

---

# Cassiopeia Documentation

*Release 3.0.x*

**Rob Rua**

**Jul 29, 2023**



## CONTENTS

|          |                             |           |
|----------|-----------------------------|-----------|
| <b>1</b> | <b>What is Cassiopeia?</b>  | <b>1</b>  |
| <b>2</b> | <b>Why use Cass?</b>        | <b>3</b>  |
| <b>3</b> | <b>An Example</b>           | <b>5</b>  |
| <b>4</b> | <b>Django web Framework</b> | <b>7</b>  |
| 4.1      | Contributing . . . . .      | 7         |
| 4.2      | Overview . . . . .          | 7         |
| 4.3      | Top Level APIs . . . . .    | 77        |
| <b>5</b> | <b>Index and Search</b>     | <b>79</b> |
|          | <b>Python Module Index</b>  | <b>81</b> |
|          | <b>Index</b>                | <b>83</b> |



---

CHAPTER  
ONE

---

## WHAT IS CASSIOPEIA?

Cassiopeia (which we fondly call Cass) is a framework for pulling and working with data from the [Riot API](#). Cass differentiates itself from other API wrappers by taking a page from one of Cassiopeia's quotes, "I'll take care of everything." Our main goal is to make your life (and ours) as developers *easy*.

Cass is composed of three key pieces:

- 1) An *interface* for pulling data from the Riot API.
- 2) A *type system* of classes for holding and working with the data pulled from Riot.
- 3) *Caches and databases* to temporarily and permanently store that data.

Together, these three pieces provide the user experience we desire. Scroll down for a quick example of how Cass works, what Cass does for you as a user, and information about contributing.



---

CHAPTER  
**TWO**

---

## **WHY USE CASS?**

- An excellent user interface that makes working with data from the Riot API easy and fun.
- “Perfect” rate limiting.
- Guaranteed optimal usage of your API key.
- Built in caching and (coming) the ability to easily hook into a database for offline storage of data.
- Extendability to non-Riot data. Because Cass is a framework and not just an API wrapper, you can integrate your own data sources into your project. Cass already supports Data Dragon and the `champion.gg` API in addition to the Riot API.
- Dynamic settings so you can configure Cass for your specific use case.



---

## CHAPTER THREE

---

### AN EXAMPLE

We will quickly and efficiently look up the champion masteries for the summoner “Kalturi” (one of the developers) and print the champions he is best at. If you just want a quick look at how the interface looks, feel free to just read these three lines and skip the explanation. The explanation explains how the three bullet points above fit together and allow this code to be run.

```
kalturi = Summoner(name="Kalturi")
good_with = kalturi.champion_masteries.filter(lambda cm: cm.level >= 6)
print([cm.champion.name for cm in good_with])

# At the time of writing this, this prints:
["Vel'Koz", 'Blitzcrank', 'Braum', 'Lulu', 'Sejuani']
```

The above three lines are relatively concise code, and if you know what lambdas and list comprehensions are then it will likely be readable. However, there is a deceptive amount of logic in these three lines, so let’s break it down. (If you don’t understand everything immediately, don’t worry, that’s why you’re using Cass. You don’t have to understand how everything works behinds the scenes, you just get to write good code.)

```
kalturi = Summoner(name="Kalturi")
```

First, we create a summoner with a `name` and `id`. Note that creating `kalturi` doesn’t trigger a call to the Riot API – it merely instantiates a `Summoner` object with a name and id.

```
... = kalturi.champion_masteries ...
```

Next we ask for the champion masteries for `kalturi` by running `kalturi.champion_masteries`. This creates an un-instantiated list which will contain champion masteries if any item in it is accessed.

```
good_with = kalturi.champion_masteries.filter(lambda cm: cm.level >= 6)
```

Third, the `.filter` method is called on the list of champion masteries. `filter` is a python built-in that operates on a list and filters the items in it based on some criteria. That criteria is defined by the `lambda` function we pass in.

A lambda is a quick way of defining functions in-line without using the `def` statement. In this case, `lambda cm:` takes in an object and assigns it to the variable `cm`, then it returns `cm.level > 6`. So this lambda will return True for any champion mastery whose mastery level is greater than or equal to 6.

The `.filter(lambda cm: cm.level > 6)` therefore operates on the list of champion masteries. When the list is iterated over, the champion masteries are queried. This requires a summoner id, which is pulled from `kalturi.id`, and the Riot API is queried for Kalturi’s champion masteries. With the champion mastery data pulled, `.filter` then filters the list looking for all champion masteries with mastery level 6 or higher.

```
print([cm.champion.name for cm in good_with])
```

Finally, the third line prints a list of the champion names for those champions.

Together these three lines illustrate the concise user interface that Cass provides, the way in which the data can be used, when the data is pulled (queried).

## DJANGO WEB FRAMEWORK

There is an integration of cassiopeia to the popular python web framework Django made by Mori(Paaksing), this integration is aimed to fix most issues/conflicts related to co-occurrence of cassiopeia and Django. In this integration will give you better tools for building your Django/DRF based app, you will have the ability to use any production tested cache backends that Django's cache framework supports.

**New in v2.0:** A new datastore called *Omnistone* is introduced in response to issue #1 of this repo, this is a refined version of *Cache* that automatically deletes expired objects when *MAX\_ENTRIES* is hit, then culls the datastore according to the *CULL\_FREQUENCY* given. The culling strategy used is the same as Django Cache Framework, which is LRU culling (Least Recently Used).

- Link to *django-cassiopeia* [repository](<https://github.com/paaksing/django-cassiopeia>) (If you love using it, make sure to star!).
- Link to *django-cassiopeia* [documentations](<https://paaksing.github.io/django-cassiopeia/>) (Production Release v2.0).
- If you have any issues or feature requests with *django-cassiopeia*, tag Mori in our discord server, or fire an issue in the repository.

Unfortunately, we currently don't have an integration to Flask and any contribution is welcome.

## 4.1 Contributing

Contributions are welcome and we have an entire *page* devoted to ways in which you can help us with Cass.

## 4.2 Overview

### 4.2.1 Using Cassiopeia

Objects that hold data from the Riot API can be created using two different interfaces. The top-level *cassiopeia* module contains methods to query for objects using method calls, as well as class constructors to create objects directly.

Example usage of the two interfaces:

```
import cassiopeia as cass
kalturi = cass.get_summoner(name="Kalturi", region="NA")

from cassiopeia import Summoner
kalturi = Summoner(name="Kalturi", region="NA")
```

Also note that many types can be pulled from `Summoner` objects. This is the preferred way to interact with these types. They are listed below:

```
from cassiopeia import Summoner
kalturi = Summoner(name="Kalturi", region="NA")
kalturi.champion_masteries
kalturi.match_history
kalturi.current_match
kalturi.leagues
```

## Django web Framework

There is an integration of cassiopeia to the popular python web framework Django made by Mori(Paaksing), this integration is aimed to fix most issues/conflicts related to co-occurrence of cassiopeia and Django. In this integration will give you better tools for building your Django/DRF based app, you will have the ability to use any production tested cache backends that Django's cache framework supports.

**New in v2.0:** A new datastore called *Omnistone* is introduced in response to issue #1 of this repo, this is a refined version of *Cache* that automatically deletes expired objects when *MAX\_ENTRIES* is hit, then culls the datastore according to the *CULL\_FREQUENCY* given. The culling strategy used is the same as Django Cache Framework, which is LRU culling (Least Recently Used).

- Link to *django-cassiopeia* [repository](<https://github.com/paaksing/django-cassiopeia>) (If you love using it, make sure to star!).
- Link to *django-cassiopeia* [documentations](<https://paaksing.github.io/django-cassiopeia/>) (Production Release v2.0).
- If you have any issues or feature requests with *django-cassiopeia*, tag Mori in our discord server, or fire an issue in the repository.

Unfortunately, we currently don't have an integration to Flask and any contribution is welcome.

## Methods and Class Constructors

See the links below for the method and class names for each type.

### Settings

```
class cassiopeia._configuration.settings.Settings(settings)
    Bases: object
    clear_sinks(type: Optional[Type[T]] = None)
    expire_sinks(type: Optional[Type[T]] = None)
    pipeline
    plugins
    set_riot_api_key(key)
    version_from_match
```

## Data and Enums

These data are available as enums (constants) and can be used to interact with many of the objects and methods in Cass.

```
class cassiopeia.data.Continent(value)
```

Bases: [Enum](#)

An enumeration.

```
americas = 'AMERICAS'
```

```
asia = 'ASIA'
```

```
europe = 'EUROPE'
```

```
sea = 'SEA'
```

```
class cassiopeia.data.Division(value)
```

Bases: [Enum](#)

An enumeration.

```
four = 'IV'
```

```
one = 'I'
```

```
three = 'III'
```

```
two = 'II'
```

```
class cassiopeia.data.GameMode(value)
```

Bases: [Enum](#)

An enumeration.

```
all_random_summoners_rift = 'ARSR'
```

```
all_random_urf_snow = 'SNOWURF'
```

```
aram = 'ARAM'
```

```
ascension = 'ASCENSION'
```

```
assassinate = 'ASSASSINATE'
```

```
classic = 'CLASSIC'
```

```
dark_star = 'DARKSTAR'
```

```
dominion = 'ODIN'
```

```
doom_bots = 'DOOMBOTSTEEMO'
```

```
nexus_blitz = 'NEXUSBLITZ'
```

```
nexus_siege = 'SIEGE'
```

```
odyssey = 'ODYSSEY'
```

```
one_for_all = 'ONEFORALL'
```

```
overcharge = 'OVERCHARGE'
poro_king = 'KINGPORO'
practice_tool = 'PRACTICETOOL'
project = 'PROJECT'
showdown = 'FIRSTBLOOD'
star_guardian = 'STARGUARDIAN'
tutorial = 'TUTORIAL'
tutorial_1 = 'TUTORIAL_MODULE_1'
tutorial_2 = 'TUTORIAL_MODULE_2'
tutorial_3 = 'TUTORIAL_MODULE_3'
urf = 'URF'
utlbook = 'ULTBOOK'

class cassiopeia.data.GameType(value)
    Bases: Enum
    An enumeration.

    custom = 'CUSTOM_GAME'
    matched = 'MATCHED_GAME'
    tutorial = 'TUTORIAL_GAME'

class cassiopeia.data.Key(value)
    Bases: Enum
    An enumeration.

    E = 'E'
    Q = 'Q'
    R = 'R'
    W = 'W'

class cassiopeia.data.Lane(value)
    Bases: Enum
    An enumeration.

    bot_lane = 'BOT_LANE'
    from_match_naming_scheme()
    jungle = 'JUNGLE'
    mid_lane = 'MID_LANE'
```

```
top_lane = 'TOP_LANE'
utility = 'UTILITY'

class cassiopeia.data.MasteryTree(value)
Bases: Enum
An enumeration.

cunning = 'Cunning'
ferocity = 'Ferocity'
resolve = 'Resolve'

class cassiopeia.data.MatchType(value)
Bases: Enum
An enumeration.

normal = 'normal'
ranked = 'ranked'
tourney = 'tourney'
tutorial = 'tutorial'

class cassiopeia.data.Platform(value)
Bases: Enum
An enumeration.

brazil = 'BR1'
continent
default_locale
europe_north_east = 'EUN1'
europe_west = 'EUW1'
static from_region(region)
japan = 'JP1'
korea = 'KR'
latin_america_north = 'LA1'
latin_america_south = 'LA2'
north_america = 'NA1'
oceania = 'OC1'
region
russia = 'RU'
```

```
turkey = 'TR1'

class cassiopeia.data.Position(value)
    Bases: Enum

    An enumeration.

    apex = 'APEX'

    bottom = 'BOTTOM'

    from_league_naming_scheme()

    jungle = 'JUNGLE'

    middle = 'MIDDLE'

    none = 'NONE'

    top = 'TOP'

    utility = 'UTILITY'

class cassiopeia.data.Queue(value)
    Bases: Enum

    An enumeration.

    all_random_summoners_rift = 'ARSR_5x5'

    all_random_urf = 'ARURF_5X5'

    all_random_urf_snow = 'SNOWURF'

    aram = 'ARAM'

    aram_butchers_bridge = 'BILGEWATER_ARAM_5x5'

    ascension = 'ASCENSION_5x5'

    black_market_brawlers = 'BILGEWATER_5x5'

    blind_fives = 'NORMAL_5V5_BLIND_PICK'

    blind_threes = 'NORMAL_3X3_BLIND_PICK'

    blood_hunt_assassin = 'ASSASSINATE_5x5'

    clash = 'CLASH'

    coop_ai_beginner_fives = 'BOT_5X5_BEGINNER'

    coop_ai_beginner_threes = 'BOT_3X3_BEGINNER'

    coop_ai_intermediate_fives = 'BOT_5X5_INTERMEDIATE'

    coop_ai_intermediate_threes = 'BOT_3X3_INTERMEDIATE'

    coop_ai_intro_fives = 'BOT_5X5_INTRO'

    coop_ai_intro_threes = 'BOT_3X3_INTRO'
```

```
custom = 'CUSTOM'

dark_star = 'DARKSTAR_3x3'

definitely_not_dominion = 'DEFINITELY_NOT_DOMINION_5x5'

deprecated_all_random_urf = 'ARURF_5X5'

deprecated_aram = 'ARAM_5x5'

deprecated_blind_dominion = 'ODIN_5x5_BLIND'

deprecated_blind_fives = 'NORMAL_5x5_BLIND'

deprecated_blind_threes = 'NORMAL_3x3'

deprecated_coop_ai_beginner_fives = 'BOT_5x5_BEGINNER_DEPRECATED'

deprecated_coop_ai_dominion = 'BOT_ODIN_5x5'

deprecated_coop_ai_fives = 'BOT_5x5'

deprecated_coop_ai_intermediate_fives = 'BOT_5x5_INTERMEDIATE_DEPRECATED'

deprecated_coop_ai_intro_fives = 'BOT_5x5_INTRO_DEPRECATED'

deprecated_coop_ai_threes = 'BOT_TT_3x3'

deprecated_doom_bots_rank_1 = 'NIGHTMARE_BOT_5x5_RANK1'

deprecated_doom_bots_rank_2 = 'NIGHTMARE_BOT_5x5_RANK2'

deprecated_doom_bots_rank_5 = 'NIGHTMARE_BOT_5x5_RANK5'

deprecated_draft_dominion = 'ODIN_5x5_DRAFT'

deprecated_draft_fives = 'NORMAL_5x5_DRAFT'

deprecated_nexus_blitz = 'NEXUS_BLITZ'

deprecated_nexus_siege = 'SIEGE'

deprecated_poro_king = 'KING_PORO_5x5'

deprecated_ranked_fives = 'TEAM_BUILDER_DRAFT_RANKED_5x5'

deprecated_ranked_flex_threes = 'RANKED_FLEX_TT_DEPRECATED'

deprecated_ranked_premade_fives = 'RANKED_PREMADE_5x5'

deprecated_ranked_premade_threes = 'RANKED_PREMADE_3x3'

deprecated_ranked_solo_fives = 'CLASSIC'

deprecated_ranked_team_fives = 'RANKED_TEAM_5x5'

deprecated_ranked_team_threes = 'RANKED_TEAM_3x3'

deprecated_team_builder_fives = 'GROUP_FINDER_5x5'

doom_bots = 'NIGHTMARE_BOT_5X5'
```

```
doom_bots_difficult = 'NIGHTMARE_BOT_5X5_VOTE'

from_id()

guardian_invasion_normal = 'INVASION_NORMAL'

guardian_invasion_onslaught = 'INVASION_ONSLAUGHT'

hexakill_summoners_rift = 'SR_6x6'

hexakill_twisted_treeline = 'HEXAKILL'

id

mirror_mode_fives = 'ONEFORALL_MIRRORMODE_5x5'

nemesis_draft = 'COUNTER_PICK'

nexus_blitz = 'NEXUS_BLITZ'

nexus_siege = 'NEXUS_SIEGE'

normal_draft_fives = 'TEAM_BUILDER_DRAFT_UNRANKED_5x5'

normal_tft = 'NORMAL_TFT'

odyssey_cadet = 'ODYSSEY_CADET'

odyssey_captain = 'ODYSSEY_CAPTAIN'

odyssey_crewmember = 'ODYSSEY_CREWMEMBER'

odyssey_intro = 'ODYSSEY_INTRO'

odyssey_onslaught = 'ODYSSEY_ONSLAUGHT'

one_for_all = 'ONEFORALL_5x5'

one_for_all_rapid = 'ONEFORALL_RAPID_5x5'

overcharge = 'OVERCHARGE'

poro_king = 'KINGPORO'

project = 'PROJECT'

ranked_flex_fives = 'RANKED_FLEX_SR'

ranked_flex_threes = 'RANKED_FLEX_TT'

ranked_solo_fives = 'RANKED_SOLO_5x5'

ranked_tft = 'RANKED_TFT'

ranked_tft_double_up = 'RANKED_TFT_DOUBLE_UP'

ranked_tft_pairs = 'RANKED_TFT_PAIRS'

showdown_1v1 = 'FIRSTBLOOD_1x1'

showdown_2v2 = 'FIRSTBLOOD_2x2'
```

```
tutorial1 = 'TUTORIAL_1'  
tutorial2 = 'TUTORIAL_2'  
tutorial3 = 'TUTORIAL_3'  
ultimate_spellbook = 'ULTIMATE_SPELLBOOK'  
urf = 'URF_5x5'  
urf_coop_ai = 'BOT_URF_5x5'  
class cassiopeia.data.Rank(tier: Tier, division: Division)  
    Bases: object  
class cassiopeia.data.Region(value)  
    Bases: Enum  
    An enumeration.  
    brazil = 'BR'  
    continent  
    default_locale  
    europe_north_east = 'EUNE'  
    europe_west = 'EUW'  
    static from_platform(platform)  
    japan = 'JP'  
    korea = 'KR'  
    latin_america_north = 'LAN'  
    latin_america_south = 'LAS'  
    north_america = 'NA'  
    oceania = 'OCE'  
    platform  
    russia = 'RU'  
    timezone  
    turkey = 'TR'  
class cassiopeia.data.Resource(value)  
    Bases: Enum  
    An enumeration.  
    blood_well = 'Blood Well'  
    courage = 'Courage'
```

```
crimson_rush = 'Crimson Rush'
energy = 'Energy'
ferocity = 'Ferocity'
flow = 'Flow'
fury = 'Fury'
heat = 'Heat'
mana = 'Mana'
no_cost = 'No Cost'
none = 'None'
rage = 'Rage'
shield = 'Shield'

class cassiopeia.data.Role(value)
    Bases: Enum
    An enumeration.

    duo = 'DUO'
    duo_carry = 'DUO_CARRY'
    duo_support = 'DUO_SUPPORT'
    from_match_naming_scheme()
    none = 'NONE'
    solo = 'SOLO'

class cassiopeia.data.Side(value)
    Bases: Enum
    An enumeration.

    blue = 100
    red = 200

class cassiopeia.data.SummonersRiftArea(value)
    Bases: Enum
    An enumeration.

    bot_lane_blue = 'BOT_LANE_BLUE'
    bot_lane_purple = 'BOT_LANE_PURPLE'
    bot_lane_red = 'BOT_LANE_RED'
    static from_position(position: Position) → SummonersRiftArea
```

```
get_lane() → Lane
get_side() → Side

jungle_bot_blue = 'JUNGLE_BOT_BLUE'
jungle_bot_red = 'JUNGLE_BOT_RED'
jungle_top_blue = 'JUNGLE_TOP_BLUE'
jungle_top_red = 'JUNGLE_TOP_RED'
mid_lane_blue = 'MID_LANE_BLUE'
mid_lane_purple = 'MID_LANE_PURPLE'
mid_lane_red = 'MID_LANE_RED'
nexus_blue = 'NEXUS_BLUE'
nexus_red = 'NEXUS_RED'
none = 'NONE'
river_bot = 'RIVER_BOT'
river_top = 'RIVER_TOP'
top_lane_blue = 'TOP_LANE_BLUE'
top_lane_purple = 'TOP_LANE_PURPLE'
top_lane_red = 'TOP_LANE_RED'

class cassiopeia.data.Tier(value)
    Bases: Enum
    An enumeration.

    bronze = 'BRONZE'
    challenger = 'CHALLENGER'
    diamond = 'DIAMOND'
    emerald = 'EMERALD'
    gold = 'GOLD'
    grandmaster = 'GRANDMASTER'
    iron = 'IRON'
    master = 'MASTER'
    platinum = 'PLATINUM'
    silver = 'SILVER'
    unranked = 'UNRANKED'
```

```
class cassiopeia.data.Tower(value)
Bases: Enum
An enumeration.

BASE = 'BASE_TURRET'
INNER = 'INNER_TURRET'
NEXUS = 'NEXUS_TURRET'
OUTER = 'OUTER_TURRET'
UNDEFINED = 'UNDEFINED_TURRET'
```

