats`, 179
`cassiopeia.type.dto.status`, 182
`cassiopeia.type.dto.summoner`, 183
`cassiopeia.type.dto.team`, 185
`cassiopeia.type.dto.tournament`, 186

Symbols

- `__add__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 139
 - `__and__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__div__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__eq__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__ge__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__getattr__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 146
 - `__getitem__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__gt__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__invert__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__le__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__lshift__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 140
 - `__lt__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__mod__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__mul__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__ne__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__neg__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__or__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__radd__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__rdiv__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__rmod__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__rmul__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 141
 - `__rshift__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142
 - `__rsub__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142
 - `__rtruediv__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142
 - `__sub__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142
 - `__truediv__()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142
- ## A
- `ability_power` (cassiopeia.type.core.staticdata.ItemStats attribute), 103
 - `ability_power_per_level` (cassiopeia.type.core.staticdata.ItemStats attribute), 103
 - `active` (cassiopeia.type.core.status.Incident attribute), 125
 - `adapt()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 146
 - `AggregatedStats` (class in cassiopeia.type.core.stats), 120
 - `AggregatedStats` (class in cassiopeia.type.dto.stats), 179
 - `all` (cassiopeia.type.core.tournament.SpectatorType attribute), 136
 - `all_()` (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142
 - `ally_monster_kills` (cassiopeia.type.core.game.Stats attribute), 67
 - `ally_monster_kills` (cassiopeia.type.core.match.ParticipantStats attribute), 83
 - `ally_tips` (cassiopeia.type.core.staticdata.Champion attribute), 95
 - `alternate_images` (cassiopeia.type.core.staticdata.Spell attribute), 114
 - `ancient_golem_assists_per_min_counts` (cassiopeia.type.core.match.ParticipantTimeline attribute), 114

attribute), 88

ancient_golem_kills_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

any_() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142

APIError, 46

aram (cassiopeia.type.core.common.GameMode attribute), 54

aram (cassiopeia.type.core.common.Queue attribute), 56

aram (cassiopeia.type.core.common.StatSummaryType attribute), 58

aram (cassiopeia.type.core.common.SubType attribute), 59

args (cassiopeia.type.api.exception.APIError attribute), 46

args (cassiopeia.type.api.exception.CassiopeiaException attribute), 46

armor (cassiopeia.type.core.staticdata.ChampionStats attribute), 97

armor (cassiopeia.type.core.staticdata.ItemStats attribute), 103

armor_penetration (cassiopeia.type.core.staticdata.ItemStats attribute), 103

armor_penetration_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 103

armor_per_level (cassiopeia.type.core.staticdata.ChampionStats attribute), 97

armor_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 103

asc() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142

ascended (cassiopeia.type.core.match.Event attribute), 77

Ascended (class in cassiopeia.type.core.common), 53

ascension (cassiopeia.type.core.common.EventType attribute), 53

ascension (cassiopeia.type.core.common.GameMode attribute), 54

ascension (cassiopeia.type.core.common.Queue attribute), 56

ascension (cassiopeia.type.core.common.StatSummaryType attribute), 58

ascension (cassiopeia.type.core.common.SubType attribute), 59

assisted_lane_deaths_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

assisted_lane_kills_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

assists (cassiopeia.type.core.game.Stats attribute), 67

assists (cassiopeia.type.core.match.Event attribute), 77

assists (cassiopeia.type.core.match.ParticipantStats attribute), 83

assists (cassiopeia.type.core.stats.AggregatedStats attribute), 120

assists (cassiopeia.type.core.team.MatchSummary attribute), 132

attack_damage (cassiopeia.type.core.staticdata.ChampionStats attribute), 97

attack_damage (cassiopeia.type.core.staticdata.ItemStats attribute), 103

attack_damage_per_level (cassiopeia.type.core.staticdata.ChampionStats attribute), 97

attack_damage_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 103

attack_range (cassiopeia.type.core.staticdata.ChampionStats attribute), 97

attack_speed (cassiopeia.type.core.staticdata.ChampionStats attribute), 97

attack_speed (cassiopeia.type.core.staticdata.ItemStats attribute), 103

author (cassiopeia.type.core.status.Message attribute), 126

average_assists (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_combat_score (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_deaths (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_games_played (cassiopeia.type.core.team.Stats attribute), 133

average_kills (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_node_capture_assists (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_node_captures (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_node_neutralization_assists (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_node_neutralizations (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_objective_score (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_score (cassiopeia.type.core.stats.AggregatedStats attribute), 120

average_team_score (cassiopeia.type.core.stats.AggregatedStats attribute), 120

tribute), 121

B

Ban (class in cassiopeia.type.core.currentgame), 60

Ban (class in cassiopeia.type.core.featuredgames), 63

Ban (class in cassiopeia.type.core.match), 76

BannedChampion (class in cassiopeia.type.dto.currentgame), 150

BannedChampion (class in cassiopeia.type.dto.featuredgames), 152

BannedChampion (class in cassiopeia.type.dto.match), 158

bans (cassiopeia.type.core.currentgame.Game attribute), 61

bans (cassiopeia.type.core.featuredgames.Game attribute), 63

bans (cassiopeia.type.core.match.Team attribute), 92

baron (cassiopeia.type.core.common.Monster attribute), 55

baron_assists_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

baron_kills (cassiopeia.type.core.match.Team attribute), 92

baron_kills_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

base (cassiopeia.type.core.staticdata.Gold attribute), 99

BaseDB (in module cassiopeia.type.dto.common), 139

BasicData (class in cassiopeia.type.dto.staticdata), 166

BasicDataStats (class in cassiopeia.type.dto.staticdata), 167

between() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142

bind_expression() (cassiopeia.type.dto.common.JSONEncoded method), 146

bind_processor() (cassiopeia.type.dto.common.JSONEncoded method), 146

black_market (cassiopeia.type.core.common.Queue attribute), 56

black_market (cassiopeia.type.core.common.StatSummaryType attribute), 58

black_market (cassiopeia.type.core.common.SubType attribute), 59

blind (cassiopeia.type.core.tournament.PickType attribute), 136

block (cassiopeia.type.core.staticdata.ItemStats attribute), 103

Block (class in cassiopeia.type.dto.staticdata), 167

BlockItem (class in cassiopeia.type.dto.staticdata), 168

blue (cassiopeia.type.core.common.Monster attribute), 55

blue (cassiopeia.type.core.common.Side attribute), 58

blue_team (cassiopeia.type.core.match.Match attribute), 79

blue_trinket (cassiopeia.type.core.common.Ward attribute), 60

blurb (cassiopeia.type.core.staticdata.Champion attribute), 95

blurb (cassiopeia.type.core.staticdata.Item attribute), 100

boneyard (cassiopeia.type.core.common.Point attribute), 55

BOOLEANTYPE (cassiopeia.type.dto.common.JSONEncoded.Comparator attribute), 139

bot (cassiopeia.type.core.currentgame.Participant attribute), 62

bot (cassiopeia.type.core.featuredgames.Participant attribute), 64

bot_beginner_fives (cassiopeia.type.core.common.Queue attribute), 56

bot_dominion (cassiopeia.type.core.common.Queue attribute), 56

bot_fives (cassiopeia.type.core.common.Queue attribute), 56

bot_fives (cassiopeia.type.core.common.StatSummaryType attribute), 58

bot_fives (cassiopeia.type.core.common.SubType attribute), 59

bot_games (cassiopeia.type.core.stats.AggregatedStats attribute), 121

bot_intermediate_fives (cassiopeia.type.core.common.Queue attribute), 56

bot_intro_fives (cassiopeia.type.core.common.Queue attribute), 56

bot_lane (cassiopeia.type.core.common.Lane attribute), 54

bot_lane (cassiopeia.type.core.common.LaneType attribute), 54

bot_threes (cassiopeia.type.core.common.Queue attribute), 56

bot_threes (cassiopeia.type.core.common.StatSummaryType attribute), 58

bot_threes (cassiopeia.type.core.common.SubType attribute), 59

bot_urf (cassiopeia.type.core.common.Queue attribute), 56

bot_urf (cassiopeia.type.core.common.StatSummaryType attribute), 58

bot_urf (cassiopeia.type.core.common.SubType attribute), 59

brazil (cassiopeia.type.core.common.Platform attribute), 55

brazil (cassiopeia.type.core.common.Region attribute), 57

brazil (cassiopeia.type.core.tournament.TournamentRegion attribute), 137
bronze (cassiopeia.type.core.common.Tier attribute), 59
building (cassiopeia.type.core.match.Event attribute), 77
Building (class in cassiopeia.type.core.common), 53
building_kill (cassiopeia.type.core.common.EventType attribute), 53
butchers_bridge (cassiopeia.type.core.common.Map attribute), 55
butchers_bridge (cassiopeia.type.core.common.Queue attribute), 56

C

Cache (class in cassiopeia.type.api.store), 48
call() (cassiopeia.type.api.rates.MultiRateLimiter method), 47
call() (cassiopeia.type.api.rates.SingleRateLimiter method), 47
call_with_ensured_size() (in module cassiopeia.core.requests), 32
calls (cassiopeia.type.api.rates.MultiRateLimiter attribute), 47
calls (cassiopeia.type.api.rates.SingleRateLimiter attribute), 47
captain (cassiopeia.type.core.team.Team attribute), 133
carry (cassiopeia.type.core.common.Role attribute), 57
cassiopeia.baseriotapi (module), 19
cassiopeia.core.championapi (module), 29
cassiopeia.core.championmasteryapi (module), 29
cassiopeia.core.currentgameapi (module), 30
cassiopeia.core.featuredgamesapi (module), 30
cassiopeia.core.gameapi (module), 30
cassiopeia.core.leagueapi (module), 30
cassiopeia.core.matchapi (module), 31
cassiopeia.core.matchlistapi (module), 32
cassiopeia.core.requests (module), 32
cassiopeia.core.staticdataapi (module), 32
cassiopeia.core.statsapi (module), 35
cassiopeia.core.statusapi (module), 35
cassiopeia.core.summonerapi (module), 35
cassiopeia.core.teamapi (module), 36
cassiopeia.core.tournamentapi (module), 37
cassiopeia.dto.championapi (module), 38
cassiopeia.dto.championmasteryapi (module), 38
cassiopeia.dto.currentgameapi (module), 39
cassiopeia.dto.featuredgamesapi (module), 39
cassiopeia.dto.gameapi (module), 39
cassiopeia.dto.leagueapi (module), 39
cassiopeia.dto.matchapi (module), 40
cassiopeia.dto.matchlistapi (module), 41
cassiopeia.dto.requests (module), 41
cassiopeia.dto.staticdataapi (module), 42
cassiopeia.dto.statsapi (module), 44
cassiopeia.dto.statusapi (module), 44

cassiopeia.dto.summonerapi (module), 44
cassiopeia.dto.teamapi (module), 45
cassiopeia.dto.tournamentapi (module), 45
cassiopeia.riotapi (module), 9
cassiopeia.type.api.exception (module), 46
cassiopeia.type.api.rates (module), 47
cassiopeia.type.api.store (module), 48
cassiopeia.type.core.champion (module), 51
cassiopeia.type.core.championmastery (module), 52
cassiopeia.type.core.common (module), 53
cassiopeia.type.core.currentgame (module), 60
cassiopeia.type.core.featuredgames (module), 63
cassiopeia.type.core.game (module), 65
cassiopeia.type.core.league (module), 74
cassiopeia.type.core.match (module), 76
cassiopeia.type.core.matchlist (module), 94
cassiopeia.type.core.staticdata (module), 95
cassiopeia.type.core.stats (module), 120
cassiopeia.type.core.status (module), 125
cassiopeia.type.core.summoner (module), 128
cassiopeia.type.core.team (module), 132
cassiopeia.type.core.tournament (module), 135
cassiopeia.type.dto.champion (module), 138
cassiopeia.type.dto.championmastery (module), 138
cassiopeia.type.dto.common (module), 139
cassiopeia.type.dto.currentgame (module), 150
cassiopeia.type.dto.featuredgames (module), 152
cassiopeia.type.dto.game (module), 153
cassiopeia.type.dto.league (module), 157
cassiopeia.type.dto.match (module), 158
cassiopeia.type.dto.matchlist (module), 166
cassiopeia.type.dto.staticdata (module), 166
cassiopeia.type.dto.stats (module), 179
cassiopeia.type.dto.status (module), 182
cassiopeia.type.dto.summoner (module), 183
cassiopeia.type.dto.team (module), 185
cassiopeia.type.dto.tournament (module), 186
CassiopeiaDto (class in cassiopeia.type.dto.common), 139
CassiopeiaException, 46
CassiopeiaObject (class in cassiopeia.type.core.common), 53
CassiopeiaParametersDto (class in cassiopeia.type.dto.common), 139
categories (cassiopeia.type.core.staticdata.Item attribute), 100
cdn (cassiopeia.type.core.staticdata.Realm attribute), 111
challenger (cassiopeia.type.core.common.Tier attribute), 59
champion (cassiopeia.type.core.champion.ChampionStatus attribute), 51
champion (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52

- `champion` (`cassiopeia.type.core.currentgame.Ban` attribute), 60
- `champion` (`cassiopeia.type.core.currentgame.Participant` attribute), 62
- `champion` (`cassiopeia.type.core.featuredgames.Ban` attribute), 63
- `champion` (`cassiopeia.type.core.featuredgames.Participant` attribute), 64
- `champion` (`cassiopeia.type.core.game.Game` attribute), 65
- `champion` (`cassiopeia.type.core.game.Participant` attribute), 66
- `champion` (`cassiopeia.type.core.match.Ban` attribute), 76
- `champion` (`cassiopeia.type.core.match.Participant` attribute), 81
- `champion` (`cassiopeia.type.core.matchlist.MatchReference` attribute), 94
- `champion` (`cassiopeia.type.core.staticdata.RecommendedItems` attribute), 112
- `Champion` (class in `cassiopeia.type.core.staticdata`), 95
- `Champion` (class in `cassiopeia.type.dto.champion`), 138
- `Champion` (class in `cassiopeia.type.dto.staticdata`), 168
- `champion_ids` (`cassiopeia.type.dto.champion.ChampionList` code attribute), 138
- `champion_ids` (`cassiopeia.type.dto.currentgame.CurrentGameInfo` attribute), 151
- `champion_ids` (`cassiopeia.type.dto.featuredgames.FeaturedGames` attribute), 152
- `champion_ids` (`cassiopeia.type.dto.featuredgames.FeaturedGames` attribute), 153
- `champion_ids` (`cassiopeia.type.dto.game.Game` attribute), 154
- `champion_ids` (`cassiopeia.type.dto.game.RecentGames` attribute), 157
- `champion_ids` (`cassiopeia.type.dto.match.MatchDetail` attribute), 161
- `champion_ids` (`cassiopeia.type.dto.matchlist.MatchList` attribute), 166
- `champion_ids` (`cassiopeia.type.dto.stats.RankedStats` attribute), 181
- `champion_level` (`cassiopeia.type.core.match.ParticipantStats` attribute), 83
- `champion_mastersies()` (`cassiopeia.type.core.summoner.Summoner` method), 129
- `champion_mastery()` (`cassiopeia.type.core.summoner.Summoner` method), 130
- `champion_mastery_score()` (`cassiopeia.type.core.summoner.Summoner` method), 130
- `ChampionInfo` (class in `cassiopeia.type.core.staticdata`), 96
- `ChampionList` (class in `cassiopeia.type.dto.champion`), 138
- `ChampionList` (class in `cassiopeia.type.dto.staticdata`), 169
- `ChampionMastery` (class in `cassiopeia.type.core.championmastery`), 52
- `ChampionMastery` (class in `cassiopeia.type.dto.championmastery`), 138
- `ChampionSpell` (class in `cassiopeia.type.dto.staticdata`), 169
- `ChampionStats` (class in `cassiopeia.type.core.staticdata`), 97
- `ChampionStats` (class in `cassiopeia.type.dto.stats`), 181
- `ChampionStatus` (class in `cassiopeia.type.core.champion`), 51
- `chest_granted` (`cassiopeia.type.core.championmastery.ChampionMastery` attribute), 52
- `class_` (`cassiopeia.type.api.store.HasAllStatus` attribute), 49
- `classic` (`cassiopeia.type.core.common.GameMode` attribute), 54
- `close()` (`cassiopeia.type.api.store.SQLAlchemyDB` method), 49
- `code` (`cassiopeia.type.core.tournament.TournamentCode` attribute), 136
- `coefficients` (`cassiopeia.type.core.staticdata.SpellVariables` attribute), 117
- `compare_info_compared_value()` (`cassiopeia.type.dto.common.JSONEncoded` method), 146
- `coerce_to_is_types` (`cassiopeia.type.dto.common.JSONEncoded` attribute), 146
- `collate()` (`cassiopeia.type.dto.common.JSONEncoded.Comparator` method), 142
- `column_expression()` (`cassiopeia.type.dto.common.JSONEncoded` method), 147
- `combat_score` (`cassiopeia.type.core.game.Stats` attribute), 67
- `combat_score` (`cassiopeia.type.core.match.ParticipantStats` attribute), 83
- `CombinedParticipant` (class in `cassiopeia.type.core.match`), 77
- `comparator_factory` (`cassiopeia.type.dto.common.JSONEncoded` attribute), 147
- `compare_against_backend()` (`cassiopeia.type.dto.common.JSONEncoded` method), 147
- `compare_values()` (`cassiopeia.type.dto.common.JSONEncoded` method), 147
- `compile()` (`cassiopeia.type.dto.common.JSONEncoded` method), 147