## Champions

```
cassiopeia.get_champions() → Champions
cassiopeia.get_champion(region: Optional[Union[Region, str]] = None) → Champion
class cassiopeia.Champions(*args, **kwargs)
Bases: CassiopeiaLazyList
append(item)
Append object to the end of the list.
clear()
Remove all items from list.
contains(item: Any) → bool
copy()
Return a shallow copy of the list.
count(object)
Return number of occurrences of value.
delete(item: Any) → None
enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]
extend(iterable)
Extend list by appending elements from the iterable.
filter(function)
find(item: Any, reverse: bool = False) → Any
classmethod from_data(*args, **kwargs)
classmethod from_generator(generator: Generator, **kwargs)
included_data
A set of tags to return additional information for this champion when it's loaded.
```

**index**(*object, start: int = 0, stop: int = 9223372036854775807*)

Return first index of value.

Raises ValueError if the value is not present.

**insert**(*index: int, object*)

Insert object before index.

**locale**

The locale for this champion.

**platform**

The platform for this champion.

**pop**(*index: int = -1*)

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

The region for this champion.

**remove**(*object*)

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse**()

Reverse *IN PLACE*.

**search**(*item: Any, streaming: bool = False, reverse: bool = False*) → Union[SearchableList, Generator[Any, None, None]]

**sort**(\**, key=None, reverse=False*)

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**to\_dict**()

**to\_json**(\*\**kwargs*)

**version**

The version for this champion.

**class cassiopeia.Champion**(\**args, \*\*kwargs*)

Bases: CassiopeiaGhost

Searchable by ['str', 'int', 'Region', 'Platform', 'bool']

**ally\_tips**

The tips for playing with this champion.

**ban\_rates**

**blurb**

A short blurb about this champion.

**enemy\_tips**

The tips for playing against this champion.

**free\_to\_play**

Whether or not the champion is currently free to play.

**free\_to\_play\_new\_players**

Whether or not the champion is currently free to play for new players.

**id**

The champion's ID.

**image**

The image information for this champion.

**included\_data**

A set of tags to return additonal information for this champion when it's loaded.

**info**

Info about this champion.

**key**

The champion's key.

**load**(*load\_groups*: *Optional[Set]* = *None*) → *Champion*

**locale**

The locale for this champion.

**lore**

The champion's lore.

**name**

The champion's name.

**passive**

This champion's passive.

**platform**

The platform for this champion.

**play\_rates**

**recommended\_itemsets**

The champion's recommended itemsets.

**region**

The region for this champion.

**release\_date**

**resource**

The type of resource this champion uses.

**skins**

This champion's skins.

**spells**

This champion's spells.

**sprite****stats**

The champion's stats.

**tags**

The tags associated with this champion.

**title**

The champion's title.

**version**

The version for this champion.

**win\_rates**

```
class cassiopeia.core.staticdata.champion.Info(**kwargs)
```

Bases: CassiopeiaObject

**attack**

How attack-oriented Riot rates this champion.

**defense**

How defense-oriented Riot rates this champion.

**difficulty**

How Riot rates the difficulty of this champion.

**magic**

How magic-oriented Riot rates this champion.

```
class cassiopeia.core.staticdata.championStats(**kwargs)
```

Bases: CassiopeiaObject

**armor****armor\_per\_level****attack\_damage****attack\_damage\_per\_level****attack\_range****attack\_speed****critical\_strike\_chance****critical\_strike\_chance\_per\_level****health****health\_per\_level****health\_regen**

```
health_regen_per_level
magic_resist
magic_resist_per_level
mana
mana_per_level
mana_regen
mana_regen_per_level
movespeed
percent_attack_speed_per_level

class cassiopeia.core.staticdata.champion.Skin(**kwargs)
Bases: CassiopeiaObject
Searchable by ['str', 'int']
champion_key
    The key for the champion this belongs to.

id
    The skin's ID.

loading_image
    The skin's loading screen image.

loading_image_url
    The skin's loading screen image url.

name
    The skin's name.

number
    The skin number.

splash
    The skin's splash art.

splash_url
    The skin's splash art url.

class cassiopeia.core.staticdata.champion.Passive(**kwargs)
Bases: CassiopeiaObject
Searchable by ['str']
description
    The spells' description.

image_info
    The info about the spell's image, which can be pulled from datadragon.

name
    The spell's name.
```

**sanitized\_description**

The spell's sanitized description.

**class cassiopeia.core.staticdata.champion.RecommendedItems(\*\*kwargs)**

Bases: CassiopeiaObject

Searchable by ['str', 'Item', 'GameMode']

**classmethod from\_data(data: CoreData, region: Region)****item\_sets**

The recommended item sets.

**map**

The name of the map these recommendations are for.

**mode**

The game mode these recommendations are for.

**priority**

Whether this is a priority recommendation.

**title**

The title of these recommendations.

**type**

The type of recommendation.

**class cassiopeia.core.staticdata.champion.ItemSet(\*\*kwargs)**

Bases: CassiopeiaObject

Searchable by ['str', 'Item']

**classmethod from\_data(data: CoreData, region: Region)****items**

A dictionary of items mapped to how many of them are recommended.

**rec\_math**

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

**type**

The item set's type (e.g. starting items).

**class cassiopeia.core.staticdata.champion.ChampionSpell(\*\*kwargs)**

Bases: CassiopeiaObject

Searchable by ['str', 'Key']

**alternative\_images**

The alternative images for this spell. These won't exist after patch NN, when Riot standardized all images.

**cooldowns**

The cooldowns of this spell (per level).

**costs**

The resource costs of this spell (per level).

**description**

The spell's description.

**effects**

The level-by-level replacements for {{ e# }} tags in other values.

**effects\_by\_level**

The level-up changes, level-by-level.

**image\_info**

The info about the spell's image, which can be pulled from datadragon.

**key**

The spell's key.

**keyboard\_key**

Q, W, E, or R

**keywords**

The keywords for this spell.

**max\_rank**

The maximum rank this spell can attain.

**name**

The spell's name.

**range**

The maximum range of this spell. *self* if it has no range.

**resource**

The resource consumed when using this spell.

**sanitized\_description**

The spell's sanitized description.

**sanitized\_tooltip**

The spell's sanitized tooltip.

**tooltip**

The spell's tooltip.

**variables**

Contains spell data.

```
class cassiopeia.core.staticdata.champion.SpellVars(**kwargs)
```

Bases: CassiopeiaObject

Searchable by ['str']

**coefficients**

The scaling coefficients for this spell.

**dynamic**

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

**key**

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

**link**

Stat this spell scales from.

**ranks\_with**

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

```
class cassiopeia_championgg.core.ChampionGGStats(**kwargs)
```

Bases: CassiopeiaObject

**assists****ban\_rate****championgg\_metadata****damage\_composition****deaths****elo****games\_played****gold\_earned****id****kills****matchups****minions\_killed****neutral\_minions\_killed\_in\_enemy\_jungle****neutral\_minions\_killed\_in\_team\_jungle****patch****performance\_score****play\_rate****play\_rate\_by\_role****role****total\_damage\_taken****total\_healed****wards\_killed****win\_rate**

```
class cassiopeia_championgg.core.ChampionGGMatchups(*args, **kwargs)
```

Bases: CassiopeiaLazyList

```
class cassiopeia_championgg.core.ChampionGGMatchup(*args, **kwargs)
```

Bases: CassiopeiaGhost

Searchable by ['str']

**elo**

```
enemy
me
nmatches
patch
region
winrate

class cassiopeia_championgg.core.ChampionGGMatchupStats(data, id)
Bases: object
assists
champion
deaths
delta_assists
delta_deaths
delta_gold_earned
delta_killing_sprees
delta_kills
delta_minions_killed
delta_neutral_minions_killed_team_jungle
delta_ten_to_twenty
delta_thirty_to_end
delta_total_damage_dealt_to_champions
delta_twenty_to_thirty
delta_weighted_score
delta_wins
delta_zero_to_ten
gold_earned
id
killing_sprees
kills
minions_killed
neutral_minions_killed_team_jungle
```

```
role
thirty_to_end
total_damage_dealt_to_champions
twenty_to_thirty
weighted_score
winrate
wins
zero_to_ten
```

## Champion Mastersies

```
cassiopeia.get_champion_mastery(champion: Union[Champion, int, str], region: Optional[Union[Region,
str]] = None) → ChampionMastery

cassiopeia.get_champion_masteries(region: Optional[Union[Region, str]] = None) → ChampionMasteries

class cassiopeia.ChampionMasteries(*args, **kwargs)
    Bases: CassiopeiaLazyList

    append(item)
        Append object to the end of the list.

    clear()
        Remove all items from list.

    contains(item: Any) → bool

    copy()
        Return a shallow copy of the list.

    count(object)
        Return number of occurrences of value.

    delete(item: Any) → None

    enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]

    extend(iterable)
        Extend list by appending elements from the iterable.

    filter(function)

    find(item: Any, reverse: bool = False) → Any

    classmethod from_data(*args, **kwargs)

    classmethod from_generator(generator: Generator, **kwargs)

    index(object, start: int = 0, stop: int = 9223372036854775807)
        Return first index of value.

        Raises ValueError if the value is not present.
```

**insert**(*index: int, object*)

Insert object before index.

**platform**

**pop**(*index: int = -1*)

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

**remove**(*object*)

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse**()

Reverse *IN PLACE*.

**search**(*item: Any, streaming: bool = False, reverse: bool = False*) → Union[SearchableList, Generator[Any, None, None]]

**sort**(\**, key=None, reverse=False*)

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**summoner**

**to\_dict**()

**to\_json**(\*\**kwargs*)

**class cassiopeia.ChampionMastery(\*args, \*\*kwargs)**

Bases: CassiopeiaGhost

Searchable by ['str', 'int', 'bool', 'Arrow', 'Champion', 'Summoner']

**champion**

Champion for this entry.

**chest\_granted**

Is chest granted for this champion or not in current season?

**last\_played**

Last time this champion was played by this player.

**level**

Champion level for specified player and champion combination.

**platform**

**points**

Total number of champion points for this player and champion combination - used to determine champion level.

**points\_since\_last\_level**

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

**points\_until\_next\_level**

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

**region****summoner**

Summoner for this entry.

**tokens**

Number of tokens earned toward next mastery level.

**Items**

`cassiopeia.get_items()` → *Items*

**class cassiopeia.Items(\*args, \*\*kwargs)**

Bases: CassiopeiaLazyList

**append(item)**

Append object to the end of the list.

**clear()**

Remove all items from list.

**contains(item: Any) → bool**

**copy()**

Return a shallow copy of the list.

**count(object)**

Return number of occurrences of value.

**delete(item: Any) → None**

**enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]**

**extend(iterable)**

Extend list by appending elements from the iterable.

**filter(function)**

**find(item: Any, reverse: bool = False) → Any**

**classmethod from\_data(\*args, \*\*kwargs)**

**classmethod from\_generator(generator: Generator, \*\*kwargs)**

**included\_data**

A set of tags to return additional information for this item when it's loaded.

**index**(*object*, *start*: *int* = 0, *stop*: *int* = 9223372036854775807)

Return first index of value.