component_of (cassiopeia.type.core.staticdata.Item attribute), 100	create_tournament() (in module cassiopeia.core.tournamentapi), 37	cas-
components (cassiopeia.type.core.staticdata.Item attribute), 100	create_tournament() (in module cassiopeia.dto.tournamentapi), 45	cas-
concat() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142	create_tournament() (in module cassiopeia.riotapi), 9	
consumable (cassiopeia.type.core.staticdata.Item attribute), 100	create_tournament_codes() (in module cassiopeia.baseriotapi), 19	cas-
consumables_bought (cassiopeia.type.core.game.Stats attribute), 67	create_tournament_codes() (in module cassiopeia.core.tournamentapi), 37	cas-
consume_on_full (cassiopeia.type.core.staticdata.Item attribute), 100	create_tournament_codes() (in module cassiopeia.dto.tournamentapi), 45	cas-
contains() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142	create_tournament_codes() (in module cassiopeia.riotapi), 9	cas-
content (cassiopeia.type.core.status.Message attribute), 126	create_tournament_provider() (in module cassiopeia.baseriotapi), 19	cas-
content (cassiopeia.type.core.status.Translation attribute), 128	create_tournament_provider() (in module cassiopeia.core.tournamentapi), 37	cas-
cooldown_burn (cassiopeia.type.core.staticdata.Spell attribute), 115	create_tournament_provider() (in module cassiopeia.dto.tournamentapi), 46	cas-
cooldown_burn (cassiopeia.type.core.staticdata.SummonerSpell attribute), 117	create_tournament_provider() (in module cassiopeia.riotapi), 10	cas-
cooldown_reduction (cassiopeia.type.core.staticdata.ItemStats attribute), 103	created (cassiopeia.type.core.status.Incident attribute), 125	
cooldown_reduction_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 103	created (cassiopeia.type.core.status.Message attribute), 126	
cooldowns (cassiopeia.type.core.staticdata.Spell attribute), 115	creation (cassiopeia.type.core.currentgame.Game attribute), 61	
cooldowns (cassiopeia.type.core.staticdata.SummonerSpell attribute), 117	creation (cassiopeia.type.core.featuredgames.Game attribute), 63	
coop_ai_enabled (cassiopeia.type.core.champion.ChampionStats attribute), 51	creation (cassiopeia.type.core.game.Game attribute), 65	
copy() (cassiopeia.type.dto.common.JSONEncoded method), 147	creation (cassiopeia.type.core.match.Match attribute), 79	
copy_value() (cassiopeia.type.dto.common.JSONEncoded method), 147	creation (cassiopeia.type.core.team.Team attribute), 133	
cost_burn (cassiopeia.type.core.staticdata.Spell attribute), 115	creation (cassiopeia.type.core.match.Event attribute), 77	
cost_burn (cassiopeia.type.core.staticdata.SummonerSpell attribute), 117	creeps_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), 89	
cost_type (cassiopeia.type.core.staticdata.Spell attribute), 115	critical_strike_chance (cassiopeia.type.core.staticdata.ChampionStats attribute), 97	cas-
cost_type (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118	critical_strike_chance (cassiopeia.type.core.staticdata.ItemStats attribute), 104	cas-
costs (cassiopeia.type.core.staticdata.Spell attribute), 115	critical_strike_chance_per_level (cassiopeia.type.core.staticdata.ChampionStats attribute), 97	cas-
costs (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118	critical_strike_chance_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 104	cas-
count (cassiopeia.type.core.staticdata.SetItem attribute), 114	critical_strike_damage (cassiopeia.type.core.staticdata.ItemStats attribute), 104	cas-
create_tournament() (in module cassiopeia.baseriotapi), 19	critical_strike_damage_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 104	cas-

- crowd_control_dealt (cassiopeia.type.core.game.Stats attribute), 67
- crowd_control_dealt (cassiopeia.type.core.match.ParticipantStats attribute), 83
- crystal_scar (cassiopeia.type.core.tournament.MapType attribute), 136
- cs (cassiopeia.type.core.match.ParticipantStats attribute), 83
- cs_diff_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), 89
- css (cassiopeia.type.core.staticdata.Realm attribute), 111
- cunning (cassiopeia.type.core.common.MasteryType attribute), 55
- current (cassiopeia.type.core.summoner.MasteryPage attribute), 128
- current (cassiopeia.type.core.summoner.RunePage attribute), 129
- current_game() (cassiopeia.type.core.summoner.Summoner method), 130
- current_gold (cassiopeia.type.core.match.ParticipantFrame attribute), 82
- CurrentGameInfo (class in cassiopeia.type.dto.currentgame), 150
- CurrentGameParticipant (class in cassiopeia.type.dto.currentgame), 151
- custom (cassiopeia.type.core.common.GameType attribute), 54
- custom (cassiopeia.type.core.common.Queue attribute), 56
- custom (cassiopeia.type.core.common.SubType attribute), 59
- custom_enabled (cassiopeia.type.core.champion.ChampionStats attribute), 51
- D**
- d_casts (cassiopeia.type.core.game.Stats attribute), 67
- damage_dealt (cassiopeia.type.core.game.Stats attribute), 67
- damage_dealt (cassiopeia.type.core.match.ParticipantStats attribute), 83
- damage_dealt (cassiopeia.type.core.stats.AggregatedStats attribute), 121
- damage_dealt_player (cassiopeia.type.core.game.Stats attribute), 68
- damage_dealt_to_champions (cassiopeia.type.core.game.Stats attribute), 68
- damage_dealt_to_champions (cassiopeia.type.core.match.ParticipantStats attribute), 83
- damage_taken (cassiopeia.type.core.game.Stats attribute), 68
- damage_taken (cassiopeia.type.core.match.ParticipantStats attribute), 84
- damage_taken (cassiopeia.type.core.stats.AggregatedStats attribute), 121
- damage_taken_diff_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), 89
- damage_taken_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), 89
- data_type_versions (cassiopeia.type.core.staticdata.Realm attribute), 111
- DataStore (class in cassiopeia.type.api.store), 48
- date (cassiopeia.type.core.team.MatchSummary attribute), 132
- death (cassiopeia.type.core.common.Ascended attribute), 53
- deaths (cassiopeia.type.core.game.Stats attribute), 68
- deaths (cassiopeia.type.core.match.ParticipantStats attribute), 84
- deaths (cassiopeia.type.core.stats.AggregatedStats attribute), 121
- deaths (cassiopeia.type.core.team.MatchSummary attribute), 132
- default_comparator (cassiopeia.type.dto.common.JSONEncoded.Comparator attribute), 142
- defense (cassiopeia.type.core.staticdata.ChampionInfo attribute), 96
- definitely_not_dominion (cassiopeia.type.core.common.Queue attribute), 56
- deserialize() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 142
- description (cassiopeia.type.core.staticdata.Item attribute), 100
- description (cassiopeia.type.core.staticdata.Passive attribute), 111
- description (cassiopeia.type.core.staticdata.Rune attribute), 113
- description (cassiopeia.type.core.staticdata.Spell attribute), 115
- description (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- descriptions (cassiopeia.type.core.staticdata.Mastery attribute), 109
- dialect_impl() (cassiopeia.type.dto.common.JSONEncoded method), 147
- diamond (cassiopeia.type.core.common.Tier attribute), 59
- difficulty (cassiopeia.type.core.staticdata.ChampionInfo attribute), 96

dispatch (cassiopeia.type.dto.common.JSONEncoded attribute), 147

distinct() (cassiopeia.type.dto.common.JSONEncoded.Compact method), 142

division (cassiopeia.type.core.league.Entry attribute), 74

Division (class in cassiopeia.type.core.common), 53

dodge_chance (cassiopeia.type.core.staticdata.ItemStats attribute), 104

dodge_chance_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 104

dominion (cassiopeia.type.core.common.GameMode attribute), 54

dominion (cassiopeia.type.core.common.StatSummaryType attribute), 58

dominion (cassiopeia.type.core.common.SubType attribute), 59

dominion_blind (cassiopeia.type.core.common.Queue attribute), 56

dominion_draft (cassiopeia.type.core.common.Queue attribute), 56

doom_bots (cassiopeia.type.core.common.StatSummaryType attribute), 58

doom_bots (cassiopeia.type.core.common.SubType attribute), 59

doom_bots_1 (cassiopeia.type.core.common.Queue attribute), 56

doom_bots_2 (cassiopeia.type.core.common.Queue attribute), 56

doom_bots_5 (cassiopeia.type.core.common.Queue attribute), 56

double_kills (cassiopeia.type.core.game.Stats attribute), 68

double_kills (cassiopeia.type.core.match.ParticipantStats attribute), 84

double_kills (cassiopeia.type.core.stats.AggregatedStats attribute), 121

draft (cassiopeia.type.core.tournament.PickType attribute), 136

dragon (cassiopeia.type.core.common.Monster attribute), 55

dragon_assists_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

dragon_kills (cassiopeia.type.core.match.Team attribute), 92

dragon_kills_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

dragon_magic (cassiopeia.type.core.staticdata.Realm attribute), 111

drill (cassiopeia.type.core.common.Point attribute), 56

dto_type (cassiopeia.type.core.champion.ChampionStatus attribute), 51

dto_type (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52

dto_type (cassiopeia.type.core.currentgame.Ban attribute), 60

dto_type (cassiopeia.type.core.currentgame.Game attribute), 61

dto_type (cassiopeia.type.core.currentgame.Participant attribute), 62

dto_type (cassiopeia.type.core.featuredgames.Ban attribute), 63

dto_type (cassiopeia.type.core.featuredgames.Game attribute), 63

dto_type (cassiopeia.type.core.featuredgames.Participant attribute), 64

dto_type (cassiopeia.type.core.game.Game attribute), 65

dto_type (cassiopeia.type.core.game.Participant attribute), 67

dto_type (cassiopeia.type.core.game.Stats attribute), 68

dto_type (cassiopeia.type.core.league.Entry attribute), 74

dto_type (cassiopeia.type.core.league.League attribute), 75

dto_type (cassiopeia.type.core.league.Series attribute), 76

dto_type (cassiopeia.type.core.match.Ban attribute), 76

dto_type (cassiopeia.type.core.match.Event attribute), 77

dto_type (cassiopeia.type.core.match.Frame attribute), 79

dto_type (cassiopeia.type.core.match.Match attribute), 79

dto_type (cassiopeia.type.core.match.Participant attribute), 81

dto_type (cassiopeia.type.core.match.ParticipantFrame attribute), 82

dto_type (cassiopeia.type.core.match.ParticipantStats attribute), 84

dto_type (cassiopeia.type.core.match.ParticipantTimeline attribute), 89

dto_type (cassiopeia.type.core.match.ParticipantTimelineData attribute), 91

dto_type (cassiopeia.type.core.match.Position attribute), 91

dto_type (cassiopeia.type.core.match.Team attribute), 92

dto_type (cassiopeia.type.core.match.Timeline attribute), 93

dto_type (cassiopeia.type.core.matchlist.MatchReference attribute), 94

dto_type (cassiopeia.type.core.staticdata.Champion attribute), 95

dto_type (cassiopeia.type.core.staticdata.ChampionInfo attribute), 97

dto_type (cassiopeia.type.core.staticdata.ChampionStats attribute), 98

dto_type (cassiopeia.type.core.staticdata.Gold attribute), 99

dto_type (cassiopeia.type.core.staticdata.Image attribute), 99

- dto_type (cassiopeia.type.core.staticdata.Item attribute), 100
- dto_type (cassiopeia.type.core.staticdata.ItemSet attribute), 102
- dto_type (cassiopeia.type.core.staticdata.ItemStats attribute), 104
- dto_type (cassiopeia.type.core.staticdata.LevelTip attribute), 109
- dto_type (cassiopeia.type.core.staticdata.MapDetails attribute), 109
- dto_type (cassiopeia.type.core.staticdata.Mastery attribute), 109
- dto_type (cassiopeia.type.core.staticdata.MetaData attribute), 110
- dto_type (cassiopeia.type.core.staticdata.Passive attribute), 111
- dto_type (cassiopeia.type.core.staticdata.Realm attribute), 111
- dto_type (cassiopeia.type.core.staticdata.RecommendedItems attribute), 112
- dto_type (cassiopeia.type.core.staticdata.Rune attribute), 113
- dto_type (cassiopeia.type.core.staticdata.SetItem attribute), 114
- dto_type (cassiopeia.type.core.staticdata.Skin attribute), 114
- dto_type (cassiopeia.type.core.staticdata.Spell attribute), 115
- dto_type (cassiopeia.type.core.staticdata.SpellVariables attribute), 117
- dto_type (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- dto_type (cassiopeia.type.core.stats.AggregatedStats attribute), 121
- dto_type (cassiopeia.type.core.stats.StatsSummary attribute), 125
- dto_type (cassiopeia.type.core.status.Incident attribute), 125
- dto_type (cassiopeia.type.core.status.Message attribute), 126
- dto_type (cassiopeia.type.core.status.Service attribute), 126
- dto_type (cassiopeia.type.core.status.Shard attribute), 127
- dto_type (cassiopeia.type.core.status.ShardStatus attribute), 127
- dto_type (cassiopeia.type.core.status.Translation attribute), 128
- dto_type (cassiopeia.type.core.summoner.MasteryPage attribute), 129
- dto_type (cassiopeia.type.core.summoner.RunePage attribute), 129
- dto_type (cassiopeia.type.core.summoner.Summoner attribute), 130
- dto_type (cassiopeia.type.core.team.MatchSummary attribute), 132
- dto_type (cassiopeia.type.core.team.Stats attribute), 133
- dto_type (cassiopeia.type.core.team.Team attribute), 134
- dto_type (cassiopeia.type.core.team.TeamMember attribute), 135
- dto_type (cassiopeia.type.core.tournament.LobbyEvent attribute), 135
- dto_type (cassiopeia.type.core.tournament.TournamentCode attribute), 136
- duo (cassiopeia.type.core.common.Role attribute), 57
- duration (cassiopeia.type.core.currentgame.Game attribute), 61
- duration (cassiopeia.type.core.featuredgames.Game attribute), 63
- duration (cassiopeia.type.core.match.Match attribute), 79
- dynamic (cassiopeia.type.core.staticdata.SpellVariables attribute), 117
- dynamic_queue (cassiopeia.type.core.common.Queue attribute), 56
- ## E
- e_casts (cassiopeia.type.core.game.Stats attribute), 68
- eager (cassiopeia.type.core.common.LoadPolicy attribute), 54
- effect (cassiopeia.type.core.staticdata.Item attribute), 101
- effect_burn (cassiopeia.type.core.staticdata.Spell attribute), 115
- effect_burn (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- effects (cassiopeia.type.core.staticdata.LevelTip attribute), 109
- effects (cassiopeia.type.core.staticdata.Spell attribute), 115
- effects (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- elder_lizard_assists_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- elder_lizard_kills_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- elite_monster_kill (cassiopeia.type.core.common.EventType attribute), 53
- elite_monsters_kills (cassiopeia.type.core.game.Stats attribute), 68
- enabled (cassiopeia.type.core.champion.ChampionStatus attribute), 51
- endsWith() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 143
- enemy_monster_kills (cassiopeia.type.core.game.Stats attribute), 68

- enemy_monster_kills (cassiopeia.type.core.match.ParticipantStats attribute), 84
 enemy_tips (cassiopeia.type.core.staticdata.Champion attribute), 95
 energy (cassiopeia.type.core.staticdata.ItemStats attribute), 104
 energy_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 104
 energy_regen (cassiopeia.type.core.staticdata.ItemStats attribute), 104
 energy_regen_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 104
 entries (cassiopeia.type.core.league.League attribute), 75
 Entry (class in cassiopeia.type.core.league), 74
 europe_north_east (cassiopeia.type.core.common.Platform attribute), 55
 europe_north_east (cassiopeia.type.core.common.Region attribute), 57
 europe_north_east (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
 europe_west (cassiopeia.type.core.common.Platform attribute), 55
 europe_west (cassiopeia.type.core.common.Region attribute), 57
 europe_west (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
 evaluates_none() (cassiopeia.type.dto.common.JSONEncoded method), 148
 Event (class in cassiopeia.type.core.match), 77
 Event (class in cassiopeia.type.dto.match), 159
 events (cassiopeia.type.core.match.Frame attribute), 79
 EventType (class in cassiopeia.type.core.common), 53
 evolution (cassiopeia.type.core.common.LevelUp attribute), 54
 execute_request() (in module cassiopeia.dto.requests), 41
 expr (cassiopeia.type.dto.common.JSONEncoded.Comparator attribute), 143
- ## F
- f_casts (cassiopeia.type.core.game.Stats attribute), 68
 FeaturedGameInfo (class in cassiopeia.type.dto.featuredgames), 152
 FeaturedGames (class in cassiopeia.type.dto.featuredgames), 153
 ferocity (cassiopeia.type.core.common.MasteryType attribute), 55
 first_baron (cassiopeia.type.core.match.Team attribute), 92
 first_blood (cassiopeia.type.core.game.Stats attribute), 68
 first_blood (cassiopeia.type.core.match.ParticipantStats attribute), 84
 first_blood (cassiopeia.type.core.match.Team attribute), 92
 first_blood_assist (cassiopeia.type.core.match.ParticipantStats attribute), 84
 first_bloods (cassiopeia.type.core.stats.AggregatedStats attribute), 121
 first_dragon (cassiopeia.type.core.match.Team attribute), 92
 first_inhibitor (cassiopeia.type.core.match.ParticipantStats attribute), 84
 first_inhibitor (cassiopeia.type.core.match.Team attribute), 92
 first_inhibitor_assist (cassiopeia.type.core.match.ParticipantStats attribute), 84
 first_rift_herald (cassiopeia.type.core.match.Team attribute), 92
 first_turret (cassiopeia.type.core.match.ParticipantStats attribute), 84
 first_turret (cassiopeia.type.core.match.Team attribute), 92
 first_turret_assist (cassiopeia.type.core.match.ParticipantStats attribute), 84
 five (cassiopeia.type.core.common.Division attribute), 53
 flex (cassiopeia.type.core.common.Queue attribute), 56
 flex (cassiopeia.type.core.common.SubType attribute), 59
 flex_summoners_rift (cassiopeia.type.core.common.StatSummaryType attribute), 58
 flex_twisted_treeline (cassiopeia.type.core.common.StatSummaryType attribute), 58
 fountain (cassiopeia.type.core.common.Turret attribute), 60
 four (cassiopeia.type.core.common.Division attribute), 53
 Frame (class in cassiopeia.type.core.match), 79
 Frame (class in cassiopeia.type.dto.match), 160
 frame_interval (cassiopeia.type.core.match.Timeline attribute), 93
 frames (cassiopeia.type.core.match.Match attribute), 79
 frames (cassiopeia.type.core.match.Timeline attribute), 93
 free (cassiopeia.type.core.champion.ChampionStatus attribute), 51
 fresh_blood (cassiopeia.type.core.league.Entry attribute), 74
- ## G
- Game (class in cassiopeia.type.core.currentgame), 61
 Game (class in cassiopeia.type.core.featuredgames), 63
 Game (class in cassiopeia.type.core.game), 65
 Game (class in cassiopeia.type.dto.game), 153