Raises ValueError if the value is not present.

**insert**(*index*: *int*, *object*)

Insert object before index.

**locale**

The locale for this item.

**platform**

**pop**(*index*: *int* = - 1)

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

**remove**(*object*)

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse**()

Reverse *IN PLACE*.

**search**(*item*: *Any*, *streaming*: *bool* = *False*, *reverse*: *bool* = *False*) → *Union*[*SearchableList*, *Generator*[*Any*, *None*, *None*]]

**sort**(\**, key=None, reverse=False*)

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**to\_dict**()

**to\_json**(\*\**kwargs*)

**version**

**class cassiopeia.Item**(\**args*, \*\**kwargs*)

Bases: CassiopeiaGhost

Searchable by ['str', 'int', 'Region', 'Platform', 'Map']

**builds\_from**

**builds\_into**

**champion**

**consume\_on\_full**

**consumed**

**description**

**effect**

**gold**

**group**

**hide**

**id**

The item's ID.

**image**

The image information for this item.

**in\_store**

**included\_data**

A set of tags to return additional information for this item when it's loaded.

**keywords**

**locale**

The locale for this item.

**maps**

**max\_stacks**

**name**

**plaintext**

**platform**

The platform for this item.

**region**

The region for this item.

**sanitized\_description**

**special\_recipe**

**sprite**

**stats**

**tags**

**tier**

**version**

The version for this item.

```
class cassiopeia.core.staticdata.item.ItemStats(**kwargs)
Bases: CassiopeiaObject

ability_power
armor
attack_damage
attack_speed
block
critical_strike_chance
critical_strike_damage
dodge
energy
energy_regen
health
health_regen
life_steal
magic_resist
mana
mana_regen
movespeed
percent_ability_power
percent_armor
percent_attack_damage
percent_attack_speed
percent_block
percent_critical_strike_damage
percent_health
percent_health_regen
percent_magic_resist
percent_mana_regen
percent_movespeed
percent_xp_bonus
spell_vamp
xp_bonus
```

## Language Strings

```
cassiopeia.get_language_strings() → LanguageStrings

class cassiopeia.LanguageStrings(*args, **kwargs)
    Bases: CassiopeiaGhost
    Searchable by []
        locale
        platform
        region
        strings
        type
        version
```

## Leagues

```
cassiopeia.Summoner.leagues

cassiopeia.get_challenger_league(region: Optional[Union[Region, str]] = None) → ChallengerLeague
cassiopeia.get_master_league(region: Optional[Union[Region, str]] = None) → MasterLeague

class cassiopeia.core.League(*args, **kwargs)
    Bases: CassiopeiaGhost
    Searchable by ['str', 'Queue', 'Tier']
        entries
        id
        name
        platform
        queue
        region
        tier

class cassiopeia.core.ChallengerLeague(*args, **kwargs)
    Bases: League
        entries
        id
        name
        platform
```

```
queue
region
tier

class cassiopeia.core.MasterLeague(*args, **kwargs)
Bases: CassiopeiaGhost

entries

id
name
platform
queue
region
tier

class cassiopeia.core.league.LeagueSummonerEntries(*args, **kwargs)
Bases: CassiopeiaLazyList

append(item)
    Append object to the end of the list.

clear()
    Remove all items from list.

contains(item: Any) → bool

copy()
    Return a shallow copy of the list.

count(object)
    Return number of occurrences of value.

delete(item: Any) → None

enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]

extend(iterable)
    Extend list by appending elements from the iterable.

filter(function)

find(item: Any, reverse: bool = False) → Any

fives

flex

classmethod from_data(*args, **kwargs)

classmethod from_generator(generator: Generator, **kwargs)
```

**index**(*object, start: int = 0, stop: int = 9223372036854775807*)

Return first index of value.

Raises ValueError if the value is not present.

**insert**(*index: int, object*)

Insert object before index.

**platform**

**pop**(*index: int = - 1*)

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

**remove**(*object*)

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse**()

Reverse *IN PLACE*.

**search**(*item: Any, streaming: bool = False, reverse: bool = False*) → Union[SearchableList, Generator[Any, None, None]]

**sort**(\**, key=None, reverse=False*)

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**threes**

**to\_dict**()

**to\_json**(\*\**kwargs*)

**class cassiopeia.core.league.MiniSeries(\*\**kwargs*)**

Bases: CassiopeiaObject

**losses**

**not\_played**

The number of games in the player's promos that they haven't played yet.

**progress**

A list of True/False for the number of games the played in the mini series indicating if the player won or lost.

**wins**

**wins\_required**

2 or 3 wins will be required for promotion.

```
class cassiopeia.core.league.LeagueEntries(*args, **kwargs)
Bases: CassiopeiaLazyList

append(item)
    Append object to the end of the list.

clear()
    Remove all items from list.

contains(item: Any) → bool

copy()
    Return a shallow copy of the list.

count(object)
    Return number of occurrences of value.

delete(item: Any) → None

division

enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]

extend(iterable)
    Extend list by appending elements from the iterable.

filter(function)

find(item: Any, reverse: bool = False) → Any

classmethod from_data(*args, **kwargs)

classmethod from_generator(generator: Generator, region: Optional[Union[Region, str]] = None,
                           queue: Optional[Queue] = None, tier: Optional[Tier] = None, division:
                           Optional[Division] = None, **kwargs)

index(object, start: int = 0, stop: int = 9223372036854775807)
    Return first index of value.

    Raises ValueError if the value is not present.

insert(index: int, object)
    Insert object before index.

platform

pop(index: int = -1)
    Remove and return item at index (default last).

    Raises IndexError if list is empty or index is out of range.

queue

region

remove(object)
    Remove first occurrence of value.

    Raises ValueError if the value is not present.
```

**reverse()**

Reverse *IN PLACE*.

**search**(*item: Any, streaming: bool = False, reverse: bool = False*) → Union[SearchableList, Generator[Any, None, None]]

**sort(\*, key=None, reverse=False)**

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**tier****to\_dict()****to\_json(\*\*kwargs)**

**class cassiopeia.core.league.LeagueEntry(\*args, \*\*kwargs)**

Bases: CassiopeiaGhost

Searchable by ['str', 'bool', 'Division', 'Summoner', 'Queue']

**division****fresh\_blood**

**classmethod from\_data**(*data: LeagueEntryData, loaded\_groups: Optional[Set[Type[CoreData]]] = None, league: Optional[League] = None*)

**hot\_streak****inactive****league****league\_points****losses****platform**

The platform for this champion.

**promos****queue****region**

The region for this champion.

**role****summoner****tier**

**veteran**

**wins**

## Locales

`cassiopeia.get_locales() → List[str]`

**class cassiopeia.Locales(\*args, \*\*kwargs)**

Bases: CassiopeiaLazyList

**append(item)**

Append object to the end of the list.

**clear()**

Remove all items from list.

**contains(item: Any) → bool**

**copy()**

Return a shallow copy of the list.

**count(object)**

Return number of occurrences of value.

**delete(item: Any) → None**

**enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]**

**extend(iterable)**

Extend list by appending elements from the iterable.

**filter(function)**

**find(item: Any, reverse: bool = False) → Any**

**classmethod from\_data(\*args, \*\*kwargs)**

**classmethod from\_generator(generator: Generator, \*\*kwargs)**

**index(object, start: int = 0, stop: int = 9223372036854775807)**

Return first index of value.

Raises ValueError if the value is not present.

**insert(index: int, object)**

Insert object before index.

**platform**

**pop(index: int = -1)**

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

**remove(*object*)**

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse()**

Reverse *IN PLACE*.

**search(*item*: Any, *streaming*: bool = False, *reverse*: bool = False) → Union[SearchableList, Generator[Any, None, None]]****sort(\*, *key*=None, *reverse*=False)**

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**to\_dict()****to\_json(\*\**kwargs*)**

## Maps

**cassiopeia.get\_maps() → Maps****class cassiopeia.Maps(\*args, \*\*kwargs)**

Bases: CassiopeiaLazyList

**append(*item*)**

Append object to the end of the list.

**clear()**

Remove all items from list.

**contains(*item*: Any) → bool****copy()**

Return a shallow copy of the list.

**count(*object*)**

Return number of occurrences of value.

**delete(*item*: Any) → None****enumerate(*item*: Any, *reverse*: bool = False) → Generator[Tuple[int, Any], None, None]****extend(*iterable*)**

Extend list by appending elements from the iterable.

**filter(*function*)****find(*item*: Any, *reverse*: bool = False) → Any**

```
classmethod from_data(*args, **kwargs)
classmethod from_generator(generator: Generator, **kwargs)
index(object, start: int = 0, stop: int = 9223372036854775807)
    Return first index of value.
    Raises ValueError if the value is not present.
insert(index: int, object)
    Insert object before index.
locale
platform
pop(index: int = -1)
    Remove and return item at index (default last).
    Raises IndexError if list is empty or index is out of range.
region
remove(object)
    Remove first occurrence of value.
    Raises ValueError if the value is not present.
reverse()
    Reverse IN PLACE.
search(item: Any, streaming: bool = False, reverse: bool = False) → Union[SearchableList, Generator[Any, None, None]]
sort(*, key=None, reverse=False)
    Sort the list in ascending order and return None.
    The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).
    If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.
    The reverse flag can be set to sort in descending order.
to_dict()
to_json(**kwargs)
version
class cassiopeia.Map(*args, **kwargs)
    Bases: CassiopeiaGhost
    Searchable by ['str', 'int']
id
    The map's ID.
image
```

**locale**

The locale for this map.

**name****platform****region****sprite****unpurchasable\_items****version**

The version for this map.

## Matches

cassiopeia.Summoner.**match\_history**

```
cassiopeia.get_match_history(puuid: Optional[str] = None, start_time: Optional[Arrow] = None, end_time: Optional[Arrow] = None, queue: Optional[Queue] = None, type: Optional[MatchType] = None, start: Optional[int] = None, count: Optional[int] = None)
```

cassiopeia.get\_match(region: Optional[Union[Region, str]] = None) → Match

**class cassiopeia.core.match.MatchHistory(\*args, \*\*kwargs)**

Bases: CassiopeiaLazyList

The match history for a summoner. By default, this will return the entire match history.

**append(item)**

Append object to the end of the list.

**clear()**

Remove all items from list.

**contains(item: Any) → bool**

**continent() → Continent**

**copy()**

Return a shallow copy of the list.

**count**

Return number of occurrences of value.

**delete(item: Any) → None**

**end\_time**

**enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]**

**extend(iterable)**

Extend list by appending elements from the iterable.

```
filter(function)
find(item: Any, reverse: bool = False) → Any
classmethod from_data(*args, **kwargs)
classmethod from_generator(generator: Generator, **kwargs)
index(object, start: int = 0, stop: int = 9223372036854775807)
    Return first index of value.
    Raises ValueError if the value is not present.
insert(index: int, object)
    Insert object before index.
match_type() → MatchType
pop(index: int = -1)
    Remove and return item at index (default last).
    Raises IndexError if list is empty or index is out of range.
queue() → Queue
remove(object)
    Remove first occurrence of value.
    Raises ValueError if the value is not present.
reverse()
    Reverse IN PLACE.
search(item: Any, streaming: bool = False, reverse: bool = False) → Union[SearchableList, Generator[Any, None, None]]
sort(*, key=None, reverse=False)
    Sort the list in ascending order and return None.
    The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).
    If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.
    The reverse flag can be set to sort in descending order.
start
start_time
to_dict()
to_json(**kwargs)
class cassiopeia.Match(*args, **kwargs)
    Bases: CassiopeiaGhost
    Searchable by ['str', 'Continent', 'Queue', 'MatchType', 'GameMode', 'Map', 'GameType', 'Item', 'Patch', 'Summoner', 'SummonerSpell']
```

**blue\_team**

**continent**  
The continent for this match.

**creation**

**duration**

**exists**

**classmethod from\_match\_reference**(*ref: MatchReferenceData*)

**game\_type**

**id**

**is\_remake**

**kills\_heatmap()**

**map**

**mode**

**participants**

**patch**

**platform**  
The platform for this match.

**queue**

**red\_team**

**region**  
The region for this match.

**start**

**teams**

**timeline**

**type**

**version**

**class cassiopeia.core.match.Team(\*\*kwargs)**  
Bases: CassiopeiaObject  
Searchable by ['str', 'bool', 'Champion', 'Summoner', 'SummonerSpell']

**bans**

**baron\_kills**

**dominion\_score**

**dragon\_kills**

```
first_baron
first_blood
first_dragon
first_inhibitor
first_rift_herald
first_tower
classmethod from_data(data: CoreData, match: Match)

inhibitor_kills
participants
rift_herald_kills
side
tower_kills
win

class cassiopeia.core.match.Participant(**kwargs)
Bases: CassiopeiaObject
Searchable by ['str', 'Summoner', 'Champion', 'Side', 'Rune', 'SummonerSpell']
champion
cumulative_timeline
ended_in_early_surrender
enemy_team
classmethod from_data(data: CoreData, match: Match)

id
individual_position
is_bot
lane
role
runes
side
skill_order
stat_runes
stats
```

```
summoner
summoner_spell_d
summoner_spell_f
team
team_position
timeline
version

class cassiopeia.core.match.ParticipantStats(**kwargs)
    Bases: CassiopeiaObject
    Searchable by ['str', 'Item']

    assists
    baron_kills
    bounty_level
    champion_experience
    champion_transform
    consumables_purchased
    damage_dealt_to_buildings
    damage_dealt_to_objectives
    damage_dealt_to_turrets
    damage_self_mitigated
    deaths
    double_kills
    dragon_kills
    first_blood_assist
    first_blood_kill
    first_tower_assist
    first_tower_kill

    classmethod from_data(data: ParticipantStatsData, match: Match, participant: Participant)
    gold_earned
    gold_spent
    inhibitor_kills
```

inhibitor\_takedowns  
inhibitors\_lost  
items  
items\_purchased  
kda  
killing\_sprees  
kills  
largest\_critical\_strike  
largest\_killing\_spree  
largest\_multi\_kill  
level  
longest\_time\_spent\_living  
magic\_damage\_dealt  
magic\_damage\_dealt\_to\_champions  
magic\_damage\_taken  
neutral\_minions\_killed  
nexus\_kills  
nexus\_lost  
nexus\_takedowns  
objectives\_stolen  
objectives\_stolen\_assists  
penta\_kills  
physical\_damage\_dealt  
physical\_damage\_dealt\_to\_champions  
physical\_damage\_taken  
quadra\_kills  
sight\_wards\_bought  
spell\_1\_casts  
spell\_2\_casts  
spell\_3\_casts  
spell\_4\_casts

```
summoner_spell_1_casts
summoner_spell_2_casts
time_CCing_others
time_played
total_damage_dealt
total_damage_dealt_to_champions
total_damage_shielded_on_teammates
total_damage_taken
total_heal
total_heals_on_teammates
total_minions_killed
total_time_cc_dealt
total_time_spent_dead
total_units_healed
triple_kills
true_damage_dealt
true_damage_dealt_to_champions
true_damage_taken
turret_kills
turret_takedowns
turrets_lost
unreal_kills
vision_score
vision_wards_bought
vision_wards_placed
wards_killed
wards_placed
win

class cassiopeia.core.match.ParticipantTimeline
Bases: object
champion_assists
```

```
champion_deaths
champion_kills
events
frames
classmethod from_data(match: Match)

class cassiopeia.core.match.Timeline(*args, **kwargs)
    Bases: CassiopeiaGhost
    continent
    first_tower_fallen
    frame_interval
    frames
    id
    platform
    region

class cassiopeia.core.match.Frame(**kwargs)
    Bases: CassiopeiaObject
    events
    participant_frames
    timestamp

class cassiopeia.core.match.ParticipantFrame(**kwargs)
    Bases: CassiopeiaObject
    creep_score
    current_gold
    dominion_score
    experience
    gold_earned
    level
    neutral_minions_killed
    participant_id
    position
    team_score
```

```
class cassiopeia.core.match.Event(**kwargs)
Bases: CassiopeiaObject
Searchable by ['str']

after_id
ascended_type
assisting_participants
before_id
building_type
captured_point
creator_id
item_id
killer_id
lane_type
level_up_type
monster_sub_type
monster_type
participant_id
position
side
skill
timestamp
tower_type
type
    CHAMPION_KILL, WARD_PLACED, WARD_KILL, BUILDING_KILL, ELITE_MONSTER_KILL,
    ITEM_PURCHASED, ITEM SOLD, ITEM_DESTROYED, ITEM_UNDO, SKILL_LEVEL_UP, AS-
    CENDED_EVENT, CAPTURE_POINT, PORO_KING_SUMMON

    Type
        Legal values

victim_id
ward_type

class cassiopeia.core.match.Position(**kwargs)
Bases: CassiopeiaObject

location
x
```

y

```
class cassiopeia.core.match.CumulativeTimeline(id: int, participant_timeline: ParticipantTimeline)
```

Bases: object

```
class cassiopeia.core.match.ParticipantState(id: int, time: timedelta, participant_timeline:
```

ParticipantTimeline)

Bases: object

The state of a participant at a given point in the timeline.

assists

creep\_score

current\_gold

deaths

dominion\_score

experience

gold\_earned

items

kda

kills

level

neutral\_minions\_killed

objectives

Number of objectives assisted in.

position

skills

team\_score

## Patch

```
class cassiopeia.Patch(region: Union[str, Region], name: str, start: Union[Arrow, float], end:
```

Optional[Union[Arrow, float]])

Bases: object

end

classmethod from\_date(date: Arrow, region: Union[Region, str]) → Patch

classmethod from\_str(string: str, region: Union[Region, str]) → Patch

classmethod latest(region: Optional[Union[Region, str]] = None) → Patch

```
major
majorminor
minor
name
region
revision
start
```

## Profile Icons

```
cassiopeia.Summoner.profile_icon
```

```
cassiopeia.get_profile_icons() → ProfileIcons
```

```
class cassiopeia.ProfileIcons(*args, **kwargs)
```

Bases: CassiopeiaLazyList

```
append(item)
```

Append object to the end of the list.

```
clear()
```

Remove all items from list.

```
contains(item: Any) → bool
```

```
copy()
```

Return a shallow copy of the list.

```
count(object)
```

Return number of occurrences of value.

```
delete(item: Any) → None
```

```
enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]
```

```
extend(iterable)
```

Extend list by appending elements from the iterable.

```
filter(function)
```

```
find(item: Any, reverse: bool = False) → Any
```

```
classmethod from_data(*args, **kwargs)
```

```
classmethod from_generator(generator: Generator, **kwargs)
```

```
index(object, start: int = 0, stop: int = 9223372036854775807)
```

Return first index of value.

Raises ValueError if the value is not present.

**insert**(*index*: *int*, *object*)

Insert object before index.

**locale**

**platform**

**pop**(*index*: *int* = - 1)

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

**remove**(*object*)

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse**()

Reverse *IN PLACE*.

**search**(*item*: *Any*, *streaming*: *bool* = *False*, *reverse*: *bool* = *False*) → *Union[SearchableList, Generator[Any, None, None]]*

**sort**(\**key=None*, *reverse=False*)

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**to\_dict**()

**to\_json**(\*\**kwargs*)

**version**

**class** cassiopeia.core.staticdata.profileicon.**ProfileIcon**(\**args*, \*\**kwargs*)

Bases: CassiopeiaGhost

Searchable by [‘int’, ‘str’, ‘Image’]

**id**

**image**

**locale**

The locale for this profile icon.

**name**

**platform**

The platform for this profile icon.

**region**

The region for this profile icon.

**url****version**

The version for this profile icon.

## Realms

`cassiopeia.get_realms()` → *Realms*

**class cassiopeia.Realms(\*args, \*\*kwargs)**

Bases: CassiopeiaGhost

Searchable by []

**cdn****css\_version****language****latest\_data\_dragon****latest\_versions**

Latest changed version for each data type listed.

**legacy\_mode****locale**

The locale for this realm.

**max\_profile\_icon\_id****platform**

The platform for this realm.

**region**

The region for this realm.

**store****version**

## Runes

`cassiopeia.get_runes()` → *Runes*

**class cassiopeia.Runes(\*args, \*\*kwargs)**

Bases: CassiopeiaLazyList

**append(item)**

Append object to the end of the list.

**clear()**

Remove all items from list.

**contains(item: Any) → bool**

**copy()**

Return a shallow copy of the list.

**count(*object*)**

Return number of occurrences of value.

**delete(*item*: *Any*) → *None***

**domination**

**enumerate(*item*: *Any*, *reverse*: *bool* = *False*) → Generator[Tuple[int, *Any*], *None*, *None*]**

**extend(*iterable*)**

Extend list by appending elements from the iterable.

**filter(*function*)**

**find(*item*: *Any*, *reverse*: *bool* = *False*) → *Any***

**classmethod from\_data(\*args, \*\*kwargs)**

**classmethod from\_generator(*generator*: *Generator*, \*\*kwargs)**

**included\_data**

A set of tags to return additonal information for this champion when it's loaded.

**index(*object*, *start*: *int* = 0, *stop*: *int* = 9223372036854775807)**

Return first index of value.

Raises ValueError if the value is not present.

**insert(*index*: *int*, *object*)**

Insert object before index.

**inspiration**

**keystones**

**locale**

The locale for this champion.

**platform**

**pop(*index*: *int* = -1)**

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**precision**

**region**

**remove(*object*)**

Remove first occurrence of value.

Raises ValueError if the value is not present.

**resolve**

**reverse()**

Reverse *IN PLACE*.

**search**(*item*: `Any`, *streaming*: `bool` = `False`, *reverse*: `bool` = `False`) → `Union[SearchableList, Generator[Any, None, None]]`

**sorcery**

**sort**(\**key*=`None`, *reverse*=`False`)

Sort the list in ascending order and return `None`.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**to\_dict**()

**to\_json**(\*\**kwargs*)

**version**

**class** `cassiopeia.Rune`(\**args*, \*\**kwargs*)

Bases: `CassiopeiaGhost`

Searchable by [‘str’, ‘int’, ‘RunePath’, ‘Region’, ‘Platform’]

**id**

The rune’s ID.

**image**

The image information for this rune.

**included\_data**

A set of tags to return additional information for this champion when it’s loaded.

**is\_keystone**

**locale**

The locale for this rune.

**long\_description**

**name**

The rune’s name.

**path**

**platform**

The platform for this rune.

**region**

The region for this rune.

**short\_description**

**tier**

**version**

The version for this rune.

## Status

```
cassiopeia.get_status() → ShardStatus

class cassiopeia.ShardStatus(*args, **kwargs)
    Bases: CassiopeiaGhost
    Searchable by []
    hostname
    locales
    name
    platform
    region
    services
    slug
```

## Spectator

```
cassiopeia.Summoner.current_match

cassiopeia.get_current_match(region: Optional[Union[Region, str]] = None) → CurrentMatch
cassiopeia.get_featured_matches() → FeaturedMatches

class cassiopeia.FeaturedMatches(*args, **kwargs)
    Bases: CassiopeiaLazyList
    append(item)
        Append object to the end of the list.
    clear()
        Remove all items from list.
    client_refresh_interval
    contains(item: Any) → bool
    copy()
        Return a shallow copy of the list.
    count(object)
        Return number of occurrences of value.
    delete(item: Any) → None
    enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]
    extend(iterable)
        Extend list by appending elements from the iterable.
```

**filter**(*function*)

**find**(*item*: *Any*, *reverse*: *bool* = *False*) → *Any*

**classmethod from\_data**(\**args*, \*\**kwargs*)

**classmethod from\_generator**(*generator*: *Generator*, \*\**kwargs*)

**index**(*object*, *start*: *int* = 0, *stop*: *int* = 9223372036854775807)

Return first index of value.

Raises ValueError if the value is not present.

**insert**(*index*: *int*, *object*)

Insert object before index.

**platform**

**pop**(*index*: *int* = - 1)

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

**remove**(*object*)

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse**()

Reverse *IN PLACE*.

**search**(*item*: *Any*, *streaming*: *bool* = *False*, *reverse*: *bool* = *False*) → Union[SearchableList, Generator[*Any*, None, None]]

**sort**(\**, key=None, reverse=False*)

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**to\_dict**()

**to\_json**(\*\**kwargs*)

**class cassiopeia.core.spectator.CurrentMatch**(\**args*, \*\**kwargs*)

Bases: CassiopeiaGhost

Searchable by []

**blue\_team**

**creation**

**duration**

```
exists
classmethod from_data(data: CurrentGameInfoData, summoner: Union[Summoner, str])
id
map
mode
observer_key
participants
platform
queue
red_team
region
teams
type

class cassiopeia.core.spectator.Team(**kwargs)
Bases: CassiopeiaObject
Searchable by []
bans
classmethod from_data(data: CoreData, match: CurrentMatch)
participants
side

class cassiopeia.core.spectator.Participant(**kwargs)
Bases: CassiopeiaObject
Searchable by ['str', 'Summoner', 'Champion']
champion
classmethod from_data(data: CoreData, match: CurrentMatch)
is_bot
runes
side
summoner
summoner_spell_d
summoner_spell_f
team
```

## Summoners

```
cassiopeia.get_summoner(*, account_id: Optional[str] = None, name: Optional[str] = None, region: Optional[Union[Region, str]] = None) → Summoner
```

```
class cassiopeia.Summoner(*args, **kwargs)
```

Bases: CassiopeiaGhost

Searchable by ['str', 'Region', 'Platform']

**account\_id**

**champion\_masteries**

**current\_match**

**exists**

**id**

**league\_entries**

**level**

**match\_history**

**match\_history\_uri**

**name**

**platform**

The platform for this summoner.

**profile\_icon**

**puuid**

**ranks**

**region**

The region for this summoner.

**revision\_date**

**sanitized\_name**

**verification\_string**

## Summoner Spells

```
cassiopeia.get_summoner_spells() → SummonerSpells
```

```
class cassiopeia.SummonerSpells(*args, **kwargs)
```

Bases: CassiopeiaLazyList

**append(item)**

Append object to the end of the list.

**clear()**

Remove all items from list.

**contains(item: Any) → bool**

**copy()**

Return a shallow copy of the list.

**count(object)**

Return number of occurrences of value.

**delete(item: Any) → None**

**enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]**

**extend(iterable)**

Extend list by appending elements from the iterable.

**filter(function)**

**find(item: Any, reverse: bool = False) → Any**

**classmethod from\_data(\*args, \*\*kwargs)**

**classmethod from\_generator(generator: Generator, \*\*kwargs)**

**included\_data**

A set of tags to return additional information for this champion when it's loaded.

**index(object, start: int = 0, stop: int = 9223372036854775807)**

Return first index of value.

Raises ValueError if the value is not present.

**insert(index: int, object)**

Insert object before index.

**locale**

The locale for this champion.

**platform**

**pop(index: int = -1)**

Remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

**region**

**remove(object)**

Remove first occurrence of value.

Raises ValueError if the value is not present.

**reverse()**

Reverse *IN PLACE*.

**search(item: Any, streaming: bool = False, reverse: bool = False) → Union[SearchableList, Generator[Any, None, None]]**

**sort(\*, key=None, reverse=False)**

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

**to\_dict()****to\_json(\*\*kwargs)****version****class cassiopeia.SummonerSpell(\*args, \*\*kwargs)**

Bases: CassiopeiaGhost

Searchable by ['str']

**alternative\_images**

The alternative images for this spell. These won't exist after patch NN, when Riot standardized all images.

**cooldowns**

The cooldowns of this spell (per level).

**costs**

The resource costs of this spell (per level).

**description**

The spell's description.

**effects**

The level-by-level replacements for {{ e# }} tags in other values.

**id**

The spell's id.

**image****included\_data**

The data to included in the query for this summoner spell.

**key**

The spell's key.

**locale**

The locale for this summoner spell.

**max\_rank**

The maximum rank this spell can attain.

**modes****name**

The spell's name.

**platform**

The platform for this summoner spell.

**range**

The maximum range of this spell. *self* if it has no range.

**region**

The region for this summoner spell.

**resource**

The resource consumed when using this spell.

**sanitized\_description**

The spell's sanitized description.

**sanitized\_tooltip**

The spell's sanitized tooltip.

**sprite**

**tooltip**

The spell's tooltip.

**variables**

Contains spell data.

**version**

The version for this summoner spell.

**class** cassiopeia.core.staticdata.summonerspell.**SpellVars**(\*\*kwargs)

Bases: CassiopeiaObject

Searchable by ['str']

**coefficients**

The scaling coefficients for this spell.

**dynamic**

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

**key**

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

**link**

Stat this spell scales from.

**ranks\_with**

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

## Versions

```
cassiopeia.get_versions() → List[str]  
cassiopeia.get_version(region: Optional[Union[Region, str]] = None) → Union[None, str]  
class cassiopeia.Versions(*args, **kwargs)  
    Bases: CassiopeiaLazyList  
append(item)  
    Append object to the end of the list.  
clear()  
    Remove all items from list.  
contains(item: Any) → bool  
copy()  
    Return a shallow copy of the list.  
count(object)  
    Return number of occurrences of value.  
delete(item: Any) → None  
enumerate(item: Any, reverse: bool = False) → Generator[Tuple[int, Any], None, None]  
extend(iterable)  
    Extend list by appending elements from the iterable.  
filter(function)  
find(item: Any, reverse: bool = False) → Any  
classmethod from_data(*args, **kwargs)  
classmethod from_generator(generator: Generator, **kwargs)  
index(object, start: int = 0, stop: int = 9223372036854775807)  
    Return first index of value.  
    Raises ValueError if the value is not present.  
insert(index: int, object)  
    Insert object before index.  
platform  
pop(index: int = -1)  
    Remove and return item at index (default last).  
    Raises IndexError if list is empty or index is out of range.  
region  
remove(object)  
    Remove first occurrence of value.  
    Raises ValueError if the value is not present.
```

### `reverse()`

Reverse *IN PLACE*.

`search(item: Any, streaming: bool = False, reverse: bool = False) → Union[SearchableList, Generator[Any, None, None]]`

### `sort(*, key=None, reverse=False)`

Sort the list in ascending order and return None.

The sort is in-place (i.e. the list itself is modified) and stable (i.e. the order of two equal elements is maintained).

If a key function is given, apply it once to each list item and sort them, ascending or descending, according to their function values.

The reverse flag can be set to sort in descending order.

### `to_dict()`

### `to_json(**kwargs)`

## 4.2.2 Setup

Cassiopeia requires Python 3.6 and we highly recommend installing Anaconda with Python 3.6.

### Install using pip

Simply `pip install cassiopeia` to get the latest release. (See the [pip](#) install page if you do not have pip installed.) If you want to pull the most recent version, you can install directly from GitHub using `pip install git+https://github.com/meraki-analytics/cassiopeia.git` instead. We may not make a PyPy release (which pip usually pulls from) for small changes to the code.

### PyCurl Issues

You may have some issues during installation due to PyCurl. Try the installation first, and if you have issues with pycurl come back and read this section. If Cass installed properly but is throwing certificate errors, skip to the 3rd paragraph.

At the moment PyCurl does not fully support installation with Python 3.6 and many people have had issues. The easiest thing to do (and what we *highly* recommend) is to install Python 3.6 via [Anaconda](#). Anaconda is a package manager for Python and provides many packages that are difficult to install without Anaconda. If you do not or can't use Anaconda, you'll need download and install Curl, then use `easy_install` to install PyCurl and link it to the proper Curl libraries. This isn't much fun, and again we recommend Anaconda to ease the process.

If you successfully installed Cass but it's throwing a certificate error, you probably just need to `pip install certifi`. This should solve any certificate errors.

If you are having more problems, let us know via the Riot API discord server or our Meraki discord server.

## Alternative library support

In case PyCurl is not installed, Cassiopeia falls back to [Requests](#), this is a direct dependency and the user is not required to install the library manually.