GameMode (class in cassiopeia.type.core.common), 54
 games_played (cassiopeia.type.core.stats.AggregatedStats attribute), 121
 GameType (class in cassiopeia.type.core.common), 54
 get() (cassiopeia.type.api.store.Cache method), 48
 get() (cassiopeia.type.api.store.DataStore method), 48
 get() (cassiopeia.type.api.store.SQLAlchemyDB method), 49
 get() (cassiopeia.type.api.store.VoidDataStore method), 50
 get() (in module cassiopeia.dto.requests), 41
 get_all() (cassiopeia.type.api.store.Cache method), 48
 get_all() (cassiopeia.type.api.store.DataStore method), 49
 get_all() (cassiopeia.type.api.store.SQLAlchemyDB method), 50
 get_all() (cassiopeia.type.api.store.VoidDataStore method), 50
 get_challenger() (in module cassiopeia.baseriotapi), 19
 get_challenger() (in module cassiopeia.core.leagueapi), 30
 get_challenger() (in module cassiopeia.dto.leagueapi), 39
 get_challenger() (in module cassiopeia.riotapi), 10
 get_champion() (in module cassiopeia.baseriotapi), 20
 get_champion() (in module cassiopeia.dto.staticdataapi), 42
 get_champion_by_id() (in module cassiopeia.core.staticdataapi), 32
 get_champion_by_id() (in module cassiopeia.riotapi), 10
 get_champion_by_name() (in module cassiopeia.core.staticdataapi), 32
 get_champion_by_name() (in module cassiopeia.riotapi), 10
 get_champion_mastersies() (in module cassiopeia.baseriotapi), 20
 get_champion_mastersies() (in module cassiopeia.core.championmasteryapi), 29
 get_champion_mastersies() (in module cassiopeia.dto.championmasteryapi), 38
 get_champion_mastersies() (in module cassiopeia.riotapi), 10
 get_champion_mastery() (in module cassiopeia.baseriotapi), 20
 get_champion_mastery() (in module cassiopeia.core.championmasteryapi), 29
 get_champion_mastery() (in module cassiopeia.dto.championmasteryapi), 39
 get_champion_mastery() (in module cassiopeia.riotapi), 10
 get_champion_mastery_score() (in module cassiopeia.baseriotapi), 20
 get_champion_mastery_score() (in module cassiopeia.core.championmasteryapi), 29
 get_champion_mastery_score() (in module cassiopeia.dto.championmasteryapi), 39
 get_champion_mastery_score() (in module cassiopeia.riotapi), 10
 get_champion_status() (in module cassiopeia.baseriotapi), 20
 get_champion_status() (in module cassiopeia.core.championapi), 29
 get_champion_status() (in module cassiopeia.dto.championapi), 38
 get_champion_statuses() (in module cassiopeia.baseriotapi), 20
 get_champion_statuses() (in module cassiopeia.core.championapi), 29
 get_champion_statuses() (in module cassiopeia.dto.championapi), 38
 get_champions() (in module cassiopeia.baseriotapi), 20
 get_champions() (in module cassiopeia.core.staticdataapi), 33
 get_champions() (in module cassiopeia.dto.staticdataapi), 42
 get_champions() (in module cassiopeia.riotapi), 11
 get_champions_by_id() (in module cassiopeia.core.staticdataapi), 33
 get_champions_by_id() (in module cassiopeia.riotapi), 11
 get_champions_by_name() (in module cassiopeia.core.staticdataapi), 33
 get_champions_by_name() (in module cassiopeia.riotapi), 11
 get_current_game() (in module cassiopeia.baseriotapi), 21
 get_current_game() (in module cassiopeia.core.currentgameapi), 30
 get_current_game() (in module cassiopeia.dto.currentgameapi), 39
 get_current_game() (in module cassiopeia.riotapi), 11
 get_dbapi_type() (cassiopeia.type.dto.common.JSONEncoded method), 148
 get_featured_games() (in module cassiopeia.baseriotapi), 21
 get_featured_games() (in module cassiopeia.core.featuredgamesapi), 30
 get_featured_games() (in module cassiopeia.dto.featuredgamesapi), 39
 get_featured_games() (in module cassiopeia.riotapi), 11
 get_item() (in module cassiopeia.baseriotapi), 21
 get_item() (in module cassiopeia.core.staticdataapi), 33
 get_item() (in module cassiopeia.dto.staticdataapi), 42
 get_item() (in module cassiopeia.riotapi), 11
 get_items() (in module cassiopeia.baseriotapi), 21
 get_items() (in module cassiopeia.core.staticdataapi), 33
 get_items() (in module cassiopeia.dto.staticdataapi), 42
 get_items() (in module cassiopeia.riotapi), 11
 get_language_strings() (in module cassiopeia.baseriotapi), 21

`get_language_strings()` (in module `cas-siopeia.core.staticdataapi`), 33

`get_language_strings()` (in module `cas-siopeia.dto.staticdataapi`), 42

`get_language_strings()` (in module `cassiopeia.riotapi`), 11

`get_languages()` (in module `cassiopeia.baseriotapi`), 21

`get_languages()` (in module `cassiopeia.core.staticdataapi`), 33

`get_languages()` (in module `cassiopeia.dto.staticdataapi`), 43

`get_languages()` (in module `cassiopeia.riotapi`), 12

`get_league_entries_by_summoner()` (in module `cas-siopeia.baseriotapi`), 21

`get_league_entries_by_summoner()` (in module `cas-siopeia.core.leagueapi`), 30

`get_league_entries_by_summoner()` (in module `cas-siopeia.dto.leagueapi`), 40

`get_league_entries_by_summoner()` (in module `cas-siopeia.riotapi`), 12

`get_league_entries_by_team()` (in module `cas-siopeia.baseriotapi`), 21

`get_league_entries_by_team()` (in module `cas-siopeia.core.leagueapi`), 30

`get_league_entries_by_team()` (in module `cas-siopeia.dto.leagueapi`), 40

`get_league_entries_by_team()` (in module `cas-siopeia.riotapi`), 12

`get_leagues_by_summoner()` (in module `cas-siopeia.baseriotapi`), 22

`get_leagues_by_summoner()` (in module `cas-siopeia.core.leagueapi`), 31

`get_leagues_by_summoner()` (in module `cas-siopeia.dto.leagueapi`), 40

`get_leagues_by_summoner()` (in module `cas-siopeia.riotapi`), 12

`get_leagues_by_team()` (in module `cas-siopeia.baseriotapi`), 22

`get_leagues_by_team()` (in module `cas-siopeia.core.leagueapi`), 31

`get_leagues_by_team()` (in module `cas-siopeia.dto.leagueapi`), 40

`get_leagues_by_team()` (in module `cassiopeia.riotapi`), 12

`get_lobby_events()` (in module `cassiopeia.baseriotapi`), 22

`get_lobby_events()` (in module `cas-siopeia.core.tournamentapi`), 37

`get_lobby_events()` (in module `cas-siopeia.dto.tournamentapi`), 46

`get_lobby_events()` (in module `cassiopeia.riotapi`), 12

`get_map_information()` (in module `cas-siopeia.core.staticdataapi`), 33

`get_map_information()` (in module `cassiopeia.riotapi`), 12

`get_maps()` (in module `cassiopeia.baseriotapi`), 22

`get_maps()` (in module `cassiopeia.dto.staticdataapi`), 43

`get_master()` (in module `cassiopeia.baseriotapi`), 22

`get_master()` (in module `cassiopeia.core.staticdataapi`), 34

`get_master()` (in module `cassiopeia.dto.staticdataapi`), 43

`get_master()` (in module `cassiopeia.riotapi`), 13

`get_masters()` (in module `cassiopeia.baseriotapi`), 22

`get_masters()` (in module `cassiopeia.core.staticdataapi`), 34

`get_masters()` (in module `cassiopeia.dto.staticdataapi`), 43

`get_masters()` (in module `cassiopeia.riotapi`), 13

`get_mastery()` (in module `cassiopeia.baseriotapi`), 22

`get_mastery()` (in module `cassiopeia.core.staticdataapi`), 34

`get_mastery()` (in module `cassiopeia.dto.staticdataapi`), 43

`get_mastery()` (in module `cassiopeia.riotapi`), 13

`get_mastery_pages()` (in module `cas-siopeia.core.summonerapi`), 35

`get_mastery_pages()` (in module `cassiopeia.riotapi`), 13

`get_match()` (in module `cassiopeia.baseriotapi`), 22

`get_match()` (in module `cassiopeia.core.matchapi`), 31

`get_match()` (in module `cassiopeia.dto.matchapi`), 40

`get_match()` (in module `cassiopeia.riotapi`), 13

`get_match_list()` (in module `cassiopeia.baseriotapi`), 23

`get_match_list()` (in module `cas-siopeia.core.matchlistapi`), 32

`get_match_list()` (in module `cassiopeia.dto.matchlistapi`), 41

`get_match_list()` (in module `cassiopeia.riotapi`), 13

`get_matches()` (in module `cassiopeia.core.matchapi`), 31

`get_matches()` (in module `cassiopeia.riotapi`), 14

`get_name()` (`cassiopeia.type.api.store.HasAllStatus` static method), 49

`get_ranked_stats()` (in module `cassiopeia.baseriotapi`), 23

`get_ranked_stats()` (in module `cassiopeia.core.statsapi`), 35

`get_ranked_stats()` (in module `cassiopeia.dto.statsapi`), 44

`get_ranked_stats()` (in module `cassiopeia.riotapi`), 14

`get_realm()` (in module `cassiopeia.baseriotapi`), 23

`get_realm()` (in module `cassiopeia.core.staticdataapi`), 34

`get_realm()` (in module `cassiopeia.dto.staticdataapi`), 43

`get_realm()` (in module `cassiopeia.riotapi`), 14

`get_recent_games()` (in module `cassiopeia.baseriotapi`), 23

`get_recent_games()` (in module `cassiopeia.core.gameapi`), 30

`get_recent_games()` (in module `cassiopeia.dto.gameapi`), 39

`get_recent_games()` (in module `cassiopeia.riotapi`), 14

`get_requests_count()` (in module `cassiopeia.baseriotapi`), 24

`get_requests_count()` (in module `cassiopeia.riotapi`), 14

`get_rune()` (in module `cassiopeia.baseriotapi`), 24

`get_rune()` (in module `cassiopeia.core.staticdataapi`), 34

`get_rune()` (in module `cassiopeia.dto.staticdataapi`), 43

`get_rune()` (in module `cassiopeia.riotapi`), 15

<code>get_rune_pages()</code>	(in module <code>cas-siopeia.core.summonerapi</code>), 35
<code>get_rune_pages()</code>	(in module <code>cassiopeia.riotapi</code>), 15
<code>get_runes()</code>	(in module <code>cassiopeia.baseriotapi</code>), 24
<code>get_runes()</code>	(in module <code>cassiopeia.core.staticdataapi</code>), 34
<code>get_runes()</code>	(in module <code>cassiopeia.dto.staticdataapi</code>), 43
<code>get_runes()</code>	(in module <code>cassiopeia.riotapi</code>), 15
<code>get_shard()</code>	(in module <code>cassiopeia.baseriotapi</code>), 24
<code>get_shard()</code>	(in module <code>cassiopeia.core.statusapi</code>), 35
<code>get_shard()</code>	(in module <code>cassiopeia.dto.statusapi</code>), 44
<code>get_shard()</code>	(in module <code>cassiopeia.riotapi</code>), 15
<code>get_shards()</code>	(in module <code>cassiopeia.baseriotapi</code>), 24
<code>get_shards()</code>	(in module <code>cassiopeia.core.statusapi</code>), 35
<code>get_shards()</code>	(in module <code>cassiopeia.dto.statusapi</code>), 44
<code>get_shards()</code>	(in module <code>cassiopeia.riotapi</code>), 15
<code>get_stats()</code>	(in module <code>cassiopeia.baseriotapi</code>), 24
<code>get_stats()</code>	(in module <code>cassiopeia.core.statsapi</code>), 35
<code>get_stats()</code>	(in module <code>cassiopeia.dto.statsapi</code>), 44
<code>get_stats()</code>	(in module <code>cassiopeia.riotapi</code>), 15
<code>get_summoner_by_id()</code>	(in module <code>cas-siopeia.core.summonerapi</code>), 35
<code>get_summoner_by_id()</code>	(in module <code>cassiopeia.riotapi</code>), 15
<code>get_summoner_by_name()</code>	(in module <code>cas-siopeia.core.summonerapi</code>), 36
<code>get_summoner_by_name()</code>	(in module <code>cas-siopeia.riotapi</code>), 16
<code>get_summoner_masteries()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 24
<code>get_summoner_masteries()</code>	(in module <code>cas-siopeia.dto.summonerapi</code>), 44
<code>get_summoner_name()</code>	(in module <code>cas-siopeia.core.summonerapi</code>), 36
<code>get_summoner_name()</code>	(in module <code>cassiopeia.riotapi</code>), 16
<code>get_summoner_names()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 25
<code>get_summoner_names()</code>	(in module <code>cas-siopeia.core.summonerapi</code>), 36
<code>get_summoner_names()</code>	(in module <code>cas-siopeia.dto.summonerapi</code>), 44
<code>get_summoner_names()</code>	(in module <code>cassiopeia.riotapi</code>), 16
<code>get_summoner_runes()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 25
<code>get_summoner_runes()</code>	(in module <code>cas-siopeia.dto.summonerapi</code>), 45
<code>get_summoner_spell()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 25
<code>get_summoner_spell()</code>	(in module <code>cas-siopeia.core.staticdataapi</code>), 34
<code>get_summoner_spell()</code>	(in module <code>cas-siopeia.dto.staticdataapi</code>), 43
<code>get_summoner_spell()</code>	(in module <code>cassiopeia.riotapi</code>), 16
<code>get_summoner_spells()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 25
<code>get_summoner_spells()</code>	(in module <code>cas-siopeia.core.staticdataapi</code>), 34
<code>get_summoner_spells()</code>	(in module <code>cas-siopeia.dto.staticdataapi</code>), 43
<code>get_summoner_spells()</code>	(in module <code>cassiopeia.riotapi</code>), 16
<code>get_summoners_by_id()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 25
<code>get_summoners_by_id()</code>	(in module <code>cas-siopeia.core.summonerapi</code>), 36
<code>get_summoners_by_id()</code>	(in module <code>cas-siopeia.dto.summonerapi</code>), 45
<code>get_summoners_by_id()</code>	(in module <code>cassiopeia.riotapi</code>), 16
<code>get_summoners_by_name()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 25
<code>get_summoners_by_name()</code>	(in module <code>cas-siopeia.core.summonerapi</code>), 36
<code>get_summoners_by_name()</code>	(in module <code>cas-siopeia.dto.summonerapi</code>), 45
<code>get_summoners_by_name()</code>	(in module <code>cas-siopeia.riotapi</code>), 16
<code>get_team()</code>	(in module <code>cassiopeia.core.teamapi</code>), 36
<code>get_team()</code>	(in module <code>cassiopeia.riotapi</code>), 16
<code>get_teams()</code>	(in module <code>cassiopeia.core.teamapi</code>), 36
<code>get_teams()</code>	(in module <code>cassiopeia.riotapi</code>), 17
<code>get_teams_by_id()</code>	(in module <code>cassiopeia.baseriotapi</code>), 25
<code>get_teams_by_id()</code>	(in module <code>cassiopeia.dto.teamapi</code>), 45
<code>get_teams_by_summoner()</code>	(in module <code>cas-siopeia.core.teamapi</code>), 36
<code>get_teams_by_summoner()</code>	(in module <code>cas-siopeia.riotapi</code>), 17
<code>get_teams_by_summoner_id()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 25
<code>get_teams_by_summoner_id()</code>	(in module <code>cas-siopeia.dto.teamapi</code>), 45
<code>get_top_champion_masteries()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 26
<code>get_top_champion_masteries()</code>	(in module <code>cas-siopeia.core.championmasteryapi</code>), 30
<code>get_top_champion_masteries()</code>	(in module <code>cas-siopeia.dto.championmasteryapi</code>), 39
<code>get_top_champion_masteries()</code>	(in module <code>cas-siopeia.riotapi</code>), 17
<code>get_tournament_code()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 26
<code>get_tournament_code()</code>	(in module <code>cas-siopeia.core.tournamentapi</code>), 38
<code>get_tournament_code()</code>	(in module <code>cas-siopeia.dto.tournamentapi</code>), 46
<code>get_tournament_code()</code>	(in module <code>cassiopeia.riotapi</code>), 17
<code>get_tournament_match_ids()</code>	(in module <code>cas-siopeia.baseriotapi</code>), 26

`get_tournament_match_ids()` (in module `cassiopeia.core.matchapi`), 31

`get_tournament_match_ids()` (in module `cassiopeia.dto.matchapi`), 40

`get_tournament_match_ids()` (in module `cassiopeia.riotapi`), 17

`get_versions()` (in module `cassiopeia.baseriotapi`), 26

`get_versions()` (in module `cassiopeia.core.staticdataapi`), 34

`get_versions()` (in module `cassiopeia.dto.staticdataapi`), 44

`get_versions()` (in module `cassiopeia.riotapi`), 17

`gold` (`cassiopeia.type.core.common.Tier` attribute), 59

`gold` (`cassiopeia.type.core.game.Stats` attribute), 68

`gold` (`cassiopeia.type.core.match.ParticipantFrame` attribute), 82

`gold` (`cassiopeia.type.core.staticdata.Item` attribute), 101

`Gold` (class in `cassiopeia.type.core.staticdata`), 99

`Gold` (class in `cassiopeia.type.dto.staticdata`), 170

`gold_earned` (`cassiopeia.type.core.game.Stats` attribute), 68

`gold_earned` (`cassiopeia.type.core.match.ParticipantStats` attribute), 84

`gold_earned` (`cassiopeia.type.core.stats.AggregatedStats` attribute), 121

`gold_per_min_deltas` (`cassiopeia.type.core.match.ParticipantTimeline` attribute), 90

`gold_per_ten` (`cassiopeia.type.core.staticdata.ItemStats` attribute), 104

`gold_spent` (`cassiopeia.type.core.game.Stats` attribute), 69

`gold_spent` (`cassiopeia.type.core.match.ParticipantStats` attribute), 84

`group` (`cassiopeia.type.core.staticdata.Image` attribute), 99

`group` (`cassiopeia.type.core.staticdata.Item` attribute), 101

`Group` (class in `cassiopeia.type.dto.staticdata`), 170

H

`has_all()` (`cassiopeia.type.api.store.Cache` method), 48

`has_all()` (`cassiopeia.type.api.store.DataStore` method), 49

`has_all()` (`cassiopeia.type.api.store.SQLAlchemyDB` method), 50

`has_all()` (`cassiopeia.type.api.store.VoidDataStore` method), 51

`HasAllStatus` (class in `cassiopeia.type.api.store`), 49

`hashable` (`cassiopeia.type.dto.common.JSONEncoded` attribute), 148

`have_all` (`cassiopeia.type.api.store.HasAllStatus` attribute), 49

`healing_done` (`cassiopeia.type.core.game.Stats` attribute), 69

`healing_done` (`cassiopeia.type.core.match.ParticipantStats` attribute), 85

`healing_done` (`cassiopeia.type.core.stats.AggregatedStats` attribute), 121

`health` (`cassiopeia.type.core.staticdata.ChampionStats` attribute), 98

`health` (`cassiopeia.type.core.staticdata.ItemStats` attribute), 104

`health_per_level` (`cassiopeia.type.core.staticdata.ChampionStats` attribute), 98

`health_per_level` (`cassiopeia.type.core.staticdata.ItemStats` attribute), 105

`health_regen` (`cassiopeia.type.core.staticdata.ChampionStats` attribute), 98

`health_regen` (`cassiopeia.type.core.staticdata.ItemStats` attribute), 105

`health_regen_per_level` (`cassiopeia.type.core.staticdata.ChampionStats` attribute), 98

`health_regen_per_level` (`cassiopeia.type.core.staticdata.ItemStats` attribute), 105

`height` (`cassiopeia.type.core.staticdata.Image` attribute), 99

`hexakill_summoners_rift` (`cassiopeia.type.core.common.Queue` attribute), 56

`hexakill_summoners_rift` (`cassiopeia.type.core.common.StatSummaryType` attribute), 58

`hexakill_summoners_rift` (`cassiopeia.type.core.common.SubType` attribute), 59

`hexakill_twisted_treeline` (`cassiopeia.type.core.common.Queue` attribute), 56

`hexakill_twisted_treeline` (`cassiopeia.type.core.common.StatSummaryType` attribute), 58

`hexakill_twisted_treeline` (`cassiopeia.type.core.common.SubType` attribute), 59

`hide` (`cassiopeia.type.core.staticdata.Item` attribute), 101

`host_name` (`cassiopeia.type.core.status.Shard` attribute), 127

`host_name` (`cassiopeia.type.core.status.ShardStatus` attribute), 127

`hot_streak` (`cassiopeia.type.core.league.Entry` attribute), 74

`howling_abyss` (`cassiopeia.type.core.common.Map` attribute), 55

`howling_abyss` (`cassiopeia.type.core.tournament.MapType` attribute), 136

I

`id` (`cassiopeia.type.core.currentgame.Game` attribute), 61

- id (cassiopeia.type.core.featuredgames.Game attribute), 63
- id (cassiopeia.type.core.game.Game attribute), 65
- id (cassiopeia.type.core.match.Match attribute), 80
- id (cassiopeia.type.core.match.Participant attribute), 81
- id (cassiopeia.type.core.matchlist.MatchReference attribute), 94
- id (cassiopeia.type.core.staticdata.Champion attribute), 95
- id (cassiopeia.type.core.staticdata.Item attribute), 101
- id (cassiopeia.type.core.staticdata.Mastery attribute), 109
- id (cassiopeia.type.core.staticdata.Rune attribute), 113
- id (cassiopeia.type.core.staticdata.Skin attribute), 114
- id (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- id (cassiopeia.type.core.status.Incident attribute), 125
- id (cassiopeia.type.core.status.Message attribute), 126
- id (cassiopeia.type.core.summoner.MasteryPage attribute), 129
- id (cassiopeia.type.core.summoner.RunePage attribute), 129
- id (cassiopeia.type.core.summoner.Summoner attribute), 130
- id (cassiopeia.type.core.team.MatchSummary attribute), 132
- id (cassiopeia.type.core.team.Team attribute), 134
- id (cassiopeia.type.core.tournament.TournamentCode attribute), 136
- ilike() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 143
- image (cassiopeia.type.core.staticdata.Champion attribute), 95
- image (cassiopeia.type.core.staticdata.Item attribute), 101
- image (cassiopeia.type.core.staticdata.MapDetails attribute), 109
- image (cassiopeia.type.core.staticdata.Mastery attribute), 110
- image (cassiopeia.type.core.staticdata.Passive attribute), 111
- image (cassiopeia.type.core.staticdata.Rune attribute), 113
- image (cassiopeia.type.core.staticdata.Spell attribute), 115
- image (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- Image (class in cassiopeia.type.core.staticdata), 99
- Image (class in cassiopeia.type.dto.staticdata), 170
- immutablemethod (class in cassiopeia.type.core.common), 60
- impl (cassiopeia.type.dto.common.JSONEncoded attribute), 148
- in_() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 143
- in_store (cassiopeia.type.core.staticdata.Item attribute), 101
- inactive (cassiopeia.type.core.league.Entry attribute), 74
- Incident (class in cassiopeia.type.core.status), 125
- Incident (class in cassiopeia.type.dto.status), 182
- incidents (cassiopeia.type.core.status.Service attribute), 126
- info (cassiopeia.type.core.staticdata.Champion attribute), 95
- Info (class in cassiopeia.type.dto.staticdata), 170
- inheritdocs() (in module cassiopeia.type.core.common), 60
- inhibitor (cassiopeia.type.core.common.Building attribute), 53
- inhibitor (cassiopeia.type.core.common.Turret attribute), 60
- inhibitor_assists_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- inhibitor_kills (cassiopeia.type.core.game.Stats attribute), 69
- inhibitor_kills (cassiopeia.type.core.match.ParticipantStats attribute), 85
- inhibitor_kills (cassiopeia.type.core.match.Team attribute), 92
- inhibitor_kills_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- inhibitor (cassiopeia.type.core.common.Turret attribute), 60
- invalid (cassiopeia.type.core.game.Game attribute), 65
- invalid (cassiopeia.type.core.team.MatchSummary attribute), 132
- invite (cassiopeia.type.core.team.TeamMember attribute), 135
- ip (cassiopeia.type.core.game.Game attribute), 65
- is_() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 143
- is_distinct_from() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 143
- isnot() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 143
- isnot_distinct_from() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 144
- item (cassiopeia.type.core.match.Event attribute), 77
- item (cassiopeia.type.core.staticdata.SetItem attribute), 114
- Item (class in cassiopeia.type.core.staticdata), 100
- Item (class in cassiopeia.type.dto.staticdata), 170
- item0 (cassiopeia.type.core.game.Stats attribute), 69
- item0 (cassiopeia.type.core.match.ParticipantStats attribute), 85
- item1 (cassiopeia.type.core.game.Stats attribute), 69

- item1 (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - item2 (cassiopeia.type.core.game.Stats attribute), 69
 - item2 (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - item3 (cassiopeia.type.core.game.Stats attribute), 69
 - item3 (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - item4 (cassiopeia.type.core.game.Stats attribute), 69
 - item4 (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - item5 (cassiopeia.type.core.game.Stats attribute), 69
 - item5 (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - item6 (cassiopeia.type.core.game.Stats attribute), 69
 - item6 (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - item_after (cassiopeia.type.core.match.Event attribute), 77
 - item_before (cassiopeia.type.core.match.Event attribute), 77
 - item_destruction (cassiopeia.type.core.common.EventType attribute), 53
 - item_ids (cassiopeia.type.dto.game.Game attribute), 154
 - item_ids (cassiopeia.type.dto.game.RecentGames attribute), 157
 - item_ids (cassiopeia.type.dto.match.MatchDetail attribute), 161
 - item_ids (cassiopeia.type.dto.staticdata.Champion attribute), 168
 - item_ids (cassiopeia.type.dto.staticdata.ChampionList attribute), 169
 - item_ids (cassiopeia.type.dto.staticdata.Item attribute), 172
 - item_purchase (cassiopeia.type.core.common.EventType attribute), 53
 - item_sale (cassiopeia.type.core.common.EventType attribute), 53
 - item_sets (cassiopeia.type.core.staticdata.RecommendedItems attribute), 112
 - item_undo (cassiopeia.type.core.common.EventType attribute), 53
 - ItemList (class in cassiopeia.type.dto.staticdata), 173
 - items (cassiopeia.type.core.game.Stats attribute), 69
 - items (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - items (cassiopeia.type.core.staticdata.ItemSet attribute), 102
 - items_bought (cassiopeia.type.core.game.Stats attribute), 69
 - ItemSet (class in cassiopeia.type.core.staticdata), 102
 - ItemStats (class in cassiopeia.type.core.staticdata), 102
 - ItemTree (class in cassiopeia.type.dto.staticdata), 173
 - iterate() (cassiopeia.type.api.store.Cache method), 48
 - iterate() (cassiopeia.type.api.store.DataStore method), 49
 - iterate() (cassiopeia.type.api.store.SQLAlchemyDB method), 50
 - iterate() (cassiopeia.type.api.store.VoidDataStore method), 51
- ## J
- japan (cassiopeia.type.core.common.Region attribute), 57
 - japan (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
 - join (cassiopeia.type.core.team.TeamMember attribute), 135
 - JSONEncoded (class in cassiopeia.type.dto.common), 139
 - JSONEncoded.Comparator (class in cassiopeia.type.dto.common), 139
 - jungle (cassiopeia.type.core.common.Lane attribute), 54
 - jungle_monsters_killed (cassiopeia.type.core.match.ParticipantFrame attribute), 82
- ## K
- kda (cassiopeia.type.core.game.Stats attribute), 70
 - kda (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - kda (cassiopeia.type.core.stats.AggregatedStats attribute), 121
 - kda (cassiopeia.type.core.team.MatchSummary attribute), 132
 - key (cassiopeia.type.core.staticdata.Champion attribute), 95
 - key (cassiopeia.type.core.staticdata.Spell attribute), 115
 - key (cassiopeia.type.core.staticdata.SpellVariables attribute), 117
 - key (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
 - keywords (cassiopeia.type.core.staticdata.Item attribute), 101
 - kill (cassiopeia.type.core.common.EventType attribute), 53
 - killer (cassiopeia.type.core.match.Event attribute), 77
 - killing_sprees (cassiopeia.type.core.game.Stats attribute), 70
 - killing_sprees (cassiopeia.type.core.match.ParticipantStats attribute), 85
 - killing_sprees (cassiopeia.type.core.stats.AggregatedStats attribute), 122
 - kills (cassiopeia.type.core.game.Stats attribute), 70
 - kills (cassiopeia.type.core.match.ParticipantStats attribute), 86
 - kills (cassiopeia.type.core.stats.AggregatedStats attribute), 122
 - kills (cassiopeia.type.core.team.MatchSummary attribute), 132

- korea (cassiopeia.type.core.common.Platform attribute), 55
- korea (cassiopeia.type.core.common.Region attribute), 57
- korea (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
- ## L
- labels (cassiopeia.type.core.staticdata.LevelTip attribute), 109
- lane (cassiopeia.type.core.game.Stats attribute), 70
- lane (cassiopeia.type.core.match.Event attribute), 78
- lane (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- lane (cassiopeia.type.core.matchlist.MatchReference attribute), 94
- Lane (class in cassiopeia.type.core.common), 54
- LaneType (class in cassiopeia.type.core.common), 54
- language (cassiopeia.type.core.staticdata.Realm attribute), 111
- LanguageStrings (class in cassiopeia.type.dto.staticdata), 173
- largest_critical_strike (cassiopeia.type.core.game.Stats attribute), 70
- largest_critical_strike (cassiopeia.type.core.match.ParticipantStats attribute), 86
- largest_killing_spree (cassiopeia.type.core.game.Stats attribute), 70
- largest_killing_spree (cassiopeia.type.core.match.ParticipantStats attribute), 86
- largest_multi_kill (cassiopeia.type.core.game.Stats attribute), 70
- largest_multi_kill (cassiopeia.type.core.match.ParticipantStats attribute), 86
- last_game (cassiopeia.type.core.team.Team attribute), 134
- last_join (cassiopeia.type.core.team.Team attribute), 134
- last_played (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52
- last_queue (cassiopeia.type.core.team.Team attribute), 134
- latin_america_north (cassiopeia.type.core.common.Platform attribute), 55
- latin_america_north (cassiopeia.type.core.common.Region attribute), 57
- latin_america_north (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
- latin_america_south (cassiopeia.type.core.common.Platform attribute), 55
- latin_america_south (cassiopeia.type.core.common.Region attribute), 57
- latin_america_south (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
- lazy (cassiopeia.type.core.common.LoadPolicy attribute), 54
- LazyProperty (class in cassiopeia.type.core.common), 54
- lazyproperty() (in module cassiopeia.type.core.common), 60
- League (class in cassiopeia.type.core.league), 75
- League (class in cassiopeia.type.dto.league), 157
- league_entries() (cassiopeia.type.core.summoner.Summoner method), 130
- league_entries() (cassiopeia.type.core.team.Team method), 134
- league_points (cassiopeia.type.core.league.Entry attribute), 74
- LeagueEntry (class in cassiopeia.type.dto.league), 158
- leagues() (cassiopeia.type.core.summoner.Summoner method), 130
- leagues() (cassiopeia.type.core.team.Team method), 134
- legacy (cassiopeia.type.core.staticdata.Realm attribute), 111
- level (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52
- level (cassiopeia.type.core.game.Game attribute), 65
- level (cassiopeia.type.core.game.Stats attribute), 70
- level (cassiopeia.type.core.match.ParticipantFrame attribute), 82
- level (cassiopeia.type.core.summoner.Summoner attribute), 130
- level_tip (cassiopeia.type.core.staticdata.Spell attribute), 115
- level_up (cassiopeia.type.core.match.Event attribute), 78
- leveltip (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- LevelTip (class in cassiopeia.type.core.staticdata), 108
- LevelTip (class in cassiopeia.type.dto.staticdata), 173
- LevelUp (class in cassiopeia.type.core.common), 54
- life_steal (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- like() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 144
- link (cassiopeia.type.core.staticdata.Image attribute), 99
- link (cassiopeia.type.core.staticdata.SpellVariables attribute), 117
- literal_processor() (cassiopeia.type.dto.common.JSONEncoded method), 148
- load_dialect_impl() (cassiopeia.type.dto.common.JSONEncoded method), 149