Furthermore, [ujson](#) is used instead of Python's default [json](#) module when available. To use ujson in conjunction with Requests, the following patch needs to be applied before running any API requests.

```
# see https://github.com/psf/requests/issues/1595#issuecomment-504030697 for more details
import requests
import ujson

requests.models.complexjson = ujson
```

## 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, run `pip install -r requirements.txt` in the newly downloaded cassiopeia directory to install the dependencies.

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 with 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/IDE.

Google can probably give you more information as well, and note that the path name you add to your `PYTHONPATH` should end in `.../cassiopeia`.

## Setting Your API Key and Other Settings

By default, Cass pulls your API key from an environment variable called `RIOT_API_KEY`. You can modify this behavior by calling `cass.set_riot_api_key(...)` or by creating your own configuration for Cass (see [Settings](#) for more info). However, we encourage new users to use Cass's default configuration.

To create an environment variable on Windows, follow the directions [here](#). On Linux or Mac, add `export RIOT_API_KEY='<YOUR_API_KEY>'` to the end of your shell rc file (this should be `~/.bashrc` for most), where `<YOUR_API_KEY>` is your Riot-issued API key. Then restart your terminal/IDE.

## Plugins

Cass has plugins that enable additional functionality. See [Plugins](#) for more information about how to install each plugin.

### 4.2.3 Settings

There are many settings in Cassiopeia that control how the framework works, and more settings will be added as the code is expanded.

Use `cass.apply_settings(...)` and pass in a json filename, a dictionary, or a `cassiopeia.Settings` object to set Cass's parameters. Cass will use its own default settings if you do not run `cass.apply_settings`.

The method `cass.get_default_config()` will return a dictionary that contains the default settings that Cass uses. You can call this method, modify the returned dictionary, then pass it to `cass.apply_settings` to overwrite the default settings.

The most important setting is your Riot API key. It can be set programmatically (which will override the value specified in the settings).

Each setting is explained below, and should be added as separate entries to your settings dictionary/json.

## Globals

The "default\_region" setting should be set to the string version of the region that the Riot API requires (in all caps), for example "NA" for North America. This can be set programmatically using `cass.set_default_region`.

The "version\_from\_match" variable determines which version of the static data for matches is loaded (this includes, for example, the items for each participant). Valid values are "version", "patch", and "latest". If set to "version", the static data for the match's version will be loaded correctly; however, this requires pulling the match data for all matches. If you only want to use match reference data (and will not pull the full data for every match), you should use either "patch" or "latest". "patch" will make a reasonable attempt to get the match's correct version based on its creation date (which is provided in the match reference data); however, if you pull a summoner's full match history, you will pull many versions of the static data, which may take a long time. In addition, the patch dates / times may be slightly off and may depend on the region. For small applications that barely uses the static data, pulling multiple versions of the static data is likely overkill. If that is the case, you should set this variable to "latest", in which case the static data for the most recent version will be used; this, however, could result in missing or incorrect data if parts of the static data are accessed that have changed from patch to patch. The default is to use the patch if the match hasn't yet been loaded, which is a nice compromise between ensuring you, the user, always have correct data while also preventing new users from pulling a massive amount of unnecessary match data. It's likely that the patch dates aren't perfect, so be aware of this and please report and inconsistencies.

Below is an example:

```
{  
    ...,  
    "global": {  
        "version_from_match": "patch",  
        "default_region": null  
    }  
    ...  
}
```

## Data Pipeline

This setting is extremely important and therefore has its own page ([Data Pipeline](#)). However, our defaults will likely work for you if you're just getting started.

## Riot API

The Riot API variable is an attribute of the `pipeline` variable, but it has a variety of settings relevant to the Riot API.

The "api\_key" should be set to your Riot API key. You can instead supply an environment variable name that contains your API key (this is recommended so that you can push your settings file to version control without revealing your API key). This variable can be set programmatically via `cass.set_riot_api_key`.

The "limit\_sharing" variable specifies what fraction of your API key should be used for your server. This is useful when you have multiple servers that you want to split your API key over. The default (if not set) is `1.0`, and valid values are between `0.0` and `1.0`.

## Request Handling

The "request\_error\_handling" variable specifies how errors returned by the Riot API should be handled. There are three options, each of which has its own set of parameters: "throw" simply causes the error returned by the Riot API to be thrown to you, the user; "exponential\_backoff" will exponentially backoff; and "retry\_from\_headers" will attempt to use the "retry-after" header in the response to retry after the specified amount of time. The 429 error code can be handled differently depending on which type of rate limiting cause it. See the example below for the specific structure for these settings.

"throw" takes no arguments.

"exponential\_backoff" takes three arguments: `initial_backoff` specifies the initial time to pause before making another request, `backoff_factor` specifies what to multiply the `initial_backoff` by for each subsequent failure, and `max_attempts` specifies the maximum number of calls to make before throwing the error.

"retry\_from\_headers" takes one argument: `max_attempts` specifies the maximum number of calls to make before throwing the error.

Below is an example, and these settings are the default if any value is not specified:

```
"RiotAPI": {
    "api_key": "RIOT_API_KEY",
    "limiting_share": 1.0,
    "request_error_handling": {
        "404": {
            "strategy": "throw"
        },
        "429": {
            "service": {
                "strategy": "exponential_backoff",
                "initial_backoff": 1.0,
                "backoff_factor": 2.0,
                "max_attempts": 4
            },
            "method": {
                "strategy": "retry_from_headers",
                "max_attempts": 5
            }
        }
    }
}
```

(continues on next page)

(continued from previous page)

```
        "application": {
            "strategy": "retry_from_headers",
            "max_attempts": 5
        }
    },
    "500": {
        "strategy": "throw"
    },
    "503": {
        "strategy": "throw"
    },
    "timeout": {
        "strategy": "throw"
    }
}
```

An alternative setting for `request_error_handling` is below, which will retry 50x errors:

```
"request_error_handling": {
    "404": {
        "strategy": "throw"
    },
    "429": {
        "service": {
            "strategy": "exponential_backoff",
            "initial_backoff": 1.0,
            "backoff_factor": 2.0,
            "max_attempts": 4
        },
        "method": {
            "strategy": "retry_from_headers",
            "max_attempts": 5
        },
        "application": {
            "strategy": "retry_from_headers",
            "max_attempts": 5
        }
    },
    "500": {
        "strategy": "exponential_backoff",
        "initial_backoff": 1.0,
        "backoff_factor": 2.0,
        "max_attempts": 4
    },
    "503": {
        "strategy": "exponential_backoff",
        "initial_backoff": 1.0,
        "backoff_factor": 2.0,
        "max_attempts": 4
    },
    "timeout": {

```

(continues on next page)

(continued from previous page)

```

        "strategy": "throw"
    },
    "403": {
        "strategy": "throw"
    }
}

```

## Logging

The "logging" section defines variables related to logging and print statements.

The "print\_calls" variable should be set to `true` or `false` and determines whether http calls (e.g. to the Riot API or Data Dragon) are printed. Similarly, the "print\_riot\_api\_key" variable will print your Riot API key if set to `true`.

"core" and "default" are two loggers that are currently implemented in Cass, and you can set the logging levels using these variables. Acceptable values are the logging levels for python's logging module (e.g. "INFO" and "WARNING").

Example:

```

"logging": {
    "print_calls": true,
    "print_riot_api_key": false,
    "default": "WARNING",
    "core": "WARNING"
}

```

## Plugins

The "plugins" section defines which plugins Cassiopeia will use. See [Plugins](#) for specifics for each plugin.

### 4.2.4 How Cass Works

There are a few major parts that make Cass work, with minor parts that go along with them. These are discussed below.

#### Two Interfaces

Cass has two interfaces that work nearly identically. Depending on your coding style, you can choose the one that you prefer. One uses `.get_...` methods to get objects, while the other prefers constructors to create objects. Both are equally good. As an example, both `cass.get_summoner(name="Kalturi", region="NA")` and `Summoner(name="Kalturi", region="NA")` work exactly the same.

### Settings

There are a few settings in Cass that should be modified, and more that can be modified. See [Settings](#) for more info.

### Ghost Loading

A *ghost object* is an object that can be instantiated without all of its data. It is therefore a shadow of itself, or a *ghost*. Ghost objects know how to load the rest of their data using what they were given at init. This is what allows you to write `kalturi = Summoner(name="Kalturi", region="NA")` followed by `kalturi.level`. The latter will trigger a call to the data pipeline (discussed below) to pull the rest of the data for `kalturi` by using `kalturi.name`.

Most top-level objects in Cass are ghost objects and therefore know how to load their own data.

For developers who are interested, the implementation for ghost objects can be found in our `meraki-commons` repository on GitHub.

### Data Pipeline

The data pipeline is the series of caches, databases, and data sources (such as the Riot API) that both provide and store data. Data sources provide data, while data sinks store data; we call both of these “data stores”. Some parts of the data pipeline are only data sources (for example, the Riot API), while others are both data sources and data sinks (for example, caches and databases). The data pipeline is a list of data stores, where the order the data stores specifies how data is pulled and stored (see the next paragraph). Usually faster data stores go at the beginning of the data pipeline.

When data is queried, a query dictionary is constructed containing the information needed to uniquely identify an object in a data source (e.g. a `region` and `summoner.id` are required when querying for `Summoner` objects). This query is passed up the data pipeline through the data sources, and at each data source the data pipeline asks if that source can supply the requested object. If the source can supply the object (for example, if the object is in the database, or if the Riot API can send the object/data), it is returned. If the source does not supply the object, the next data source in the pipeline is queried. If no data source can provide an object for the query, a `datapipelines.NotFoundError` is thrown.

After an object is returned by a data source, the object gets passed back down the pipeline. Any data sinks along the way store the object that was returned by the data source. In this way, the cache (which should be at the front of the data pipeline) will store any object that a database or the Riot API returned.

A data pipeline containing an in-memory cache and the Riot API is created by default. The pipeline can be accessed via `settings.pipeline`, although users should rarely if ever touch this object after it has been instantiated.

See [Data Pipeline](#) for more details.

### Searchable Containers

Most lists, dictionaries, and sets (all of which are containers) can be searched by most values that make sense. For example, the below line of code finds the first game in which Teemo was played in the match history of the specified summoner (note that all participants in the match are searched, not just the specific summoner for whom the match history was pulled).

```
a_teemo_game = Summoner(name="Dabblegamer", region="NA").match_history["Teemo"]
```

You can also search using objects rather than strings:

```
all_champions = Champions(region="NA")
teemo = all_champions["Teemo"]
a_teemo_game = Summoner(name="Dabblegamer", region="NA").match_history[teemo]
```

All matches in a summoner's match history where Teemo was in the game can be found by using `.search` rather than the `[...]` syntax:

```
# We will truncate the summoner's match history so we don't pull thousands of matches
match_history = Summoner(name="Dabblegamer", region="NA").match_history(begin_time=Patch.
    ↪from_str("9.1", region="NA").start)
all_teemo_games = match_history.search("Teemo")
```

You can also index on items in a match. For example:

```
...match_history["Sightstone"]
```

will find a game in the summoner's match history where someone ended the game with a Sightstone (or Ruby Sightstone) in their inventory.

Below is a final (very convenient) snippet that allows you to get your participant in a match:

```
me = Summoner(name="Kalturi", region="NA")
match = me.match_history[0]
champion_played = match.participants[me].champion
```

Searchable containers are extremely powerful and are one of the reasons why writing code using Cass is both fun and intuitive.

## Match Histories Work Slightly Differently

The match history of a summoner is handled slightly differently than most objects in Cass. Most importantly, it is not Cached or stored in databases we create. This is largely because the logic for doing so is non-trivial, and we haven't implemented it yet – although we hope to. Therefore match histories are requested from the Riot API every time the method is called. You are encouraged to cache the results yourself if you wish.

Match histories are also lazily loaded.

### 4.2.5 Data Pipeline

The data pipeline is a fundamental piece of Cass. It controls the flow of data into and out of an in-memory cache, your databases, the Riot API, and any other data sources/sinks you provide.

The data pipeline consists of a list of `DataSources` and `DataSinks`. A `DataSource` is any entity that *provides* data (for example, the Riot API and databases are both data sources). A `DataSink` is any entity that *stores* data (databases are also data sinks). Any entity that is a data sink will almost certainly be a data source as well. We refer to an entity that is both a data source and data sink as a `data store`.

The data sources and sinks are *ordered* in the data pipeline, and their order determines the order in which data is requested. Generally speaking, slower data stores / sinks should go towards the end of the pipeline.

For example, if your data pipeline consists of a cache, a database, and the Riot API (in that order), when you ask for a Champion Cassiopeia will first look in the cache, then in your database, then in the Riot API. If the data is found in the cache, it will be returned and the database and Riot API will not be queried. Similarly, if the data is found in the database, the Riot API will not be queried.

After data is found in a data source, the data propagates back down the data pipeline from whence it came. Any data sink encountered along the way will store that data. So, continuing the above example, if you asked for a Champion and it was provided by the Riot API, the champion data would be stored in your database, then stored in the cache. A data sink will only store data that it “accepts”. Cass's built-in data sinks accept all of Cass's data types.

Each data sink has expiration periods defined for each type of data it accepts. When data is put into a data sink, a clock starts ticking (metaphorically, programmatically this is handled differently). When that clock finishes, the data is expelled from the data sink. Static data should have an infinite expiration period (because it is stored per-version, and the static data for a given version never changes). Other types like `CurrentMatch` might have very short expiration periods. Each data sink defines its own default expiration periods, which are documented under the specific data sinks below.

A few notes: 1) Users can force all expired objects in data sinks to be removed using `settings.pipeline.expire()`. 2) Individual data sinks handle their own expirations, so if you write a database, you must decide how to handle expirations for data in your database.

Below is an example (which uses more datastores than Cass uses by default):

```
{  
    "pipeline": {  
        "Cache": {},  
  
        "SimpleKVDiskStore": {  
            "package": "cassiopeia_diskstore"  
        },  
  
        "DDragon": {},  
  
        "RiotAPI": {  
            "api_key": "RIOT_API_KEY"  
        },  
  
        "ChampionGG": {  
            "package": "cassiopeia_championgg",  
            "api_key": "CHAMPIONGG_KEY"  # See api.champion.gg  
        }  
    }  
}
```

In brief, this means that the sequence for looking for data will be: 1) Look in the cache, 2) look in our disk-based database, 3) if it's static data, get it from data dragon, 4) pull the data from the Riot API, 5) pull the data from ChampionGG.

### Defining Components in your Settings

The components of the data pipeline are defined explicitly below, and you can choose which you want to use by setting the `"pipelines"` attribute in your settings. By default, Cass uses the in-memory cache, data dragon, and the Riot API.

Each component has it's own set of parameters, also described below.

*Settings* has an example data pipeline you can use in your settings if you want to modify the defaults.

## Components

### In-Memory Cache

The in-memory cache, simply called the cache, is a data store and provides fast read / write storage of data. It is used by including Cache in the data pipeline settings. If you are constantly creating the same data over and over, the cache is extremely useful. However, if you only using pulling a given piece of data once, it is likely unnecessary.

The cache should be the first element in your pipeline.