- loading (cassiopeia.type.core.staticdata.Skin attribute), 114
- LoadPolicy (class in cassiopeia.type.core.common), 54
- lobby (cassiopeia.type.core.tournament.SpectatorType attribute), 136
- LobbyEvent (class in cassiopeia.type.core.tournament), 135
- LobbyEvent (class in cassiopeia.type.dto.tournament), 186
- LobbyEventWrapper (class in cassiopeia.type.dto.tournament), 187
- locale (cassiopeia.type.core.status.Translation attribute), 128
- locales (cassiopeia.type.core.status.Shard attribute), 127
- locales (cassiopeia.type.core.status.ShardStatus attribute), 128
- lore (cassiopeia.type.core.staticdata.Champion attribute), 95
- losses (cassiopeia.type.core.league.Entry attribute), 74
- losses (cassiopeia.type.core.league.Series attribute), 76
- losses (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- losses (cassiopeia.type.core.stats.StatsSummary attribute), 125
- losses (cassiopeia.type.core.team.Stats attribute), 133
- M**
- magic (cassiopeia.type.core.staticdata.ChampionInfo attribute), 97
- magic_damage_dealt (cassiopeia.type.core.game.Stats attribute), 70
- magic_damage_dealt (cassiopeia.type.core.match.ParticipantStats attribute), 86
- magic_damage_dealt (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- magic_damage_dealt_to_champions (cassiopeia.type.core.game.Stats attribute), 70
- magic_damage_dealt_to_champions (cassiopeia.type.core.match.ParticipantStats attribute), 86
- magic_damage_taken (cassiopeia.type.core.game.Stats attribute), 70
- magic_damage_taken (cassiopeia.type.core.match.ParticipantStats attribute), 86
- magic_penetration (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- magic_penetration_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- magic_resist (cassiopeia.type.core.staticdata.ChampionStats attribute), 98
- magic_resist (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- magic_resist_per_level (cassiopeia.type.core.staticdata.ChampionStats attribute), 98
- magic_resist_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- make_request() (in module cassiopeia.dto.requests), 41
- mana (cassiopeia.type.core.staticdata.ChampionStats attribute), 98
- mana (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- mana_per_level (cassiopeia.type.core.staticdata.ChampionStats attribute), 98
- mana_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- mana_regen (cassiopeia.type.core.staticdata.ChampionStats attribute), 98
- mana_regen (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- mana_regen_per_level (cassiopeia.type.core.staticdata.ChampionStats attribute), 98
- mana_regen_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 105
- map (cassiopeia.type.core.currentgame.Game attribute), 61
- map (cassiopeia.type.core.featuredgames.Game attribute), 64
- map (cassiopeia.type.core.game.Game attribute), 66
- map (cassiopeia.type.core.match.Match attribute), 80
- map (cassiopeia.type.core.staticdata.MapDetails attribute), 109
- map (cassiopeia.type.core.staticdata.RecommendedItems attribute), 112
- map (cassiopeia.type.core.team.MatchSummary attribute), 132
- map (cassiopeia.type.core.tournament.TournamentCode attribute), 136
- Map (class in cassiopeia.type.core.common), 54
- map_id (cassiopeia.type.core.staticdata.MapDetails attribute), 109
- MapData (class in cassiopeia.type.dto.staticdata), 173
- MapDetails (class in cassiopeia.type.core.staticdata), 109
- MapDetails (class in cassiopeia.type.dto.staticdata), 174
- maps (cassiopeia.type.core.staticdata.Item attribute), 101
- MapType (class in cassiopeia.type.core.tournament), 136
- master (cassiopeia.type.core.common.Tier attribute), 59
- masteries (cassiopeia.type.core.currentgame.Participant attribute), 62

- masteries (cassiopeia.type.core.match.Participant attribute), 81
- masteries (cassiopeia.type.core.summoner.MasteryPage attribute), 129
- Mastery (class in cassiopeia.type.core.staticdata), 109
- Mastery (class in cassiopeia.type.dto.currentgame), 151
- Mastery (class in cassiopeia.type.dto.match), 160
- Mastery (class in cassiopeia.type.dto.staticdata), 174
- Mastery (class in cassiopeia.type.dto.summoner), 183
- mastery_ids (cassiopeia.type.dto.currentgame.CurrentGameInfo attribute), 151
- mastery_ids (cassiopeia.type.dto.match.MatchDetail attribute), 161
- mastery_ids (cassiopeia.type.dto.staticdata.Mastery attribute), 175
- mastery_ids (cassiopeia.type.dto.summoner.MasteryPage attribute), 184
- mastery_ids (cassiopeia.type.dto.summoner.MasteryPages attribute), 184
- mastery_level() (cassiopeia.type.core.staticdata.Champion method), 95
- mastery_pages() (cassiopeia.type.core.summoner.Summoner method), 130
- MasteryList (class in cassiopeia.type.dto.staticdata), 175
- MasteryPage (class in cassiopeia.type.core.summoner), 128
- MasteryPage (class in cassiopeia.type.dto.summoner), 183
- MasteryPages (class in cassiopeia.type.dto.summoner), 184
- MasteryTree (class in cassiopeia.type.dto.staticdata), 175
- MasteryTreeItem (class in cassiopeia.type.dto.staticdata), 175
- MasteryTreeList (class in cassiopeia.type.dto.staticdata), 175
- MasteryType (class in cassiopeia.type.core.common), 55
- Match (class in cassiopeia.type.core.match), 79
- match() (cassiopeia.type.core.matchlist.MatchReference method), 94
- match() (cassiopeia.type.core.team.MatchSummary method), 132
- match() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 144
- match_history (cassiopeia.type.core.team.Team attribute), 134
- match_history_uri (cassiopeia.type.core.match.Participant attribute), 81
- match_list() (cassiopeia.type.core.summoner.Summoner method), 130
- MatchDetail (class in cassiopeia.type.dto.match), 160
- matched (cassiopeia.type.core.common.GameType attribute), 54
- MatchHistorySummary (class in cassiopeia.type.dto.team), 185
- MatchList (class in cassiopeia.type.dto.matchlist), 166
- MatchReference (class in cassiopeia.type.core.matchlist), 94
- MatchReference (class in cassiopeia.type.dto.matchlist), 166
- MatchSummary (class in cassiopeia.type.core.team), 132
- max_assists (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_combat_score (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_crit (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_deaths (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_game_time (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_killing_spree (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_kills (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_kills_per_session (cassiopeia.type.core.stats.AggregatedStats attribute), 122
- max_node_capture_assists (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- max_node_captures (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- max_node_neutralizations (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- max_node_neutralize_assist (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- max_objective_score (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- max_rank (cassiopeia.type.core.staticdata.Mastery attribute), 110
- max_rank (cassiopeia.type.core.staticdata.Spell attribute), 116
- max_rank (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
- max_score (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- max_spells_cast (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- max_team_score (cassiopeia.type.core.stats.AggregatedStats attribute), 123

max_time_alive (cassiopeia.type.core.stats.AggregatedStats movespeed (cassiopeia.type.core.staticdata.ItemStats attribute), 123
Message (class in cassiopeia.type.core.status), 126
Message (class in cassiopeia.type.dto.status), 182
meta_data (cassiopeia.type.core.staticdata.Item attribute), 101
meta_data (cassiopeia.type.core.staticdata.Rune attribute), 113
meta_data (cassiopeia.type.core.tournament.TournamentCode attribute), 137
metadata (cassiopeia.type.api.store.HasAllStatus attribute), 49
MetaData (class in cassiopeia.type.core.staticdata), 110
MetaData (class in cassiopeia.type.dto.staticdata), 176
mid_lane (cassiopeia.type.core.common.Lane attribute), 54
mid_lane (cassiopeia.type.core.common.LaneType attribute), 54
minion_denies (cassiopeia.type.core.game.Stats attribute), 70
minion_kills (cassiopeia.type.core.game.Stats attribute), 71
minion_kills (cassiopeia.type.core.match.ParticipantFrame attribute), 82
minion_kills (cassiopeia.type.core.match.ParticipantStats attribute), 86
minions_killed (cassiopeia.type.core.stats.AggregatedStats attribute), 123
MiniSeries (class in cassiopeia.type.dto.league), 158
mode (cassiopeia.type.core.currentgame.Game attribute), 61
mode (cassiopeia.type.core.featuredgames.Game attribute), 64
mode (cassiopeia.type.core.game.Game attribute), 66
mode (cassiopeia.type.core.match.Match attribute), 80
mode (cassiopeia.type.core.staticdata.RecommendedItems attribute), 112
mode (cassiopeia.type.core.team.MatchSummary attribute), 132
modes (cassiopeia.type.core.staticdata.SummonerSpell attribute), 118
modify (cassiopeia.type.core.team.Team attribute), 134
modify_date (cassiopeia.type.core.stats.StatsSummary attribute), 125
modify_date (cassiopeia.type.core.summoner.Summoner attribute), 131
monster (cassiopeia.type.core.match.Event attribute), 78
Monster (class in cassiopeia.type.core.common), 55
monster_kills (cassiopeia.type.core.game.Stats attribute), 71
monster_kills (cassiopeia.type.core.match.ParticipantStats attribute), 86
movespeed (cassiopeia.type.core.staticdata.ChampionStats attribute), 98
movespeed_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 106
MultiRateLimiter (class in cassiopeia.type.api.rates), 47
mushroom (cassiopeia.type.core.common.Ward attribute), 60
N
name (cassiopeia.type.core.league.League attribute), 75
name (cassiopeia.type.core.staticdata.Champion attribute), 95
name (cassiopeia.type.core.staticdata.Item attribute), 101
name (cassiopeia.type.core.staticdata.Mastery attribute), 110
name (cassiopeia.type.core.staticdata.Passive attribute), 111
name (cassiopeia.type.core.staticdata.RecommendedItems attribute), 112
name (cassiopeia.type.core.staticdata.Rune attribute), 113
name (cassiopeia.type.core.staticdata.Skin attribute), 114
name (cassiopeia.type.core.staticdata.Spell attribute), 116
name (cassiopeia.type.core.staticdata.SummonerSpell attribute), 119
name (cassiopeia.type.core.status.Service attribute), 126
name (cassiopeia.type.core.status.Shard attribute), 127
name (cassiopeia.type.core.status.ShardStatus attribute), 128
name (cassiopeia.type.core.summoner.MasteryPage attribute), 129
name (cassiopeia.type.core.summoner.RunePage attribute), 129
name (cassiopeia.type.core.summoner.Summoner attribute), 131
name (cassiopeia.type.core.team.Team attribute), 134
name (cassiopeia.type.core.tournament.TournamentCode attribute), 137
nemesis_draft (cassiopeia.type.core.common.Queue attribute), 56
nemesis_draft (cassiopeia.type.core.common.StatSummaryType attribute), 58
nemesis_draft (cassiopeia.type.core.common.SubType attribute), 59
neutral_monster_killed (cassiopeia.type.core.stats.AggregatedStats attribute), 123
nexus (cassiopeia.type.core.common.Turret attribute), 60
nexus_killed (cassiopeia.type.core.game.Stats attribute), 71
nexus_siege (cassiopeia.type.core.common.GameMode attribute), 54
nexus_siege (cassiopeia.type.core.common.Queue attribute), 56

- nexus_siege (cassiopeia.type.core.common.StatSummaryType attribute), 58
- nexus_siege (cassiopeia.type.core.common.SubType attribute), 59
- node_capture_assists (cassiopeia.type.core.game.Stats attribute), 71
- node_capture_assists (cassiopeia.type.core.match.ParticipantStats attribute), 86
- node_captured (cassiopeia.type.core.game.Stats attribute), 71
- node_captures (cassiopeia.type.core.stats.AggregatedStats attribute), 123
- node_neutralization_assists (cassiopeia.type.core.game.Stats attribute), 71
- node_neutralization_assists (cassiopeia.type.core.match.ParticipantStats attribute), 86
- node_neutralizations (cassiopeia.type.core.game.Stats attribute), 71
- node_neutralizations (cassiopeia.type.core.match.ParticipantStats attribute), 86
- node_neutralizations (cassiopeia.type.core.stats.AggregatedStats attribute), 124
- nodes_captured (cassiopeia.type.core.match.ParticipantStats attribute), 87
- none (cassiopeia.type.core.common.Role attribute), 57
- none (cassiopeia.type.core.tournament.SpectatorType attribute), 136
- normal (cassiopeia.type.core.common.LevelUp attribute), 54
- normal_blind_fives (cassiopeia.type.core.common.Queue attribute), 56
- normal_blind_threes (cassiopeia.type.core.common.Queue attribute), 56
- normal_draft_fives (cassiopeia.type.core.common.Queue attribute), 56
- normal_fives (cassiopeia.type.core.common.StatSummaryType attribute), 58
- normal_fives (cassiopeia.type.core.common.SubType attribute), 59
- normal_games (cassiopeia.type.core.stats.AggregatedStats attribute), 124
- normal_threes (cassiopeia.type.core.common.StatSummaryType attribute), 58
- normal_threes (cassiopeia.type.core.common.SubType attribute), 59
- north_america (cassiopeia.type.core.common.Platform attribute), 55
- north_america (cassiopeia.type.core.common.Region attribute), 57
- north_america (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
- notlike() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 144
- notin() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 144
- notlike() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 145
- npc (cassiopeia.type.core.common.Ascended attribute), 53
- nullsfirst() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 145
- nullslast() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 145
- number (cassiopeia.type.core.staticdata.Skin attribute), 114
- ## O
- objective_score (cassiopeia.type.core.game.Stats attribute), 71
- objective_score (cassiopeia.type.core.match.ParticipantStats attribute), 87
- objectives (cassiopeia.type.core.game.Stats attribute), 71
- Observer (class in cassiopeia.type.dto.currentgame), 151
- Observer (class in cassiopeia.type.dto.featuredgames), 153
- observer_token (cassiopeia.type.core.currentgame.Game attribute), 61
- observer_token (cassiopeia.type.core.featuredgames.Game attribute), 64
- oceania (cassiopeia.type.core.common.Platform attribute), 55
- oceania (cassiopeia.type.core.common.Region attribute), 57
- oceania (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
- one (cassiopeia.type.core.common.Division attribute), 53
- one_for_all (cassiopeia.type.core.common.GameMode attribute), 54
- one_for_all (cassiopeia.type.core.common.Queue attribute), 56
- one_for_all (cassiopeia.type.core.common.StatSummaryType attribute), 58
- one_for_all (cassiopeia.type.core.common.SubType attribute), 59
- one_for_all_mirror (cassiopeia.type.core.common.Queue attribute), 56
- op() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 145
- operate() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 145
- opponent (cassiopeia.type.core.team.MatchSummary attribute), 133

opponent_kills (cassiopeia.type.core.team.MatchSummary attribute), 133

outer (cassiopeia.type.core.common.Turret attribute), 60

P

participant (cassiopeia.type.core.match.Event attribute), 78

participant (cassiopeia.type.core.match.ParticipantFrame attribute), 82

Participant (class in cassiopeia.type.core.currentgame), 62

Participant (class in cassiopeia.type.core.featuredgames), 64

Participant (class in cassiopeia.type.core.game), 66

Participant (class in cassiopeia.type.core.match), 81

Participant (class in cassiopeia.type.dto.featuredgames), 153

Participant (class in cassiopeia.type.dto.match), 161

participant_entry (cassiopeia.type.core.league.League attribute), 75

participant_frames (cassiopeia.type.core.match.Frame attribute), 79

ParticipantFrame (class in cassiopeia.type.core.match), 82

ParticipantFrame (class in cassiopeia.type.dto.match), 162

ParticipantIdentity (class in cassiopeia.type.dto.match), 162

participants (cassiopeia.type.core.currentgame.Game attribute), 61

participants (cassiopeia.type.core.featuredgames.Game attribute), 64

participants (cassiopeia.type.core.game.Game attribute), 66

participants (cassiopeia.type.core.match.Match attribute), 80

participants (cassiopeia.type.core.match.Team attribute), 93

participants (cassiopeia.type.core.tournament.TournamentCode attribute), 137

ParticipantStats (class in cassiopeia.type.core.match), 83

ParticipantStats (class in cassiopeia.type.dto.match), 162

ParticipantTimeline (class in cassiopeia.type.core.match), 88

ParticipantTimeline (class in cassiopeia.type.dto.match), 162

ParticipantTimelineData (class in cassiopeia.type.core.match), 91

ParticipantTimelineData (class in cassiopeia.type.dto.match), 163

passive (cassiopeia.type.core.staticdata.Champion attribute), 96

Passive (class in cassiopeia.type.core.staticdata), 110

Passive (class in cassiopeia.type.dto.staticdata), 176

password (cassiopeia.type.core.tournament.TournamentCode attribute), 137

pbe (cassiopeia.type.core.common.Region attribute), 57

pbe (cassiopeia.type.core.tournament.TournamentRegion attribute), 138

penta_kills (cassiopeia.type.core.game.Stats attribute), 71

penta_kills (cassiopeia.type.core.match.ParticipantStats attribute), 87

penta_kills (cassiopeia.type.core.stats.AggregatedStats attribute), 124

percent_ability_power (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_armor (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_armor_penetration (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_armor_penetration_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_attack_damage (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_attack_speed (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_attack_speed_per_level (cassiopeia.type.core.staticdata.ChampionStats attribute), 98

percent_attack_speed_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_base_attack_damage (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_base_health_regen (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_base_mana_regen (cassiopeia.type.core.staticdata.ItemStats attribute), 106

percent_block (cassiopeia.type.core.staticdata.ItemStats attribute), 107

percent_bonus_armor_penetration (cassiopeia.type.core.staticdata.ItemStats attribute), 107

percent_bonus_health (cassiopeia.type.core.staticdata.ItemStats attribute), 107

percent_critical_strike_damage (cassiopeia.type.core.staticdata.ItemStats attribute), 107

- percent_health (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_health_regen (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_magic_pen_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_magic_penetration (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_magic_resist (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_mana (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_mana_regen (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_movespeed (cassiopeia.type.core.staticdata.ItemStats attribute), 107
- percent_movespeed_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- percent_time_dead (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- percent_time_dead_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- percent_xp_bonus (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- physical (cassiopeia.type.core.staticdata.ChampionInfo attribute), 97
- physical_damage_dealt (cassiopeia.type.core.game.Stats attribute), 71
- physical_damage_dealt (cassiopeia.type.core.match.ParticipantStats attribute), 87
- physical_damage_dealt (cassiopeia.type.core.stats.AggregatedStats attribute), 124
- physical_damage_dealt_to_champions (cassiopeia.type.core.game.Stats attribute), 72
- physical_damage_dealt_to_champions (cassiopeia.type.core.match.ParticipantStats attribute), 87
- physical_damage_taken (cassiopeia.type.core.game.Stats attribute), 72
- physical_damage_taken (cassiopeia.type.core.match.ParticipantStats attribute), 87
- pick_turn (cassiopeia.type.core.currentgame.Ban attribute), 60
- pick_turn (cassiopeia.type.core.featuredgames.Ban attribute), 63
- pick_turn (cassiopeia.type.core.match.Ban attribute), 77
- pick_type (cassiopeia.type.core.tournament.TournamentCode attribute), 137
- PickType (class in cassiopeia.type.core.tournament), 136
- platform (cassiopeia.type.core.currentgame.Game attribute), 61
- platform (cassiopeia.type.core.featuredgames.Game attribute), 64
- platform (cassiopeia.type.core.match.Match attribute), 80
- platform (cassiopeia.type.core.matchlist.MatchReference attribute), 94
- platform (cassiopeia.type.core.status.Shard attribute), 127
- platform (cassiopeia.type.core.status.ShardStatus attribute), 128
- Platform (class in cassiopeia.type.core.common), 55
- platinum (cassiopeia.type.core.common.Tier attribute), 60
- player (cassiopeia.type.core.common.Ascended attribute), 53
- Player (class in cassiopeia.type.dto.game), 154
- Player (class in cassiopeia.type.dto.match), 165
- PlayerStatsSummary (class in cassiopeia.type.dto.stats), 181
- PlayerStatsSummaryList (class in cassiopeia.type.dto.stats), 181
- Point (class in cassiopeia.type.core.common), 55
- point_capture (cassiopeia.type.core.common.EventType attribute), 53
- point_captured (cassiopeia.type.core.match.Event attribute), 78
- points (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52
- points_since_last_level (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52
- points_until_next_level (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52
- poro_king (cassiopeia.type.core.common.GameMode attribute), 54
- poro_king (cassiopeia.type.core.common.Queue attribute), 56
- poro_king (cassiopeia.type.core.common.StatSummaryType attribute), 58
- poro_king (cassiopeia.type.core.common.SubType attribute), 59
- position (cassiopeia.type.core.match.Event attribute), 78
- position (cassiopeia.type.core.match.ParticipantFrame attribute), 82
- Position (class in cassiopeia.type.core.match), 91

Position (class in cassiopeia.type.dto.match), 165
post() (in module cassiopeia.dto.requests), 42
prerequisite (cassiopeia.type.core.staticdata.Mastery attribute), 110
preseason_3 (cassiopeia.type.core.common.Season attribute), 57
preseason_4 (cassiopeia.type.core.common.Season attribute), 57
preseason_5 (cassiopeia.type.core.common.Season attribute), 57
preseason_6 (cassiopeia.type.core.common.Season attribute), 58
preseason_7 (cassiopeia.type.core.common.Season attribute), 58
previous_season_tier (cassiopeia.type.core.match.Participant attribute), 81
print_calls() (in module cassiopeia.baseriotapi), 26
print_calls() (in module cassiopeia.riotapi), 17
priority (cassiopeia.type.core.staticdata.RecommendedItems attribute), 112
process_bind_param() (cassiopeia.type.dto.common.JSONEncoded method), 149
process_literal_param() (cassiopeia.type.dto.common.JSONEncoded method), 149
process_result_value() (cassiopeia.type.dto.common.JSONEncoded method), 149
profile_icon_id (cassiopeia.type.core.currentgame.Participant attribute), 62
profile_icon_id (cassiopeia.type.core.featuredgames.Participant attribute), 64
profile_icon_id (cassiopeia.type.core.summoner.Summoner attribute), 131
profile_icon_id_max (cassiopeia.type.core.staticdata.Realm attribute), 112
progress (cassiopeia.type.core.league.Series attribute), 76
provider_id (cassiopeia.type.core.tournament.TournamentCode attribute), 137
ProviderRegistrationParameters (class in cassiopeia.type.dto.tournament), 187
purchasable (cassiopeia.type.core.staticdata.Gold attribute), 99
put() (in module cassiopeia.dto.requests), 42
python_type (cassiopeia.type.dto.common.JSONEncoded attribute), 149

Q

q_casts (cassiopeia.type.core.game.Stats attribute), 72
quadra_kills (cassiopeia.type.core.game.Stats attribute), 72

quadra_kills (cassiopeia.type.core.match.ParticipantStats attribute), 87
quadra_kills (cassiopeia.type.core.stats.AggregatedStats attribute), 124
quarry (cassiopeia.type.core.common.Point attribute), 56
queue (cassiopeia.type.core.currentgame.Game attribute), 62
queue (cassiopeia.type.core.featuredgames.Game attribute), 64
queue (cassiopeia.type.core.league.League attribute), 75
queue (cassiopeia.type.core.match.Match attribute), 80
queue (cassiopeia.type.core.matchlist.MatchReference attribute), 94
queue (cassiopeia.type.core.team.Stats attribute), 133
Queue (class in cassiopeia.type.core.common), 56

R

r_casts (cassiopeia.type.core.game.Stats attribute), 72
random (cassiopeia.type.core.tournament.PickType attribute), 136
random_urf (cassiopeia.type.core.common.Queue attribute), 57
range (cassiopeia.type.core.staticdata.Spell attribute), 116
range (cassiopeia.type.core.staticdata.SummonerSpell attribute), 119
range_burn (cassiopeia.type.core.staticdata.Spell attribute), 116
range_burn (cassiopeia.type.core.staticdata.SummonerSpell attribute), 119
ranked_dynamic_queue (cassiopeia.type.core.common.Queue attribute), 57
ranked_enabled (cassiopeia.type.core.champion.ChampionStatus attribute), 52
ranked_fives (cassiopeia.type.core.common.Queue attribute), 57
ranked_fives (cassiopeia.type.core.common.StatSummaryType attribute), 58
ranked_fives (cassiopeia.type.core.common.SubType attribute), 59
ranked_premade_fives (cassiopeia.type.core.common.Queue attribute), 57
ranked_premade_fives (cassiopeia.type.core.common.StatSummaryType attribute), 58
ranked_premade_games (cassiopeia.type.core.stats.AggregatedStats attribute), 124
ranked_premade_threes (cassiopeia.type.core.common.Queue attribute), 57
ranked_premade_threes (cassiopeia.type.core.common.StatSummaryType attribute), 58

- attribute), 58
- ranked_solo (cassiopeia.type.core.common.Queue attribute), 57
- ranked_solo (cassiopeia.type.core.common.StatSummaryType attribute), 58
- ranked_solo (cassiopeia.type.core.common.SubType attribute), 59
- ranked_solo_games (cassiopeia.type.core.stats.AggregatedStats attribute), 124
- ranked_stats() (cassiopeia.type.core.summoner.Summoner method), 131
- ranked_threes (cassiopeia.type.core.common.Queue attribute), 57
- ranked_threes (cassiopeia.type.core.common.StatSummaryType attribute), 58
- ranked_threes (cassiopeia.type.core.common.SubType attribute), 59
- RankedStats (class in cassiopeia.type.dto.stats), 181
- ranks_with (cassiopeia.type.core.staticdata.SpellVariables attribute), 117
- RawStats (class in cassiopeia.type.dto.game), 154
- Realm (class in cassiopeia.type.core.staticdata), 111
- Realm (class in cassiopeia.type.dto.staticdata), 176
- rec_math (cassiopeia.type.core.staticdata.ItemSet attribute), 102
- recent_games() (cassiopeia.type.core.summoner.Summoner method), 131
- RecentGames (class in cassiopeia.type.dto.game), 157
- Recommended (class in cassiopeia.type.dto.staticdata), 176
- recommended_items (cassiopeia.type.core.staticdata.Champion attribute), 96
- RecommendedItems (class in cassiopeia.type.core.staticdata), 112
- red (cassiopeia.type.core.common.Monster attribute), 55
- red (cassiopeia.type.core.common.Side attribute), 58
- red_team (cassiopeia.type.core.match.Match attribute), 80
- refinery (cassiopeia.type.core.common.Point attribute), 56
- region (cassiopeia.type.core.match.Match attribute), 80
- region (cassiopeia.type.core.status.Shard attribute), 127
- region (cassiopeia.type.core.status.ShardStatus attribute), 128
- region (cassiopeia.type.core.tournament.TournamentCode attribute), 137
- Region (class in cassiopeia.type.core.common), 57
- required_champion (cassiopeia.type.core.staticdata.Item attribute), 101
- reset_in() (cassiopeia.type.api.rates.MultiRateLimiter method), 47
- reset_in() (cassiopeia.type.api.rates.SingleRateLimiter method), 47
- resolve (cassiopeia.type.core.common.MasteryType attribute), 55
- resource (cassiopeia.type.core.staticdata.Champion attribute), 96
- resource (cassiopeia.type.core.staticdata.Spell attribute), 116
- resource (cassiopeia.type.core.staticdata.SummonerSpell attribute), 119
- result_processor() (cassiopeia.type.dto.common.JSONEncoded method), 149
- reverse_operate() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 146
- rift_herald (cassiopeia.type.core.common.Monster attribute), 55
- rift_herald_kills (cassiopeia.type.core.match.Team attribute), 93
- role (cassiopeia.type.core.game.Stats attribute), 72
- role (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- role (cassiopeia.type.core.matchlist.MatchReference attribute), 94
- Role (class in cassiopeia.type.core.common), 57
- roster (cassiopeia.type.core.team.Team attribute), 134
- Roster (class in cassiopeia.type.dto.team), 185
- rune (cassiopeia.type.core.staticdata.MetaData attribute), 110
- Rune (class in cassiopeia.type.core.staticdata), 113
- Rune (class in cassiopeia.type.dto.currentgame), 152
- Rune (class in cassiopeia.type.dto.match), 166
- Rune (class in cassiopeia.type.dto.staticdata), 177
- rune_ids (cassiopeia.type.dto.currentgame.CurrentGameInfo attribute), 151
- rune_ids (cassiopeia.type.dto.match.MatchDetail attribute), 161
- rune_ids (cassiopeia.type.dto.summoner.RunePage attribute), 184
- rune_ids (cassiopeia.type.dto.summoner.RunePages attribute), 184
- rune_pages() (cassiopeia.type.core.summoner.Summoner method), 131
- rune_type (cassiopeia.type.core.staticdata.Rune attribute), 113
- RuneList (class in cassiopeia.type.dto.staticdata), 177
- RunePage (class in cassiopeia.type.core.summoner), 129
- RunePage (class in cassiopeia.type.dto.summoner), 184
- RunePages (class in cassiopeia.type.dto.summoner), 184
- runes (cassiopeia.type.core.currentgame.Participant attribute), 62
- runes (cassiopeia.type.core.match.Participant attribute), 81

- runes (cassiopeia.type.core.summoner.RunePage attribute), 129
- RuneSlot (class in cassiopeia.type.dto.summoner), 184
- russia (cassiopeia.type.core.common.Platform attribute), 55
- russia (cassiopeia.type.core.common.Region attribute), 57
- russia (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
- ## S
- sanitized_description (cassiopeia.type.core.staticdata.Item attribute), 102
- sanitized_description (cassiopeia.type.core.staticdata.Passive attribute), 111
- sanitized_description (cassiopeia.type.core.staticdata.Rune attribute), 113
- sanitized_description (cassiopeia.type.core.staticdata.Spell attribute), 116
- sanitized_description (cassiopeia.type.core.staticdata.SummonerSpell attribute), 119
- sanitized_descriptions (cassiopeia.type.core.staticdata.Mastery attribute), 110
- sanitized_tooltip (cassiopeia.type.core.staticdata.Spell attribute), 116
- sanitized_tooltip (cassiopeia.type.core.staticdata.SummonerSpell attribute), 119
- sanitized_tooltip_for_level() (cassiopeia.type.core.staticdata.Spell method), 116
- sanitized_tooltip_for_level() (cassiopeia.type.core.staticdata.SummonerSpell method), 119
- score (cassiopeia.type.core.game.Stats attribute), 72
- score (cassiopeia.type.core.match.ParticipantFrame attribute), 83
- score (cassiopeia.type.core.match.ParticipantStats attribute), 87
- score_rank (cassiopeia.type.core.game.Stats attribute), 72
- score_rank (cassiopeia.type.core.match.ParticipantStats attribute), 87
- season (cassiopeia.type.core.match.Match attribute), 80
- season (cassiopeia.type.core.matchlist.MatchReference attribute), 94
- Season (class in cassiopeia.type.core.common), 57
- season_3 (cassiopeia.type.core.common.Season attribute), 58
- season_4 (cassiopeia.type.core.common.Season attribute), 58
- season_5 (cassiopeia.type.core.common.Season attribute), 58
- season_6 (cassiopeia.type.core.common.Season attribute), 58
- season_7 (cassiopeia.type.core.common.Season attribute), 58
- second_to_last_join (cassiopeia.type.core.team.Team attribute), 134
- sell (cassiopeia.type.core.staticdata.Gold attribute), 99
- series (cassiopeia.type.core.league.Entry attribute), 74
- Series (class in cassiopeia.type.core.league), 76
- Service (class in cassiopeia.type.core.status), 126
- Service (class in cassiopeia.type.dto.status), 182
- services (cassiopeia.type.core.status.ShardStatus attribute), 128
- set_api_key() (in module cassiopeia.baseriotapi), 26
- set_api_key() (in module cassiopeia.riotapi), 17
- set_data_store() (in module cassiopeia.riotapi), 17
- set_load_policy() (in module cassiopeia.riotapi), 18
- set_locale() (in module cassiopeia.baseriotapi), 26
- set_locale() (in module cassiopeia.riotapi), 18
- set_proxy() (in module cassiopeia.baseriotapi), 26
- set_proxy() (in module cassiopeia.riotapi), 18
- set_rate_limit() (in module cassiopeia.baseriotapi), 26
- set_rate_limit() (in module cassiopeia.riotapi), 18
- set_rate_limits() (in module cassiopeia.baseriotapi), 27
- set_rate_limits() (in module cassiopeia.riotapi), 18
- set_region() (in module cassiopeia.baseriotapi), 27
- set_region() (in module cassiopeia.riotapi), 18
- set_tournament_api_key() (in module cassiopeia.baseriotapi), 27
- set_tournament_api_key() (in module cassiopeia.riotapi), 18
- set_tournament_rate_limit() (in module cassiopeia.baseriotapi), 27
- set_tournament_rate_limit() (in module cassiopeia.riotapi), 18
- set_tournament_rate_limits() (in module cassiopeia.baseriotapi), 27
- set_tournament_rate_limits() (in module cassiopeia.riotapi), 18
- SetItem (class in cassiopeia.type.core.staticdata), 114
- severity (cassiopeia.type.core.status.Message attribute), 126
- Shard (class in cassiopeia.type.core.status), 127
- Shard (class in cassiopeia.type.dto.status), 182
- ShardStatus (class in cassiopeia.type.core.status), 127
- ShardStatus (class in cassiopeia.type.dto.status), 183
- should_evaluate_none (cassiopeia.type.dto.common.JSONEncoded attribute), 149