It takes one optional parameter (called `expirations`), which is a mapping of expiration times (in seconds or `datetime.timedelta` if set programmatically) for each data type stored in the cache. Valid type names and their defaults are below (a value of -1 means “do not expire” and 0 means “do not store in the data sink”):

```
ChampionRotationData: datetime.timedelta(hours=6),
Realms: datetime.timedelta(hours=6),
Versions: datetime.timedelta(hours=6),
Champion: datetime.timedelta(days=20),
Rune: datetime.timedelta(days=20),
Item: datetime.timedelta(days=20),
SummonerSpell: datetime.timedelta(days=20),
Map: datetime.timedelta(days=20),
ProfileIcon: datetime.timedelta(days=20),
Locales: datetime.timedelta(days=20),
LanguageStrings: datetime.timedelta(days=20),
SummonerSpells: datetime.timedelta(days=20),
Items: datetime.timedelta(days=20),
Champions: datetime.timedelta(days=20),
Runes: datetime.timedelta(days=20),
Maps: datetime.timedelta(days=20),
ProfileIcons: datetime.timedelta(days=20),
ChampionMastery: datetime.timedelta(days=7),
ChampionMasteries: datetime.timedelta(days=7),
LeagueSummonerEntries: datetime.timedelta(hours=6),
League: datetime.timedelta(hours=6),
ChallengerLeague: datetime.timedelta(hours=6),
MasterLeague: datetime.timedelta(hours=6),
Match: datetime.timedelta(days=3),
Timeline: datetime.timedelta(days=1),
Summoner: datetime.timedelta(days=1),
ShardStatus: datetime.timedelta(hours=1),
CurrentMatch: datetime.timedelta(hours=0.5),
FeaturedMatches: datetime.timedelta(hours=0.5)
```

TODO: The cache currently does not automatically expire its data, so it's possible to run out of memory. To prevent this, users can trigger an expiration of all data or all data of one type by using the method `settings.pipeline.expire`. We will fix this so that the cache does automatically expire it's data, but we haven't gotten to it yet. Using the `expire` method is a temporary workaround.

## Data Dragon

Data Dragon is a data source and provides all of Cass's static data. This is largely due to the static data rate limits enforced by the Riot API. If you are testing your app and running it repeatedly without a database, you will need to continuously request the static data and will quickly hit the Riot API's rate limits. Data Dragon provides exactly the same data without some of the niceties that the Riot API provides.

Data Dragon should therefore come before the Riot API in your pipeline, but likely after your databases.

It takes no parameters (i.e. {}).

## Riot API

Hopefully you already know what this is. It's where you're planning on getting your data, and it's a data source. It should come after your data bases, and will likely always be the last thing in your data pipeline.

This component can have complicated settings, so see [Settings](#) for its parameters.

## Kernel

Cassiopeia can query a proxy server that mirrors Riot API endpoints. An example of such server is Kernel.

To configure the address and ports of the proxy, use the following configuration within your pipeline:

```
{  
    "pipeline": {  
        ...  
        "Kernel": {  
            "server_url": "http://localhost",  
            "port": 80  
        }  
        ...  
    }  
}
```

## Simple Disk Database

This is a simple filesystem database, and is therefore both a data source and data sink. It is not provided by Cass by default, and needs to be installed separately. See [Plugins](#) for more information.

## SQLAlchemy Database Support

This is a database system that supports all databases that [SQLAlchemy](#) supports. It is not provided by Cass by default, and needs to be installed separately. See [Plugins](#) for more information.

## ChampionGG

The ChampionGG plugin has its own data source if it is included. See [Plugins](#).

### Unloaded Ghost Store

As a user, it's very likely that you don't need to worry about what this store does. Cass automatically puts this store in your datapipeline.

The `UnloadedGhostStore` provides unloaded ghost objects to the rest of Cass when a new ghost object is created. This allows us to have a single location where all top-level objects are created, which alleviates some complicated issues that crop up when caching core objects and using ghost loading. In general, it should always be in your pipeline.

If you wish to override how Cass inserts it into your pipeline, you can include it in your pipeline and Cass won't insert it automatically. Normally, it should go immediately after the cache, and if you are not using a cache, it should be the first element in the data pipeline.

## 4.2.6 Plugins

Plugins monkeypatch Cass to provide modified or additional functionality. They are listed below.

The plugins for Cass are stored in two different repositories: `cassiopeia-plugins` and `cassiopeia-datastores`. `cassiopeia-plugins` contains functionality that modify the behavior of Cass's objects, while `cassiopeia-datastores` provides additional datastores (such as databases). Both of these are called "plugins" in this documentation.

Plugins can be added to Cass by downloading the appropriate plugin and putting it on your `PYTHONPATH` environment variable. Then, in your settings file, you specify the name of the module for that plugin (using the `package` keyword) as if you were directly importing it into your project. The name of the package specifies the data store that that will be loaded from that package and put on the pipeline.

## ChampionGG

Install by running `pip install cassiopeia-championgg`.

The ChampionGG plugin pulls data from the [champion.gg api](#). This data is accessible via the `Champion.championgg` attribute.

To enable this plugin, add the following to your settings' data pipeline:

```
"pipeline": {
    ...
    "ChampionGG": {
        "package": "cassiopeia_championgg",
        "api_key": "CHAMPIONGG_KEY"
    },
    ...
}
```

where "`CHAMPIONGG_KEY`" is your champion.gg API key or an environment variable that contains it.

## Simple KV Disk Store

Install by running `pip install cassiopeia-diskstore`.

This plugin provides a disk-database. It is especially useful for staticdata, which never changes. It works for all data types except `MatchHistory`.

To enable this plugin, add the following to your settings' data pipeline between the Cache and DDragon stores:

```
"pipeline": {  
    ...,  
    "SimpleKVDiskStore": {  
        "package": "cassiopeia_diskstore",  
        "path": "/absolute/path/to/store/data/"  
    },  
    ...  
}
```

The "path" parameter specifies a directory path where the data will be stored. There is also another optional "expirations" parameter that is left out of the above example for clarity. The "expirations" parameter is a mapping of type names to expiration periods analogous to those for the cache. The allowed type names and default values are below (a value of -1 means "do not expire" and 0 means "do not store in the data sink"):

```
RealmDto: datetime.timedelta(hours=6),  
VersionListDto: datetime.timedelta(hours=6),  
ChampionDto: -1,  
ChampionListDto: -1,  
RuneDto: -1,  
RuneListDto: -1,  
ItemDto: -1,  
ItemListDto: -1,  
SummonerSpellDto: -1,  
SummonerSpellListDto: -1,  
MapDto: -1,  
MapListDto: -1,  
ProfileIconDetailsDto: -1,  
ProfileIconDataDto: -1,  
LanguagesDto: -1,  
LanguageStringsDto: -1,  
ChampionRotationDto: datetime.timedelta(days=1),  
ChampionMasteryDto: datetime.timedelta(days=7),  
ChampionMasteryListDto: datetime.timedelta(days=7),  
ShardStatusDto: datetime.timedelta(hours=1),
```

Some objects share the same expiration time: `FeaturedGamesDto` shares expiration of `CurrentGameInfoDto`, `ChallengerLeagueListDto` and `MasterLeagueListDto` share expiration of `LeagueListDto`, `ChampionMasteryListDto` shares expiration of `ChampionMasteryDto`, and `ChampionListDto` shares expiration of `ChampionDto`. Only the latter in each category need to be set.

This store only supports the above types (for now).

## 4.2.7 Contributing

Contributions are welcome! If you have idea or opinions on how things can be improved, don't hesitate to let us know by posting an issue on GitHub or @ing us on the Meraki or Riot API Discord channel. And we always want to hear from our users, even (especially) if it's just letting us know how you are using Cass.

As a user you get to ignore the details and just use the features of Cass. But as a developer you get to dive into the nitty-gritty and pick apart the implementation that makes everything work. If you don't want to dive too deep, you can likely contribute even without knowing all the details. You can read more about how Cass works [here](#), and you can find opportunities to help by looking at our issues that are tagged with `help-wanted` as well as looking at the list below.

If you have an idea but aren't sure about it, feel free to @ us on Discord and we can chat.

### Things we need help with!

- We current don't support the tournament API but need to.
- Very few methods / properties have doc strings. While not glorious, it is an incredibly helpful thing to do and you will quickly learning all the pieces of Cass.
- In the previous version of Cass, we used regex to pull item stats from tooltips, because the static data is missing a significant number of stats. The old code can be found [here](#) and needs to be ported to this version of Cass.
- We want to support Redis and Mongo databases in addition to those we already support. To do so, new data-sources should be added (along side the Riot API and the in-memory cache) to support these databases. This functionality should be added to our [cassiopeia-datastores](#) repository.
- We have some very basic tests in place, but a thorough testing of all attributes of all objects would be extremely helpful.
- Some data from the [champion.gg api](#) is available through Cass (via the `Champion` object). The remaining data should be added as well. You can find the relevant code in the `plugins/championgg` directory.
- Implement better logging.

## 4.3 Top Level APIs

- *Settings*
- *Data and Enums*
- *Champions*
- *Champion Masteries*
- *Items*
- *Language Strings*
- *Leagues*
- *Locales*
- *Maps*
- *Matches*
- *Patch*
- *Profile Icons*
- *Realms*

- *Runes*
- *Status*
- *Spectator*
- *Summoners*
- *Summoner Spells*
- *Versions*

---

**CHAPTER  
FIVE**

---

**INDEX AND SEARCH**

- genindex



## PYTHON MODULE INDEX

### C

cassiopeia.data, 9



# INDEX

## A

ability\_power (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
account\_id (*cassiopeia.Summoner attribute*), 59  
after\_id (*cassiopeia.core.match.Event attribute*), 49  
all\_random\_summoners\_rift (*cassiopeia.data.GameMode attribute*), 9  
all\_random\_summoners\_rift (*cassiopeia.data.Queue attribute*), 12  
all\_random\_urf (*cassiopeia.data.Queue attribute*), 12  
all\_random\_urf\_snow (*cassiopeia.data.GameMode attribute*), 9  
all\_random\_urf\_snow (*cassiopeia.data.Queue attribute*), 12  
ally\_tips (*cassiopeia.Champion attribute*), 19  
alternative\_images (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 23  
alternative\_images (*cassiopeia.SummonerSpell attribute*), 61  
americas (*cassiopeia.data.Continent attribute*), 9  
apex (*cassiopeia.data.Position attribute*), 12  
append() (*cassiopeia.ChampionMasterys method*), 27  
append() (*cassiopeia.Champions method*), 18  
append() (*cassiopeia.core.league.LeagueEntries method*), 36  
append() (*cassiopeia.core.league.LeagueSummonerEntries method*), 34  
append() (*cassiopeia.core.match.MatchHistory method*), 41  
append() (*cassiopeia.FeaturedMatches method*), 56  
append() (*cassiopeia.Items method*), 29  
append() (*cassiopeia.Locales method*), 38  
append() (*cassiopeia.Maps method*), 39  
append() (*cassiopeia.ProfileIcons method*), 51  
append() (*cassiopeia.Runes method*), 53  
append() (*cassiopeia.SummonerSpells method*), 59  
append() (*cassiopeia.Versions method*), 63  
aram (*cassiopeia.data.GameMode attribute*), 9  
aram (*cassiopeia.data.Queue attribute*), 12  
aram\_butchers\_bridge (*cassiopeia.data.Queue attribute*), 12

armor (*cassiopeia.core.staticdata.championStats attribute*), 21  
armor (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
armor\_per\_level (*cassiopeia.core.staticdata.championStats attribute*), 21  
ascended\_type (*cassiopeia.core.match.Event attribute*), 49  
ascension (*cassiopeia.data.GameMode attribute*), 9  
ascension (*cassiopeia.data.Queue attribute*), 12  
asia (*cassiopeia.data.Continent attribute*), 9  
assassinate (*cassiopeia.data.GameMode attribute*), 9  
assisting\_participants (*cassiopeia.core.match.Event attribute*), 49  
assists (*cassiopeia.core.match.ParticipantState attribute*), 50  
assists (*cassiopeia.core.match.ParticipantStats attribute*), 45  
assists (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26  
assists (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
attack (*cassiopeia.core.staticdata.champion.Info attribute*), 21  
attack\_damage (*cassiopeia.core.staticdata.championStats attribute*), 21  
attack\_damage (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
attack\_damage\_per\_level (*cassiopeia.core.staticdata.championStats attribute*), 21  
attack\_range (*cassiopeia.core.staticdata.championStats attribute*), 21  
attack\_speed (*cassiopeia.core.staticdata.championStats attribute*), 21  
attack\_speed (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32

## B

ban\_rate (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25

**ban\_rates** (*cassiopeia.Champion* attribute), 19  
**bans** (*cassiopeia.core.match.Team* attribute), 43  
**bans** (*cassiopeia.core.spectator.Team* attribute), 58  
**baron\_kills** (*cassiopeia.core.match.ParticipantStats* attribute), 45  
**baron\_kills** (*cassiopeia.core.match.Team* attribute), 43  
**BASE** (*cassiopeia.data.Tower* attribute), 18  
**before\_id** (*cassiopeia.core.match.Event* attribute), 49  
**black\_market\_brawlers** (*cassiopeia.data.Queue* attribute), 12  
**blind\_fives** (*cassiopeia.data.Queue* attribute), 12  
**blind\_threes** (*cassiopeia.data.Queue* attribute), 12  
**block** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**blood\_hunt\_assassin** (*cassiopeia.data.Queue* attribute), 12  
**blood\_well** (*cassiopeia.data.Resource* attribute), 15  
**blue** (*cassiopeia.data.Side* attribute), 16  
**blue\_team** (*cassiopeia.core.spectator.CurrentMatch* attribute), 57  
**blue\_team** (*cassiopeia.Match* attribute), 42  
**blurb** (*cassiopeia.Champion* attribute), 19  
**bot\_lane** (*cassiopeia.data.Lane* attribute), 10  
**bot\_lane\_blue** (*cassiopeia.data.SummonersRiftArea* attribute), 16  
**bot\_lane\_purple** (*cassiopeia.data.SummonersRiftArea* attribute), 16  
**bot\_lane\_red** (*cassiopeia.data.SummonersRiftArea* attribute), 16  
**bottom** (*cassiopeia.data.Position* attribute), 12  
**bounty\_level** (*cassiopeia.core.match.ParticipantStats* attribute), 45  
**brazil** (*cassiopeia.data.Platform* attribute), 11  
**brazil** (*cassiopeia.data.Region* attribute), 15  
**bronze** (*cassiopeia.data.Tier* attribute), 17  
**building\_type** (*cassiopeia.core.match.Event* attribute), 49  
**builds\_from** (*cassiopeia.Item* attribute), 30  
**builds\_into** (*cassiopeia.Item* attribute), 30

**C**

**captured\_point** (*cassiopeia.core.match.Event* attribute), 49  
**cassiopeia.data**  
  **module**, 9  
**cdn** (*cassiopeia.Realms* attribute), 53  
**challenger** (*cassiopeia.data.Tier* attribute), 17  
**ChallengerLeague** (class in *cassiopeia.core*), 33  
**champion** (*cassiopeia.ChampionMastery* attribute), 28  
**champion** (*cassiopeia.core.match.Participant* attribute), 44  
**champion** (*cassiopeia.core.spectator.Participant* attribute), 58  
**champion** (*cassiopeia.Item* attribute), 30

**champion** (*cassiopeia\_championgg.core.ChampionGGMatchupStats* attribute), 26  
**Champion** (class in *cassiopeia*), 19  
**champion\_assists** (cas-  
  *siopeia.core.match.ParticipantTimeline* at-  
  tribute), 47  
**champion\_deaths** (cas-  
  *siopeia.core.match.ParticipantTimeline* at-  
  tribute), 47  
**champion\_experience** (cas-  
  *siopeia.core.match.ParticipantStats* attribute), 45  
**champion\_key** (*cassiopeia.core.staticdata.champion.Skin* attribute), 22  
**champion\_kills** (cas-  
  *siopeia.core.match.ParticipantTimeline* at-  
  tribute), 48  
**champion\_masteries** (*cassiopeia.Summoner* attribute), 59  
**champion\_transform** (cas-  
  *siopeia.core.match.ParticipantStats* attribute), 45  
**championgg\_metadata** (cas-  
  *siopeia\_championgg.core.ChampionGGStats* attribute), 25  
**ChampionGGMatchup** (class in *cassiopeia\_championgg.core*), 25  
**ChampionGGMatchups** (class in *cassiopeia\_championgg.core*), 25  
**ChampionGGMatchupStats** (class in *cassiopeia\_championgg.core*), 26  
**ChampionGGStats** (class in *cassiopeia\_championgg.core*), 25  
**ChampionMasteries** (class in *cassiopeia*), 27  
**ChampionMastery** (class in *cassiopeia*), 28  
**Champions** (class in *cassiopeia*), 18  
**ChampionSpell** (class in *cassiopeia.core.staticdata.champion*), 23  
**chest\_granted** (*cassiopeia.ChampionMastery* attribute), 28  
**clash** (*cassiopeia.data.Queue* attribute), 12  
**classic** (*cassiopeia.data.GameMode* attribute), 9  
**clear()** (*cassiopeia.ChampionMasteries* method), 27  
**clear()** (*cassiopeia.Champions* method), 18  
**clear()** (*cassiopeia.core.league.LeagueEntries* method), 36  
**clear()** (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
**clear()** (*cassiopeia.core.match.MatchHistory* method), 41  
**clear()** (*cassiopeia.FeaturedMatches* method), 56  
**clear()** (*cassiopeia.Items* method), 29  
**clear()** (*cassiopeia.Locales* method), 38  
**clear()** (*cassiopeia.Maps* method), 39