- showdown (cassiopeia.type.core.common.GameMode attribute), 54
- showdown_duo (cassiopeia.type.core.common.Queue attribute), 57
- showdown_duo (cassiopeia.type.core.common.StatSummaryType attribute), 58
- showdown_duo (cassiopeia.type.core.common.SubType attribute), 59
- showdown_solo (cassiopeia.type.core.common.Queue attribute), 57
- showdown_solo (cassiopeia.type.core.common.StatSummaryType attribute), 59
- showdown_solo (cassiopeia.type.core.common.SubType attribute), 59
- side (cassiopeia.type.core.currentgame.Ban attribute), 61
- side (cassiopeia.type.core.currentgame.Participant attribute), 62
- side (cassiopeia.type.core.featuredgames.Ban attribute), 63
- side (cassiopeia.type.core.featuredgames.Participant attribute), 65
- side (cassiopeia.type.core.game.Game attribute), 66
- side (cassiopeia.type.core.game.Participant attribute), 67
- side (cassiopeia.type.core.game.Stats attribute), 72
- side (cassiopeia.type.core.match.Event attribute), 78
- side (cassiopeia.type.core.match.Participant attribute), 81
- side (cassiopeia.type.core.match.Team attribute), 93
- Side (class in cassiopeia.type.core.common), 58
- sight (cassiopeia.type.core.common.Ward attribute), 60
- sight_wards_bought (cassiopeia.type.core.game.Stats attribute), 72
- sight_wards_bought (cassiopeia.type.core.match.ParticipantStats attribute), 87
- silver (cassiopeia.type.core.common.Tier attribute), 60
- SingleRateLimiter (class in cassiopeia.type.api.rates), 47
- skill_slot (cassiopeia.type.core.match.Event attribute), 78
- skill_up (cassiopeia.type.core.common.EventType attribute), 53
- Skin (class in cassiopeia.type.core.staticdata), 114
- Skin (class in cassiopeia.type.dto.staticdata), 177
- skins (cassiopeia.type.core.staticdata.Champion attribute), 96
- slug (cassiopeia.type.core.status.Service attribute), 127
- solo (cassiopeia.type.core.common.Role attribute), 57
- special_recipe (cassiopeia.type.core.staticdata.Item attribute), 102
- spectator_type (cassiopeia.type.core.tournament.Tournament attribute), 137
- SpectatorType (class in cassiopeia.type.core.tournament), 136
- Spell (class in cassiopeia.type.core.staticdata), 114
- spell_vamp (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- spells (cassiopeia.type.core.staticdata.Champion attribute), 96
- SpellVariables (class in cassiopeia.type.core.staticdata), 117
- SpellVars (class in cassiopeia.type.dto.staticdata), 177
- spider (cassiopeia.type.core.common.Monster attribute), 55
- spider_assists_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- spider_kills_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
- splash (cassiopeia.type.core.staticdata.Skin attribute), 114
- sprite (cassiopeia.type.core.staticdata.Image attribute), 99
- SQLAlchemyDB (class in cassiopeia.type.api.store), 49
- SQLAlchemyDB.Iterator (class in cassiopeia.type.api.store), 49
- stacks (cassiopeia.type.core.staticdata.Item attribute), 102
- startswith() (cassiopeia.type.dto.common.JSONEncoded.Comparator method), 146
- stat (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- stats (cassiopeia.type.core.game.Game attribute), 66
- stats (cassiopeia.type.core.match.Participant attribute), 81
- stats (cassiopeia.type.core.staticdata.Champion attribute), 96
- stats (cassiopeia.type.core.staticdata.Item attribute), 102
- stats (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- stats (cassiopeia.type.core.staticdata.Rune attribute), 113
- stats (cassiopeia.type.core.stats.StatsSummary attribute), 125
- stats (cassiopeia.type.core.team.Team attribute), 134
- Stats (class in cassiopeia.type.core.game), 67
- Stats (class in cassiopeia.type.core.team), 133
- Stats (class in cassiopeia.type.dto.staticdata), 178
- stats() (cassiopeia.type.core.summoner.Summoner method), 131
- StatsSummary (class in cassiopeia.type.core.stats), 125
- StatSummaryType (class in cassiopeia.type.core.common), 58
- status (cassiopeia.type.core.status.Service attribute), 127
- status (cassiopeia.type.core.team.Team attribute), 135
- status (cassiopeia.type.core.team.TeamMember attribute), 135
- status() (cassiopeia.type.core.staticdata.ChampionCode method), 96
- store (cassiopeia.type.core.staticdata.Realm attribute), 112
- store() (cassiopeia.type.api.store.Cache method), 48
- store() (cassiopeia.type.api.store.DataStore method), 49
- store() (cassiopeia.type.api.store.SQLAlchemyDB method), 50

store() (cassiopeia.type.api.store.VoidDataStore method), 51

sub_type (cassiopeia.type.core.game.Game attribute), 66

SubType (class in cassiopeia.type.core.common), 59

summoner (cassiopeia.type.core.championmastery.ChampionMastery attribute), 52

summoner (cassiopeia.type.core.currentgame.Participant attribute), 62

summoner (cassiopeia.type.core.game.Game attribute), 66

summoner (cassiopeia.type.core.game.Participant attribute), 67

summoner (cassiopeia.type.core.league.Entry attribute), 74

summoner (cassiopeia.type.core.league.League attribute), 75

summoner (cassiopeia.type.core.match.Participant attribute), 81

summoner (cassiopeia.type.core.team.TeamMember attribute), 135

summoner (cassiopeia.type.core.tournament.LobbyEvent attribute), 135

Summoner (class in cassiopeia.type.core.summoner), 129

Summoner (class in cassiopeia.type.dto.summoner), 185

summoner_id (cassiopeia.type.core.match.Participant attribute), 81

summoner_ids (cassiopeia.type.dto.currentgame.CurrentGameInfo attribute), 151

summoner_ids (cassiopeia.type.dto.game.Game attribute), 154

summoner_ids (cassiopeia.type.dto.game.RecentGames attribute), 157

summoner_ids (cassiopeia.type.dto.league.League attribute), 157

summoner_ids (cassiopeia.type.dto.match.MatchDetail attribute), 161

summoner_ids (cassiopeia.type.dto.team.Team attribute), 186

summoner_level (cassiopeia.type.core.staticdata.SummonerSpell attribute), 119

summoner_name (cassiopeia.type.core.currentgame.Participant attribute), 62

summoner_name (cassiopeia.type.core.featuredgames.Participant attribute), 65

summoner_name (cassiopeia.type.core.league.Entry attribute), 74

summoner_name (cassiopeia.type.core.match.Participant attribute), 81

summoner_spell_d (cassiopeia.type.core.currentgame.Participant attribute), 63

summoner_spell_d (cassiopeia.type.core.featuredgames.Participant attribute), 65

summoner_spell_d (cassiopeia.type.core.game.Game attribute), 66

summoner_spell_d (cassiopeia.type.core.match.Participant attribute), 82

summoner_spell_f (cassiopeia.type.core.currentgame.Participant attribute), 63

summoner_spell_f (cassiopeia.type.core.featuredgames.Participant attribute), 65

summoner_spell_f (cassiopeia.type.core.game.Game attribute), 66

summoner_spell_f (cassiopeia.type.core.match.Participant attribute), 82

summoner_spell_ids (cassiopeia.type.dto.currentgame.CurrentGameInfo attribute), 151

summoner_spell_ids (cassiopeia.type.dto.featuredgames.FeaturedGameInfo attribute), 152

summoner_spell_ids (cassiopeia.type.dto.featuredgames.FeaturedGames attribute), 153

summoner_spell_ids (cassiopeia.type.dto.game.Game attribute), 154

summoner_spell_ids (cassiopeia.type.dto.game.RecentGames attribute), 157

summoner_spell_ids (cassiopeia.type.dto.match.MatchDetail attribute), 161

SummonerIdParams (class in cassiopeia.type.dto.tournament), 187

summoners_rift (cassiopeia.type.core.common.Map attribute), 55

summoners_rift (cassiopeia.type.core.tournament.MapType attribute), 136

summoners_rift_autumn (cassiopeia.type.core.common.Map attribute), 55

summoners_rift_summer (cassiopeia.type.core.common.Map attribute), 55

SummonerSpell (class in cassiopeia.type.core.staticdata), 117

SummonerSpell (class in cassiopeia.type.dto.staticdata), 178

SummonerSpellList (class in cassiopeia.type.dto.staticdata), 178

summoning (cassiopeia.type.core.common.EventType attribute), 53

support (cassiopeia.type.core.common.Role attribute), 57

T

- tag (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- tag (cassiopeia.type.core.team.Team attribute), 135
- tags (cassiopeia.type.core.staticdata.Champion attribute), 96
- tags (cassiopeia.type.core.staticdata.Item attribute), 102
- tags (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- tags (cassiopeia.type.core.staticdata.Rune attribute), 113
- team (cassiopeia.type.core.league.Entry attribute), 75
- team (cassiopeia.type.core.league.League attribute), 76
- Team (class in cassiopeia.type.core.match), 92
- Team (class in cassiopeia.type.core.team), 133
- Team (class in cassiopeia.type.dto.match), 166
- Team (class in cassiopeia.type.dto.team), 185
- team_builder (cassiopeia.type.core.common.Queue attribute), 57
- team_builder (cassiopeia.type.core.common.StatSummaryType attribute), 59
- team_builder (cassiopeia.type.core.common.SubType attribute), 59
- team_ids (cassiopeia.type.dto.league.League attribute), 158
- team_name (cassiopeia.type.core.league.Entry attribute), 75
- team_objectives (cassiopeia.type.core.match.ParticipantStats attribute), 87
- team_score (cassiopeia.type.core.match.ParticipantFrame attribute), 83
- team_size (cassiopeia.type.core.tournament.TournamentCode attribute), 137
- TeamMember (class in cassiopeia.type.core.team), 135
- TeamMemberInfo (class in cassiopeia.type.dto.team), 186
- teams() (cassiopeia.type.core.summoner.Summoner method), 131
- TeamStatDetail (class in cassiopeia.type.dto.team), 186
- ten_to_twenty (cassiopeia.type.core.match.ParticipantTimelineData attribute), 91
- tenacity (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- the_crystal_scar (cassiopeia.type.core.common.Map attribute), 55
- the_proving_grounds (cassiopeia.type.core.common.Map attribute), 55
- third_to_last_join (cassiopeia.type.core.team.Team attribute), 135
- thirty_to_end (cassiopeia.type.core.match.ParticipantTimelineData attribute), 91
- three (cassiopeia.type.core.common.Division attribute), 53
- tier (cassiopeia.type.core.league.League attribute), 76
- tier (cassiopeia.type.core.staticdata.Item attribute), 102
- tier (cassiopeia.type.core.staticdata.Metadata attribute), 110
- Tier (class in cassiopeia.type.core.common), 59
- tier_3_items_bought (cassiopeia.type.core.game.Stats attribute), 72
- time_dead (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- time_dead_per_level (cassiopeia.type.core.staticdata.ItemStats attribute), 108
- time_played (cassiopeia.type.core.game.Stats attribute), 72
- timeline (cassiopeia.type.core.match.Match attribute), 80
- timeline (cassiopeia.type.core.match.Participant attribute), 82
- Timeline (class in cassiopeia.type.core.match), 93
- Timeline (class in cassiopeia.type.dto.match), 166
- timestamp (cassiopeia.type.core.match.Event attribute), 78
- timestamp (cassiopeia.type.core.match.Frame attribute), 79
- timestamp (cassiopeia.type.core.matchlist.MatchReference attribute), 94
- timestamp (cassiopeia.type.core.tournament.LobbyEvent attribute), 136
- timetuple (cassiopeia.type.dto.common.JSONEncoded.Comparator attribute), 146
- title (cassiopeia.type.core.staticdata.Champion attribute), 96
- to_json() (cassiopeia.type.core.champion.ChampionStatus method), 52
- to_json() (cassiopeia.type.core.championmastery.ChampionMastery method), 52
- to_json() (cassiopeia.type.core.common.CassiopeiaObject method), 53
- to_json() (cassiopeia.type.core.currentgame.Ban method), 61
- to_json() (cassiopeia.type.core.currentgame.Game method), 62
- to_json() (cassiopeia.type.core.currentgame.Participant method), 63
- to_json() (cassiopeia.type.core.featuredgames.Ban method), 63
- to_json() (cassiopeia.type.core.featuredgames.Game method), 64
- to_json() (cassiopeia.type.core.featuredgames.Participant method), 65
- to_json() (cassiopeia.type.core.game.Game method), 66
- to_json() (cassiopeia.type.core.game.Participant method), 67
- to_json() (cassiopeia.type.core.game.Stats method), 73
- to_json() (cassiopeia.type.core.league.Entry method), 75
- to_json() (cassiopeia.type.core.league.League method), 76

`to_json()` (`cassiopeia.type.core.league.Series` method), [76](#)
`to_json()` (`cassiopeia.type.core.match.Ban` method), [77](#)
`to_json()` (`cassiopeia.type.core.match.CombinedParticipant` method), [77](#)
`to_json()` (`cassiopeia.type.core.match.Event` method), [78](#)
`to_json()` (`cassiopeia.type.core.match.Frame` method), [79](#)
`to_json()` (`cassiopeia.type.core.match.Match` method), [80](#)
`to_json()` (`cassiopeia.type.core.match.Participant` method), [82](#)
`to_json()` (`cassiopeia.type.core.match.ParticipantFrame` method), [83](#)
`to_json()` (`cassiopeia.type.core.match.ParticipantStats` method), [87](#)
`to_json()` (`cassiopeia.type.core.match.ParticipantTimeline` method), [90](#)
`to_json()` (`cassiopeia.type.core.match.ParticipantTimelineData` method), [91](#)
`to_json()` (`cassiopeia.type.core.match.Position` method), [91](#)
`to_json()` (`cassiopeia.type.core.match.Team` method), [93](#)
`to_json()` (`cassiopeia.type.core.match.Timeline` method), [93](#)
`to_json()` (`cassiopeia.type.core.matchlist.MatchReference` method), [94](#)
`to_json()` (`cassiopeia.type.core.staticdata.Champion` method), [96](#)
`to_json()` (`cassiopeia.type.core.staticdata.ChampionInfo` method), [97](#)
`to_json()` (`cassiopeia.type.core.staticdata.ChampionStats` method), [99](#)
`to_json()` (`cassiopeia.type.core.staticdata.Gold` method), [99](#)
`to_json()` (`cassiopeia.type.core.staticdata.Image` method), [100](#)
`to_json()` (`cassiopeia.type.core.staticdata.Item` method), [102](#)
`to_json()` (`cassiopeia.type.core.staticdata.ItemSet` method), [102](#)
`to_json()` (`cassiopeia.type.core.staticdata.ItemStats` method), [108](#)
`to_json()` (`cassiopeia.type.core.staticdata.LevelTip` method), [109](#)
`to_json()` (`cassiopeia.type.core.staticdata.MapDetails` method), [109](#)
`to_json()` (`cassiopeia.type.core.staticdata.Mastery` method), [110](#)
`to_json()` (`cassiopeia.type.core.staticdata.MetaData` method), [110](#)
`to_json()` (`cassiopeia.type.core.staticdata.Passive` method), [111](#)
`to_json()` (`cassiopeia.type.core.staticdata.Realm` method), [112](#)
`to_json()` (`cassiopeia.type.core.staticdata.RecommendedItems` method), [113](#)
`to_json()` (`cassiopeia.type.core.staticdata.Rune` method), [113](#)
`to_json()` (`cassiopeia.type.core.staticdata.SetItem` method), [114](#)
`to_json()` (`cassiopeia.type.core.staticdata.Skin` method), [114](#)
`to_json()` (`cassiopeia.type.core.staticdata.Spell` method), [116](#)
`to_json()` (`cassiopeia.type.core.staticdata.SpellVariables` method), [117](#)
`to_json()` (`cassiopeia.type.core.staticdata.SummonerSpell` method), [119](#)
`to_json()` (`cassiopeia.type.core.stats.AggregatedStats` method), [124](#)
`to_json()` (`cassiopeia.type.core.stats.StatsSummary` method), [125](#)
`to_json()` (`cassiopeia.type.core.status.Incident` method), [125](#)
`to_json()` (`cassiopeia.type.core.status.Message` method), [126](#)
`to_json()` (`cassiopeia.type.core.status.Service` method), [127](#)
`to_json()` (`cassiopeia.type.core.status.Shard` method), [127](#)
`to_json()` (`cassiopeia.type.core.status.ShardStatus` method), [128](#)
`to_json()` (`cassiopeia.type.core.status.Translation` method), [128](#)
`to_json()` (`cassiopeia.type.core.summoner.MasteryPage` method), [129](#)
`to_json()` (`cassiopeia.type.core.summoner.RunePage` method), [129](#)
`to_json()` (`cassiopeia.type.core.summoner.Summoner` method), [131](#)
`to_json()` (`cassiopeia.type.core.team.MatchSummary` method), [133](#)
`to_json()` (`cassiopeia.type.core.team.Stats` method), [133](#)
`to_json()` (`cassiopeia.type.core.team.Team` method), [135](#)
`to_json()` (`cassiopeia.type.core.team.TeamMember` method), [135](#)
`to_json()` (`cassiopeia.type.core.tournament.LobbyEvent` method), [136](#)
`to_json()` (`cassiopeia.type.core.tournament.TournamentCode` method), [137](#)
`to_json()` (`cassiopeia.type.dto.champion.Champion` method), [138](#)
`to_json()` (`cassiopeia.type.dto.champion.ChampionList` method), [138](#)
`to_json()` (`cassiopeia.type.dto.championmastery.ChampionMastery` method), [139](#)
`to_json()` (`cassiopeia.type.dto.common.CassiopeiaDto` method), [139](#)
`to_json()` (`cassiopeia.type.dto.common.CassiopeiaParametersDto` method), [139](#)

[to_json\(\) \(cassiopeia.type.dto.currentgame.BannedChampion method\), 150](#)
[to_json\(\) \(cassiopeia.type.dto.currentgame.CurrentGameInfo method\), 151](#)
[to_json\(\) \(cassiopeia.type.dto.currentgame.CurrentGameParticipant method\), 151](#)
[to_json\(\) \(cassiopeia.type.dto.currentgame.Mastery method\), 151](#)
[to_json\(\) \(cassiopeia.type.dto.currentgame.Observer method\), 151](#)
[to_json\(\) \(cassiopeia.type.dto.currentgame.Rune method\), 152](#)
[to_json\(\) \(cassiopeia.type.dto.featuredgames.BannedChampion method\), 152](#)
[to_json\(\) \(cassiopeia.type.dto.featuredgames.FeaturedGameInfo method\), 152](#)
[to_json\(\) \(cassiopeia.type.dto.featuredgames.FeaturedGames method\), 153](#)
[to_json\(\) \(cassiopeia.type.dto.featuredgames.Observer method\), 153](#)
[to_json\(\) \(cassiopeia.type.dto.featuredgames.Participant method\), 153](#)
[to_json\(\) \(cassiopeia.type.dto.game.Game method\), 154](#)
[to_json\(\) \(cassiopeia.type.dto.game.Player method\), 154](#)
[to_json\(\) \(cassiopeia.type.dto.game.RawStats method\), 157](#)
[to_json\(\) \(cassiopeia.type.dto.game.RecentGames method\), 157](#)
[to_json\(\) \(cassiopeia.type.dto.league.League method\), 158](#)
[to_json\(\) \(cassiopeia.type.dto.league.LeagueEntry method\), 158](#)
[to_json\(\) \(cassiopeia.type.dto.league.MiniSeries method\), 158](#)
[to_json\(\) \(cassiopeia.type.dto.match.BannedChampion method\), 159](#)
[to_json\(\) \(cassiopeia.type.dto.match.Event method\), 160](#)
[to_json\(\) \(cassiopeia.type.dto.match.Frame method\), 160](#)
[to_json\(\) \(cassiopeia.type.dto.match.Mastery method\), 160](#)
[to_json\(\) \(cassiopeia.type.dto.match.MatchDetail method\), 161](#)
[to_json\(\) \(cassiopeia.type.dto.match.Participant method\), 161](#)
[to_json\(\) \(cassiopeia.type.dto.match.ParticipantFrame method\), 162](#)
[to_json\(\) \(cassiopeia.type.dto.match.ParticipantIdentity method\), 162](#)
[to_json\(\) \(cassiopeia.type.dto.match.ParticipantStats method\), 162](#)
[to_json\(\) \(cassiopeia.type.dto.match.ParticipantTimeline method\), 162](#)
[to_json\(\) \(cassiopeia.type.dto.match.ParticipantTimelineData method\), 165](#)
[to_json\(\) \(cassiopeia.type.dto.match.Player method\), 165](#)
[to_json\(\) \(cassiopeia.type.dto.match.Position method\), 165](#)
[to_json\(\) \(cassiopeia.type.dto.match.Rune method\), 166](#)
[to_json\(\) \(cassiopeia.type.dto.match.Team method\), 166](#)
[to_json\(\) \(cassiopeia.type.dto.match.Timeline method\), 166](#)
[to_json\(\) \(cassiopeia.type.dto.matchlist.MatchList method\), 166](#)
[to_json\(\) \(cassiopeia.type.dto.matchlist.MatchReference method\), 166](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.BasicData method\), 167](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.BasicDataStats method\), 167](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Block method\), 168](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.BlockItem method\), 168](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Champion method\), 168](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.ChampionList method\), 169](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.ChampionSpell method\), 170](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Gold method\), 170](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Group method\), 170](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Image method\), 170](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Info method\), 170](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Item method\), 173](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.ItemList method\), 173](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.ItemTree method\), 173](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.LanguageStrings method\), 173](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.LevelTip method\), 173](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.MapData method\), 174](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.MapDetails method\), 174](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.Mastery method\), 175](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.MasteryList method\), 175](#)
[to_json\(\) \(cassiopeia.type.dto.staticdata.MasteryTree method\), 175](#)

`to_json()` (`cassiopeia.type.dto.staticdata.MasteryTreeItem` method), 175

`to_json()` (`cassiopeia.type.dto.staticdata.MasteryTreeList` method), 175

`to_json()` (`cassiopeia.type.dto.staticdata.MetaData` method), 176

`to_json()` (`cassiopeia.type.dto.staticdata.Passive` method), 176

`to_json()` (`cassiopeia.type.dto.staticdata.Realm` method), 176

`to_json()` (`cassiopeia.type.dto.staticdata.Recommended` method), 177

`to_json()` (`cassiopeia.type.dto.staticdata.Rune` method), 177

`to_json()` (`cassiopeia.type.dto.staticdata.RuneList` method), 177

`to_json()` (`cassiopeia.type.dto.staticdata.Skin` method), 177

`to_json()` (`cassiopeia.type.dto.staticdata.SpellVars` method), 178

`to_json()` (`cassiopeia.type.dto.staticdata.Stats` method), 178

`to_json()` (`cassiopeia.type.dto.staticdata.SummonerSpell` method), 178

`to_json()` (`cassiopeia.type.dto.staticdata.SummonerSpellList` method), 179

`to_json()` (`cassiopeia.type.dto.stats.AggregatedStats` method), 181

`to_json()` (`cassiopeia.type.dto.stats.ChampionStats` method), 181

`to_json()` (`cassiopeia.type.dto.stats.PlayerStatsSummary` method), 181

`to_json()` (`cassiopeia.type.dto.stats.PlayerStatsSummaryList` method), 181

`to_json()` (`cassiopeia.type.dto.stats.RankedStats` method), 182

`to_json()` (`cassiopeia.type.dto.status.Incident` method), 182

`to_json()` (`cassiopeia.type.dto.status.Message` method), 182

`to_json()` (`cassiopeia.type.dto.status.Service` method), 182

`to_json()` (`cassiopeia.type.dto.status.Shard` method), 183

`to_json()` (`cassiopeia.type.dto.status.ShardStatus` method), 183

`to_json()` (`cassiopeia.type.dto.status.Translation` method), 183

`to_json()` (`cassiopeia.type.dto.summoner.Mastery` method), 183

`to_json()` (`cassiopeia.type.dto.summoner.MasteryPage` method), 184

`to_json()` (`cassiopeia.type.dto.summoner.MasteryPages` method), 184

`to_json()` (`cassiopeia.type.dto.summoner.RunePage` method), 184

`to_json()` (`cassiopeia.type.dto.summoner.RunePages` method), 184

`to_json()` (`cassiopeia.type.dto.summoner.RuneSlot` method), 185

`to_json()` (`cassiopeia.type.dto.summoner.Summoner` method), 185

`to_json()` (`cassiopeia.type.dto.team.MatchHistorySummary` method), 185

`to_json()` (`cassiopeia.type.dto.team.Roster` method), 185

`to_json()` (`cassiopeia.type.dto.team.Team` method), 186

`to_json()` (`cassiopeia.type.dto.team.TeamMemberInfo` method), 186

`to_json()` (`cassiopeia.type.dto.team.TeamStatDetail` method), 186

`to_json()` (`cassiopeia.type.dto.tournament.LobbyEvent` method), 187

`to_json()` (`cassiopeia.type.dto.tournament.LobbyEventWrapper` method), 187

`to_json()` (`cassiopeia.type.dto.tournament.ProviderRegistrationParameters` method), 187

`to_json()` (`cassiopeia.type.dto.tournament.SummonerIdParams` method), 187

`to_json()` (`cassiopeia.type.dto.tournament.TournamentCode` method), 188

`to_json()` (`cassiopeia.type.dto.tournament.TournamentCodeParameters` method), 188

`to_json()` (`cassiopeia.type.dto.tournament.TournamentCodeUpdateParameters` method), 189

`to_json()` (`cassiopeia.type.dto.tournament.TournamentRegistrationParameters` method), 189

`tokens_earned` (`cassiopeia.type.core.championmastery.ChampionMastery` attribute), 52

`tooltip` (`cassiopeia.type.core.staticdata.Spell` attribute), 116

`tooltip` (`cassiopeia.type.core.staticdata.SummonerSpell` attribute), 119

`tooltip_for_level()` (`cassiopeia.type.core.staticdata.Spell` method), 116

`tooltip_for_level()` (`cassiopeia.type.core.staticdata.SummonerSpell` method), 119

`top_champion_mastersies()` (`cassiopeia.type.core.summoner.Summoner` method), 131

`top_lane` (`cassiopeia.type.core.common.Lane` attribute), 54

`top_lane` (`cassiopeia.type.core.common.LaneType` attribute), 54

`total` (`cassiopeia.type.core.staticdata.Gold` attribute), 99

`tournament_draft` (`cassiopeia.type.core.tournament.PickType` attribute), 136

tournament_id (cassiopeia.type.core.tournament.TournamentCode attribute), 137
 TournamentCode (class in cassiopeia.type.core.tournament), 136
 TournamentCode (class in cassiopeia.type.dto.tournament), 187
 TournamentCodeParameters (class in cassiopeia.type.dto.tournament), 188
 TournamentCodeUpdateParameters (class in cassiopeia.type.dto.tournament), 188
 TournamentRegion (class in cassiopeia.type.core.tournament), 137
 TournamentRegistrationParameters (class in cassiopeia.type.dto.tournament), 189
 tower (cassiopeia.type.core.match.Event attribute), 78
 Translation (class in cassiopeia.type.core.status), 128
 Translation (class in cassiopeia.type.dto.status), 183
 translations (cassiopeia.type.core.status.Message attribute), 126
 tree (cassiopeia.type.core.staticdata.Mastery attribute), 110
 triple_kills (cassiopeia.type.core.game.Stats attribute), 73
 triple_kills (cassiopeia.type.core.match.ParticipantStats attribute), 88
 triple_kills (cassiopeia.type.core.stats.AggregatedStats attribute), 124
 true_damage_dealt (cassiopeia.type.core.game.Stats attribute), 73
 true_damage_dealt (cassiopeia.type.core.match.ParticipantStats attribute), 88
 true_damage_dealt_to_champions (cassiopeia.type.core.game.Stats attribute), 73
 true_damage_dealt_to_champions (cassiopeia.type.core.match.ParticipantStats attribute), 88
 true_damage_taken (cassiopeia.type.core.game.Stats attribute), 73
 true_damage_taken (cassiopeia.type.core.match.ParticipantStats attribute), 88
 turkey (cassiopeia.type.core.common.Platform attribute), 55
 turkey (cassiopeia.type.core.common.Region attribute), 57
 turkey (cassiopeia.type.core.tournament.TournamentRegion attribute), 138
 turret (cassiopeia.type.core.common.Building attribute), 53
 Turret (class in cassiopeia.type.core.common), 60
 turret_assists_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
 turret_kills (cassiopeia.type.core.game.Stats attribute), 73
 turret_kills (cassiopeia.type.core.match.ParticipantStats attribute), 88
 turret_kills (cassiopeia.type.core.match.Team attribute), 93
 turret_kills_per_min_counts (cassiopeia.type.core.match.ParticipantTimeline attribute), 91
 turret_kills_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), 90
 turrets_killed (cassiopeia.type.core.stats.AggregatedStats attribute), 124
 tutorial (cassiopeia.type.core.common.GameMode attribute), 54
 tutorial (cassiopeia.type.core.common.GameType attribute), 54
 twenty_to_thirty (cassiopeia.type.core.match.ParticipantTimelineData attribute), 91
 twisted_treeline (cassiopeia.type.core.common.Map attribute), 55
 twisted_treeline (cassiopeia.type.core.tournament.MapType attribute), 136
 twisted_treeline_original (cassiopeia.type.core.common.Map attribute), 55
 two (cassiopeia.type.core.common.Division attribute), 53
 type (cassiopeia.type.core.currentgame.Game attribute), 62
 type (cassiopeia.type.core.featuredgames.Game attribute), 64
 type (cassiopeia.type.core.game.Game attribute), 66
 type (cassiopeia.type.core.match.Event attribute), 78
 type (cassiopeia.type.core.match.Match attribute), 80
 type (cassiopeia.type.core.staticdata.ItemSet attribute), 102
 type (cassiopeia.type.core.staticdata.MetaData attribute), 110
 type (cassiopeia.type.core.staticdata.RecommendedItems attribute), 113
 type (cassiopeia.type.core.stats.StatsSummary attribute), 125
 type (cassiopeia.type.core.tournament.LobbyEvent attribute), 136
 type (cassiopeia.type.dto.common.JSONEncoded.Comparator attribute), 146
 type_engine() (cassiopeia.type.dto.common.JSONEncoded method), 149

U

undefined (cassiopeia.type.core.common.Turret attribute), 60
 undefined (cassiopeia.type.core.common.Ward attribute), 60

units_healed (cassiopeia.type.core.game.Stats attribute), [73](#)
units_healed (cassiopeia.type.core.match.ParticipantStats attribute), [88](#)
unpurchasable_items (cassiopeia.type.core.staticdata.MapDetails attribute), [109](#)
unranked (cassiopeia.type.core.common.Tier attribute), [60](#)
unreal_kills (cassiopeia.type.core.game.Stats attribute), [73](#)
unreal_kills (cassiopeia.type.core.match.ParticipantStats attribute), [88](#)
unreal_kills (cassiopeia.type.core.stats.AggregatedStats attribute), [124](#)
update_tournament_code() (in module cassiopeia.baseriotapi), [27](#)
update_tournament_code() (in module cassiopeia.core.tournamentapi), [38](#)
update_tournament_code() (in module cassiopeia.dto.tournamentapi), [46](#)
update_tournament_code() (in module cassiopeia.riotapi), [19](#)
updated (cassiopeia.type.core.status.Message attribute), [126](#)
updated (cassiopeia.type.core.status.Translation attribute), [128](#)
updates (cassiopeia.type.core.status.Incident attribute), [125](#)
upgraded_yellow_trinket (cassiopeia.type.core.common.Ward attribute), [60](#)
urf (cassiopeia.type.core.common.Queue attribute), [57](#)
urf (cassiopeia.type.core.common.StatSummaryType attribute), [59](#)
urf (cassiopeia.type.core.common.SubType attribute), [59](#)

V

variables (cassiopeia.type.core.staticdata.Spell attribute), [117](#)
variables (cassiopeia.type.core.staticdata.SummonerSpell attribute), [119](#)
version (cassiopeia.type.core.match.Match attribute), [80](#)
version (cassiopeia.type.core.staticdata.Realm attribute), [112](#)
veteran (cassiopeia.type.core.league.Entry attribute), [75](#)
victim (cassiopeia.type.core.match.Event attribute), [79](#)
victory_points (cassiopeia.type.core.game.Stats attribute), [73](#)
victory_score (cassiopeia.type.core.match.Team attribute), [93](#)
vilemaw_kills (cassiopeia.type.core.match.Team attribute), [93](#)
vision (cassiopeia.type.core.common.Ward attribute), [60](#)

vision_wards_bought (cassiopeia.type.core.game.Stats attribute), [73](#)
vision_wards_bought (cassiopeia.type.core.match.ParticipantStats attribute), [88](#)
VoidDataStore (class in cassiopeia.type.api.store), [50](#)

W

w_casts (cassiopeia.type.core.game.Stats attribute), [73](#)
wait() (cassiopeia.type.api.rates.MultiRateLimiter method), [47](#)
wait() (cassiopeia.type.api.rates.SingleRateLimiter method), [47](#)
ward (cassiopeia.type.core.match.Event attribute), [79](#)
Ward (class in cassiopeia.type.core.common), [60](#)
ward_kill (cassiopeia.type.core.common.EventType attribute), [53](#)
ward_kills (cassiopeia.type.core.game.Stats attribute), [73](#)
ward_kills (cassiopeia.type.core.match.ParticipantStats attribute), [88](#)
ward_placement (cassiopeia.type.core.common.EventType attribute), [53](#)
wards_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), [91](#)
wards_placed (cassiopeia.type.core.game.Stats attribute), [73](#)
wards_placed (cassiopeia.type.core.match.ParticipantStats attribute), [88](#)
width (cassiopeia.type.core.staticdata.Image attribute), [100](#)
win (cassiopeia.type.core.game.Stats attribute), [74](#)
win (cassiopeia.type.core.match.ParticipantStats attribute), [88](#)
win (cassiopeia.type.core.match.Team attribute), [93](#)
win (cassiopeia.type.core.team.MatchSummary attribute), [133](#)
windmill (cassiopeia.type.core.common.Point attribute), [56](#)
wins (cassiopeia.type.core.league.Entry attribute), [75](#)
wins (cassiopeia.type.core.league.Series attribute), [76](#)
wins (cassiopeia.type.core.stats.AggregatedStats attribute), [124](#)
wins (cassiopeia.type.core.stats.StatsSummary attribute), [125](#)
wins (cassiopeia.type.core.team.Stats attribute), [133](#)
wins_required (cassiopeia.type.core.league.Series attribute), [76](#)
with_traceback() (cassiopeia.type.api.exception.APIError method), [46](#)
with_traceback() (cassiopeia.type.api.exception.CassiopeiaException method), [47](#)
with_variant() (cassiopeia.type.dto.common.JSONEncoded method), [149](#)

X

x (cassiopeia.type.core.match.Position attribute), [91](#)

x (cassiopeia.type.core.staticdata.Image attribute), [100](#)

xp (cassiopeia.type.core.match.ParticipantFrame attribute), [83](#)

xp_bonus (cassiopeia.type.core.staticdata.ItemStats attribute), [108](#)

xp_diff_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), [91](#)

xp_per_min_deltas (cassiopeia.type.core.match.ParticipantTimeline attribute), [91](#)

Y

y (cassiopeia.type.core.match.Position attribute), [92](#)

y (cassiopeia.type.core.staticdata.Image attribute), [100](#)

yellow_trinket (cassiopeia.type.core.common.Ward attribute), [60](#)

Z

zero_to_ten (cassiopeia.type.core.match.ParticipantTimelineData attribute), [91](#)