**clear()** (*cassiopeia.ProfileIcons* method), 51  
**clear()** (*cassiopeia.Runes* method), 53  
**clear()** (*cassiopeia.SummonerSpells* method), 59  
**clear()** (*cassiopeia.Versions* method), 63  
**clear\_sinks()** (*cassiopeia.\_configuration.settings.Settings* method), 8  
**client\_refresh\_interval** (*cassiopeia.FeaturedMatches* attribute), 56  
**coefficients** (*cassiopeia.core.staticdata.champion.Spell* attribute), 24  
**coefficients** (*cassiopeia.core.staticdata.summonerspell.Spell* attribute), 62  
**consumables\_purchased** (*cassiopeia.core.match.ParticipantStats* attribute), 45  
**consume\_on\_full** (*cassiopeia.Item* attribute), 30  
**consumed** (*cassiopeia.Item* attribute), 31  
**contains()** (*cassiopeia.ChampionMasteries* method), 27  
**contains()** (*cassiopeia.Champions* method), 18  
**contains()** (*cassiopeia.core.league.LeagueEntries* method), 36  
**contains()** (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
**contains()** (*cassiopeia.core.match.MatchHistory* method), 41  
**contains()** (*cassiopeia.FeaturedMatches* method), 56  
**contains()** (*cassiopeia.Items* method), 29  
**contains()** (*cassiopeia.Locales* method), 38  
**contains()** (*cassiopeia.Maps* method), 39  
**contains()** (*cassiopeia.ProfileIcons* method), 51  
**contains()** (*cassiopeia.Runes* method), 53  
**contains()** (*cassiopeia.SummonerSpells* method), 60  
**contains()** (*cassiopeia.Versions* method), 63  
**continent** (*cassiopeia.core.match.Timeline* attribute), 48  
**continent** (*cassiopeia.data.Platform* attribute), 11  
**continent** (*cassiopeia.data.Region* attribute), 15  
**continent** (*cassiopeia.Match* attribute), 43  
**Continent** (class in *cassiopeia.data*), 9  
**continent()** (*cassiopeia.core.match.MatchHistory* method), 41  
**cooldowns** (*cassiopeia.core.staticdata.champion.Champion* attribute), 23  
**cooldowns** (*cassiopeia.SummonerSpell* attribute), 61  
**coop\_ai\_beginner\_fives** (*cassiopeia.data.Queue* attribute), 12  
**coop\_ai\_beginner\_threes** (*cassiopeia.data.Queue* attribute), 12  
**coop\_ai\_intermediate\_fives** (*cassiopeia.data.Queue* attribute), 12  
**coop\_ai\_intermediate\_threes** (*cassiopeia.data.Queue* attribute), 12  
**coop\_ai\_intro\_fives** (*cassiopeia.data.Queue* attribute), 12  
**coop\_ai\_intro\_threes** (*cassiopeia.data.Queue* attribute), 12  
**copy()** (*cassiopeia.ChampionMasteries* method), 27  
**copy()** (*cassiopeia.Champions* method), 18  
**copy()** (*cassiopeia.core.league.LeagueEntries* method), 36  
**copy()** (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
**copy()** (*cassiopeia.core.match.MatchHistory* method), 41  
**copy()** (*cassiopeia.FeaturedMatches* method), 56  
**copy()** (*cassiopeia.Items* method), 29  
**copy()** (*cassiopeia.Locales* method), 38  
**copy()** (*cassiopeia.Maps* method), 39  
**copy()** (*cassiopeia.ProfileIcons* method), 51  
**copy()** (*cassiopeia.Runes* method), 53  
**copy()** (*cassiopeia.SummonerSpells* method), 60  
**copy()** (*cassiopeia.Versions* method), 63  
**costs** (*cassiopeia.core.staticdata.champion.Champion* attribute), 23  
**costs** (*cassiopeia.SummonerSpell* attribute), 61  
**count** (*cassiopeia.core.match.MatchHistory* attribute), 41  
**count()** (*cassiopeia.ChampionMasteries* method), 27  
**count()** (*cassiopeia.Champions* method), 18  
**count()** (*cassiopeia.core.league.LeagueEntries* method), 36  
**count()** (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
**count()** (*cassiopeia.FeaturedMatches* method), 56  
**count()** (*cassiopeia.Items* method), 29  
**count()** (*cassiopeia.Locales* method), 38  
**count()** (*cassiopeia.Maps* method), 39  
**count()** (*cassiopeia.ProfileIcons* method), 51  
**count()** (*cassiopeia.Runes* method), 54  
**count()** (*cassiopeia.SummonerSpells* method), 60  
**count()** (*cassiopeia.Versions* method), 63  
**courage** (*cassiopeia.data.Resource* attribute), 15  
**creation** (*cassiopeia.core.spectator.CurrentMatch* attribute), 57  
**Spell** (*cassiopeia.Match* attribute), 43  
**creator\_id** (*cassiopeia.core.match.Event* attribute), 49  
**creep\_score** (*cassiopeia.core.match.ParticipantFrame* attribute), 48  
**creep\_score** (*cassiopeia.core.match.ParticipantState* attribute), 50  
**crimson\_rush** (*cassiopeia.data.Resource* attribute), 15  
**critical\_strike\_chance** (*cassiopeia.core.staticdata.champion.Stats* attribute), 21  
**critical\_strike\_chance** (*cassiopeia.core.staticdata.item.ItemStats* attribute)

tribute), 32

**critical\_strike\_chance\_per\_level** (*cassiopeia.core.staticdata.champion.Stats attribute*), 21

**critical\_strike\_damage** (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32

**css\_version** (*cassiopeia.Realms attribute*), 53

**cumulative\_timeline** (*cassiopeia.core.match.Participant attribute*), 44

**CumulativeTimeline** (*class in cassiopeia.core.match*), 50

**cunning** (*cassiopeia.data.MasteryTree attribute*), 11

**current\_gold** (*cassiopeia.core.match.ParticipantFrame attribute*), 48

**current\_gold** (*cassiopeia.core.match.ParticipantState attribute*), 50

**current\_match** (*cassiopeia.cassiopeia.Summoner attribute*), 56

**current\_match** (*cassiopeia.Summoner attribute*), 59

**CurrentMatch** (*class in cassiopeia.core.spectator*), 57

**custom** (*cassiopeia.data.GameType attribute*), 10

**custom** (*cassiopeia.data.Queue attribute*), 12

**D**

**damage\_composition** (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25

**damage\_dealt\_to\_buildings** (*cassiopeia.core.match.ParticipantStats attribute*), 45

**damage\_dealt\_to\_objectives** (*cassiopeia.core.match.ParticipantStats attribute*), 45

**damage\_dealt\_to\_turrets** (*cassiopeia.core.match.ParticipantStats attribute*), 45

**damage\_self\_mitigated** (*cassiopeia.core.match.ParticipantStats attribute*), 45

**dark\_star** (*cassiopeia.data.GameMode attribute*), 9

**dark\_star** (*cassiopeia.data.Queue attribute*), 13

**deaths** (*cassiopeia.core.match.ParticipantState attribute*), 50

**deaths** (*cassiopeia.core.match.ParticipantStats attribute*), 45

**deaths** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**deaths** (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25

**default\_locale** (*cassiopeia.data.Platform attribute*), 11

**default\_locale** (*cassiopeia.data.Region attribute*), 15

**defense** (*cassiopeia.core.staticdata.champion.Info attribute*), 21

**definitely\_not\_dominion** (*cassiopeia.data.Queue attribute*), 13

**delete()** (*cassiopeia.ChampionMasteries method*), 27

**delete()** (*cassiopeia.Champions method*), 18

**delete()** (*cassiopeia.core.league.LeagueEntries method*), 36

**delete()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 34

**delete()** (*cassiopeia.core.match.MatchHistory method*), 41

**delete()** (*cassiopeia.FeaturedMatches method*), 56

**delete()** (*cassiopeia.Items method*), 29

**delete()** (*cassiopeia.Locales method*), 38

**delete()** (*cassiopeia.Maps method*), 39

**delete()** (*cassiopeia.ProfileIcons method*), 51

**delete()** (*cassiopeia.Runes method*), 54

**delete()** (*cassiopeia.SummonerSpells method*), 60

**delete()** (*cassiopeia.Versions method*), 63

**delta\_assists** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_deaths** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_gold\_earned** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_killing\_sprees** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_kills** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_minions\_killed** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_neutral\_minions\_killed\_team\_jungle** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_ten\_to\_twenty** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_thirty\_to\_end** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_total\_damage\_dealt\_to\_champions** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_twenty\_to\_thirty** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_weighted\_score** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

**delta\_wins** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

*attribute), 26*

`delta_zero_to_ten` (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26

`deprecated_all_random_urf` (*cassiopeia.data.Queue attribute*), 13

`deprecated_aram` (*cassiopeia.data.Queue attribute*), 13

`deprecated_blind_dominion` (*cassiopeia.data.Queue attribute*), 13

`deprecated_blind_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_blind_threes` (*cassiopeia.data.Queue attribute*), 13

`deprecated_coop_ai_beginner_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_coop_ai_dominion` (*cassiopeia.data.Queue attribute*), 13

`deprecated_coop_ai_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_coop_ai_intermediate_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_coop_ai_intro_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_coop_ai_threes` (*cassiopeia.data.Queue attribute*), 13

`deprecated_doom_bots_rank_1` (*cassiopeia.data.Queue attribute*), 13

`deprecated_doom_bots_rank_2` (*cassiopeia.data.Queue attribute*), 13

`deprecated_doom_bots_rank_5` (*cassiopeia.data.Queue attribute*), 13

`deprecated_draft_dominion` (*cassiopeia.data.Queue attribute*), 13

`deprecated_draft_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_nexus_bltiz` (*cassiopeia.data.Queue attribute*), 13

`deprecated_nexus_siege` (*cassiopeia.data.Queue attribute*), 13

`deprecated_poro_king` (*cassiopeia.data.Queue attribute*), 13

`deprecated_ranked_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_ranked_flex_threes` (*cassiopeia.data.Queue attribute*), 13

`deprecated_ranked_premade_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_ranked_premade_threes` (*cassiopeia.data.Queue attribute*), 13

`deprecated_ranked_solo_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_ranked_team_fives` (*cassiopeia.data.Queue attribute*), 13

`deprecated_ranked_team_threes` (*cassiopeia.data.Queue attribute*), 13

`siopeia.data.Queue attribute`), 13

`deprecated_team_builder_fives` (*cassiopeia.data.Queue attribute*), 13

`description` (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 23

`description` (*cassiopeia.core.staticdata.champion.Passive attribute*), 22

`description` (*cassiopeia.Item attribute*), 31

`description` (*cassiopeia.SummonerSpell attribute*), 61

`diamond` (*cassiopeia.data.Tier attribute*), 17

`difficulty` (*cassiopeia.core.staticdata.champion.Info attribute*), 21

`division` (*cassiopeia.core.league.LeagueEntries attribute*), 36

`division` (*cassiopeia.core.league.LeagueEntry attribute*), 37

`Division` (*class in cassiopeia.data*), 9

`dodge` (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32

`domination` (*cassiopeia.Runes attribute*), 54

`dominion` (*cassiopeia.data.GameMode attribute*), 9

`dominion_score` (*cassiopeia.core.match.ParticipantFrame attribute*), 48

`dominion_score` (*cassiopeia.core.match.ParticipantState attribute*), 50

`dominion_score` (*cassiopeia.core.match.Team attribute*), 43

`doom_bots` (*cassiopeia.data.GameMode attribute*), 9

`doom_bots` (*cassiopeia.data.Queue attribute*), 13

`doom_bots_difficult` (*cassiopeia.data.Queue attribute*), 13

`double_kills` (*cassiopeia.core.match.ParticipantStats attribute*), 45

`dragon_kills` (*cassiopeia.core.match.ParticipantStats attribute*), 45

`dragon_kills` (*cassiopeia.core.match.Team attribute*), 43

`duo` (*cassiopeia.data.Role attribute*), 16

`duo_carry` (*cassiopeia.data.Role attribute*), 16

`duo_support` (*cassiopeia.data.Role attribute*), 16

`duration` (*cassiopeia.core.spectator.CurrentMatch attribute*), 57

`duration` (*cassiopeia.Match attribute*), 43

`dynamic` (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 24

`dynamic` (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 62

**E**

`E` (*cassiopeia.data.Key attribute*), 10

`effect` (*cassiopeia.Item attribute*), 31

effects (*cassiopeia.core.staticdata.champion.ChampionSpell* attribute), 23  
effects (*cassiopeia.SummonerSpell* attribute), 61  
effects\_by\_level (*cassiopeia.core.staticdata.champion.ChampionSpell* attribute), 24  
elo (*cassiopeia\_championgg.core.ChampionGGMatchup* attribute), 25  
elo (*cassiopeia\_championgg.core.ChampionGGStats* attribute), 25  
emerald (*cassiopeia.data.Tier* attribute), 17  
end (*cassiopeia.Patch* attribute), 50  
end\_time (*cassiopeia.core.match.MatchHistory* attribute), 41  
ended\_in\_early\_surrender (*cassiopeia.core.match.Participant* attribute), 44  
enemy (*cassiopeia\_championgg.core.ChampionGGMatchup* attribute), 25  
enemy\_team (*cassiopeia.core.match.Participant* attribute), 44  
enemy\_tips (*cassiopeia.Champion* attribute), 20  
energy (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
energy (*cassiopeia.data.Resource* attribute), 16  
energy\_regen (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
entries (*cassiopeia.core.ChallengerLeague* attribute), 33  
entries (*cassiopeia.core.league.League* attribute), 33  
entries (*cassiopeia.core.MasterLeague* attribute), 34  
enumerate() (*cassiopeia.ChampionMasteries* method), 27  
enumerate() (*cassiopeia.Champions* method), 18  
enumerate() (*cassiopeia.core.league.LeagueEntries* method), 36  
enumerate() (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
enumerate() (*cassiopeia.core.match.MatchHistory* method), 41  
enumerate() (*cassiopeia.FeaturedMatches* method), 56  
enumerate() (*cassiopeia.Items* method), 29  
enumerate() (*cassiopeia.Locales* method), 38  
enumerate() (*cassiopeia.Maps* method), 39  
enumerate() (*cassiopeia.ProfileIcons* method), 51  
enumerate() (*cassiopeia.Runes* method), 54  
enumerate() (*cassiopeia.SummonerSpells* method), 60  
enumerate() (*cassiopeia.Versions* method), 63  
europe (*cassiopeia.data.Continent* attribute), 9  
europe\_north\_east (*cassiopeia.data.Platform* attribute), 11  
europe\_north\_east (*cassiopeia.data.Region* attribute), 15  
europe\_west (*cassiopeia.data.Platform* attribute), 11  
europe\_west (*cassiopeia.data.Region* attribute), 15  
Event (class in *cassiopeia.core.match*), 48  
events (*cassiopeia.core.match.Frame* attribute), 48  
events (*cassiopeia.core.match.ParticipantTimeline* attribute), 48  
exists (*cassiopeia.core.spectator.CurrentMatch* attribute), 57  
exists (*cassiopeia.Match* attribute), 43  
exists (*cassiopeia.Summoner* attribute), 59  
experience (*cassiopeia.core.match.ParticipantFrame* attribute), 48  
experience (*cassiopeia.core.match.ParticipantState* attribute), 50  
expire\_sinks() (*cassiopeia.\_configuration.settings.Settings* method), 8  
extend() (*cassiopeia.ChampionMasteries* method), 27  
extend() (*cassiopeia.Champions* method), 18  
extend() (*cassiopeia.core.league.LeagueEntries* method), 36  
extend() (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
extend() (*cassiopeia.core.match.MatchHistory* method), 41  
extend() (*cassiopeia.FeaturedMatches* method), 56  
extend() (*cassiopeia.Items* method), 29  
extend() (*cassiopeia.Locales* method), 38  
extend() (*cassiopeia.Maps* method), 39  
extend() (*cassiopeia.ProfileIcons* method), 51  
extend() (*cassiopeia.Runes* method), 54  
extend() (*cassiopeia.SummonerSpells* method), 60  
extend() (*cassiopeia.Versions* method), 63

## F

FeaturedMatches (class in *cassiopeia*), 56  
ferocity (*cassiopeia.data.MasteryTree* attribute), 11  
ferocity (*cassiopeia.data.Resource* attribute), 16  
filter() (*cassiopeia.ChampionMasteries* method), 27  
filter() (*cassiopeia.Champions* method), 18  
filter() (*cassiopeia.core.league.LeagueEntries* method), 36  
filter() (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
filter() (*cassiopeia.core.match.MatchHistory* method), 41  
filter() (*cassiopeia.FeaturedMatches* method), 56  
filter() (*cassiopeia.Items* method), 29  
filter() (*cassiopeia.Locales* method), 38  
filter() (*cassiopeia.Maps* method), 39  
filter() (*cassiopeia.ProfileIcons* method), 51  
filter() (*cassiopeia.Runes* method), 54  
filter() (*cassiopeia.SummonerSpells* method), 60  
filter() (*cassiopeia.Versions* method), 63  
find() (*cassiopeia.ChampionMasteries* method), 27

**find()** (*cassiopeia.Champions* method), 18  
**find()** (*cassiopeia.core.league.LeagueEntries* method), 36  
**find()** (*cassiopeia.core.league.LeagueSummonerEntries* method), 34  
**find()** (*cassiopeia.core.match.MatchHistory* method), 42  
**find()** (*cassiopeia.FeaturedMatches* method), 57  
**find()** (*cassiopeia.Items* method), 29  
**find()** (*cassiopeia.Locales* method), 38  
**find()** (*cassiopeia.Maps* method), 39  
**find()** (*cassiopeia.ProfileIcons* method), 51  
**find()** (*cassiopeia.Runes* method), 54  
**find()** (*cassiopeia.SummonerSpells* method), 60  
**find()** (*cassiopeia.Versions* method), 63  
**first\_baron** (*cassiopeia.core.match.Team* attribute), 43  
**first\_blood** (*cassiopeia.core.match.Team* attribute), 44  
**first\_blood\_assist** (*cassiopeia.core.match.ParticipantStats* attribute), 45  
**first\_blood\_kill** (*cassiopeia.core.match.ParticipantStats* attribute), 45  
**first\_dragon** (*cassiopeia.core.match.Team* attribute), 44  
**first\_inhibitor** (*cassiopeia.core.match.Team* attribute), 44  
**first\_rift\_herald** (*cassiopeia.core.match.Team* attribute), 44  
**first\_tower** (*cassiopeia.core.match.Team* attribute), 44  
**first\_tower\_assist** (*cassiopeia.core.match.ParticipantStats* attribute), 45  
**first\_tower\_fallen** (*cassiopeia.core.match.Timeline* attribute), 48  
**first\_tower\_kill** (*cassiopeia.core.match.ParticipantStats* attribute), 45  
**fives** (*cassiopeia.core.league.LeagueSummonerEntries* attribute), 34  
**flex** (*cassiopeia.core.league.LeagueSummonerEntries* attribute), 34  
**flow** (*cassiopeia.data.Resource* attribute), 16  
**four** (*cassiopeia.data.Division* attribute), 9  
**Frame** (*class* in *cassiopeia.core.match*), 48  
**frame\_interval** (*cassiopeia.core.match.Timeline* attribute), 48  
**frames** (*cassiopeia.core.match.ParticipantTimeline* attribute), 48  
**frames** (*cassiopeia.core.match.Timeline* attribute), 48  
**free\_to\_play** (*cassiopeia.Champion* attribute), 20  
**free\_to\_play\_new\_players** (*cassiopeia.Champion* attribute), 20  
**fresh\_blood** (*cassiopeia.core.league.LeagueEntry* attribute), 37  
**from\_data()** (*cassiopeia.ChampionMasteries* class method), 27  
**from\_data()** (*cassiopeia.Champions* class method), 18  
**from\_data()** (*cassiopeia.core.league.LeagueEntries* class method), 36  
**from\_data()** (*cassiopeia.core.league.LeagueEntry* class method), 37  
**from\_data()** (*cassiopeia.core.league.LeagueSummonerEntries* class method), 34  
**from\_data()** (*cassiopeia.core.match.MatchHistory* class method), 42  
**from\_data()** (*cassiopeia.core.match.Participant* class method), 44  
**from\_data()** (*cassiopeia.core.match.ParticipantStats* class method), 45  
**from\_data()** (*cassiopeia.core.match.ParticipantTimeline* class method), 48  
**from\_data()** (*cassiopeia.core.match.Team* class method), 44  
**from\_data()** (*cassiopeia.core.spectator.CurrentMatch* class method), 58  
**from\_data()** (*cassiopeia.core.spectator.Participant* class method), 58  
**from\_data()** (*cassiopeia.core.spectator.Team* class method), 58  
**from\_data()** (*cassiopeia.core.staticdata.champion.ItemSet* class method), 23  
**from\_data()** (*cassiopeia.core.staticdata.champion.RecommendedItems* class method), 23  
**from\_data()** (*cassiopeia.FeaturedMatches* class method), 57  
**from\_data()** (*cassiopeia.Items* class method), 29  
**from\_data()** (*cassiopeia.Locales* class method), 38  
**from\_data()** (*cassiopeia.Maps* class method), 39  
**from\_data()** (*cassiopeia.ProfileIcons* class method), 51  
**from\_data()** (*cassiopeia.Runes* class method), 54  
**from\_data()** (*cassiopeia.SummonerSpells* class method), 60  
**from\_data()** (*cassiopeia.Versions* class method), 63  
**from\_date()** (*cassiopeia.Patch* class method), 50  
**from\_generator()** (*cassiopeia.ChampionMasteries* class method), 27  
**from\_generator()** (*cassiopeia.Champions* class method), 18  
**from\_generator()** (*cassiopeia.core.league.LeagueEntries* class method), 36  
**from\_generator()** (*cassiopeia.core.league.LeagueSummonerEntries* class method), 34  
**from\_generator()** (*cassiopeia.core.match.MatchHistory* class method), 42

from\_generator() (*cassiopeia.FEATURED\_MATCHES class method*), 57  
from\_generator() (*cassiopeia.ITEMS class method*), 29  
from\_generator() (*cassiopeia.LOCALES class method*), 38  
from\_generator() (*cassiopeia.MAPS class method*), 40  
from\_generator() (*cassiopeia.PROFILE\_ICONS class method*), 51  
from\_generator() (*cassiopeia.RUNES class method*), 54  
from\_generator() (*cassiopeia.SUMMONER\_SPELLS class method*), 60  
from\_generator() (*cassiopeia.VERSIONS class method*), 63  
from\_id() (*cassiopeia.data.QUEUE method*), 14  
from\_league\_naming\_scheme() (*cassiopeia.data.POSITION method*), 12  
from\_match\_naming\_scheme() (*cassiopeia.data.LANE method*), 10  
from\_match\_naming\_scheme() (*cassiopeia.data.ROLE method*), 16  
from\_match\_reference() (*cassiopeia.MATCH class method*), 43  
from\_platform() (*cassiopeia.data.REGION static method*), 15  
from\_position() (*cassiopeia.data.SUMMONERS\_RIFT\_AREA static method*), 16  
from\_region() (*cassiopeia.data.PLATFORM static method*), 11  
from\_str() (*cassiopeia.PATCH class method*), 50  
fury (*cassiopeia.data.RESOURCE attribute*), 16

**G**

game\_type (*cassiopeia.MATCH attribute*), 43  
GameMode (*class in cassiopeia.data*), 9  
games\_played (*cassiopeia\_championgg.core.CHAMPIONGG\_STATS attribute*), 25  
GameType (*class in cassiopeia.data*), 10  
get\_challenger\_league() (*cassiopeia.CASSIOPEIA method*), 33  
get\_champion() (*cassiopeia.CASSIOPEIA method*), 18  
get\_champion\_masteryies() (*cassiopeia.CASSIOPEIA method*), 27  
get\_champion\_mastery() (*cassiopeia.CASSIOPEIA method*), 27  
get\_champions() (*cassiopeia.CASSIOPEIA method*), 18  
get\_current\_match() (*cassiopeia.CASSIOPEIA method*), 56  
get\_featured\_matches() (*cassiopeia.CASSIOPEIA method*), 56  
get\_items() (*cassiopeia.CASSIOPEIA method*), 29  
get\_lane() (*cassiopeia.data.SUMMONERS\_RIFT\_AREA method*), 16  
get\_language\_strings() (*cassiopeia.CASSIOPEIA method*), 33  
get\_locales() (*cassiopeia.CASSIOPEIA method*), 38  
get\_maps() (*cassiopeia.CASSIOPEIA method*), 39  
get\_master\_league() (*cassiopeia.CASSIOPEIA method*), 33  
get\_match() (*cassiopeia.CASSIOPEIA method*), 41  
get\_match\_history() (*cassiopeia.CASSIOPEIA method*), 41  
get\_profile\_icons() (*cassiopeia.CASSIOPEIA method*), 51  
get\_realms() (*cassiopeia.CASSIOPEIA method*), 53  
get\_runes() (*cassiopeia.CASSIOPEIA method*), 53  
get\_side() (*cassiopeia.data.SUMMONERS\_RIFT\_AREA method*), 17  
get\_status() (*cassiopeia.CASSIOPEIA method*), 56  
get\_summoner() (*cassiopeia.CASSIOPEIA method*), 59  
get\_summoner\_spells() (*cassiopeia.CASSIOPEIA method*), 59  
get\_version() (*cassiopeia.CASSIOPEIA method*), 63  
get\_versions() (*cassiopeia.CASSIOPEIA method*), 63  
gold (*cassiopeia.data.TIER attribute*), 17  
gold (*cassiopeia.ITEM attribute*), 31  
gold\_earned (*cassiopeia.core.MATCH.PARTICIPANT\_FRAME attribute*), 48  
gold\_earned (*cassiopeia.core.MATCH.PARTICIPANT\_STATE attribute*), 50  
gold\_earned (*cassiopeia.core.MATCH.PARTICIPANT\_STATS attribute*), 45  
gold\_earned (*cassiopeia\_championgg.core.CHAMPIONGG\_MATCHUP\_STATS attribute*), 26  
gold\_earned (*cassiopeia\_championgg.core.CHAMPIONGG\_STATS attribute*), 25  
gold\_spent (*cassiopeia.core.MATCH.PARTICIPANT\_STATS attribute*), 45  
grandmaster (*cassiopeia.data.TIER attribute*), 17  
grasp (*cassiopeia.ITEM attribute*), 31  
guardian\_invasion\_normal (*cassiopeia.data.QUEUE attribute*), 14  
guardian\_invasion\_onslaught (*cassiopeia.data.QUEUE attribute*), 14

**H**

health (*cassiopeia.core.staticdata.CHAMPION STATS attribute*), 21  
health (*cassiopeia.core.staticdata.ITEM ITEM STATS attribute*), 32  
health\_per\_level (*cassiopeia.core.staticdata.CHAMPION STATS attribute*), 21  
health\_regen (*cassiopeia.core.staticdata.CHAMPION STATS attribute*), 21  
health\_regen (*cassiopeia.core.staticdata.ITEM ITEM STATS attribute*), 32  
health\_regen\_per\_level (*cassiopeia.core.staticdata.CHAMPION STATS attribute*)

*tribute), 21*  
**heat** (*cassiopeia.data.Resource attribute*), 16  
**hexakill\_summoners\_rift** (*cassiopeia.data.Queue attribute*), 14  
**hexakill\_twisted\_treeline** (*cassiopeia.data.Queue attribute*), 14  
**hide** (*cassiopeia.Item attribute*), 31  
**hostname** (*cassiopeia.ShardStatus attribute*), 56  
**hot\_streak** (*cassiopeia.core.league.LeagueEntry attribute*), 37

|

**id** (*cassiopeia.Champion attribute*), 20  
**id** (*cassiopeia.core.ChallengerLeague attribute*), 33  
**id** (*cassiopeia.core.league.League attribute*), 33  
**id** (*cassiopeia.core.MasterLeague attribute*), 34  
**id** (*cassiopeia.core.match.Participant attribute*), 44  
**id** (*cassiopeia.core.match.Timeline attribute*), 48  
**id** (*cassiopeia.core.spectator.CurrentMatch attribute*), 58  
**id** (*cassiopeia.core.staticdata.champion.Skin attribute*), 22  
**id** (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 52  
**id** (*cassiopeia.data.Queue attribute*), 14  
**id** (*cassiopeia.Item attribute*), 31  
**id** (*cassiopeia.Map attribute*), 40  
**id** (*cassiopeia.Match attribute*), 43  
**id** (*cassiopeia.Rune attribute*), 55  
**id** (*cassiopeia.Summoner attribute*), 59  
**id** (*cassiopeia.SummonerSpell attribute*), 61  
**id** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26  
**id** (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
**image** (*cassiopeia.Champion attribute*), 20  
**image** (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 52  
**image** (*cassiopeia.Item attribute*), 31  
**image** (*cassiopeia.Map attribute*), 40  
**image** (*cassiopeia.Rune attribute*), 55  
**image** (*cassiopeia.SummonerSpell attribute*), 61  
**image\_info** (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
**image\_info** (*cassiopeia.core.staticdata.champion.Passive attribute*), 22  
**in\_store** (*cassiopeia.Item attribute*), 31  
**inactive** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**included\_data** (*cassiopeia.Champion attribute*), 20  
**included\_data** (*cassiopeia.Champions attribute*), 18  
**included\_data** (*cassiopeia.Item attribute*), 31  
**included\_data** (*cassiopeia.Items attribute*), 29  
**included\_data** (*cassiopeia.Rune attribute*), 55  
**included\_data** (*cassiopeia.Runes attribute*), 54  
**included\_data** (*cassiopeia.SummonerSpell attribute*), 61  
**included\_data** (*cassiopeia.SummonerSpells attribute*), 60  
**index()** (*cassiopeia.ChampionMasteries method*), 27  
**index()** (*cassiopeia.Champions method*), 18  
**index()** (*cassiopeia.core.league.LeagueEntries method*), 36  
**index()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 34  
**index()** (*cassiopeia.core.match.MatchHistory method*), 42  
**index()** (*cassiopeia.FeaturedMatches method*), 57  
**index()** (*cassiopeia.Items method*), 30  
**index()** (*cassiopeia.Locales method*), 38  
**index()** (*cassiopeia.Maps method*), 40  
**index()** (*cassiopeia.ProfileIcons method*), 51  
**index()** (*cassiopeia.Runes method*), 54  
**index()** (*cassiopeia.SummonerSpells method*), 60  
**index()** (*cassiopeia.Versions method*), 63  
**individual\_position** (*cassiopeia.core.match.Participant attribute*), 44  
**info** (*cassiopeia.Champion attribute*), 20  
**Info** (*class in cassiopeia.core.staticdata.champion*), 21  
**inhibitor\_kills** (*cassiopeia.core.match.ParticipantStats attribute*), 45  
**inhibitor\_kills** (*cassiopeia.core.match.Team attribute*), 44  
**inhibitor\_takedowns** (*cassiopeia.core.match.ParticipantStats attribute*), 45  
**inhibitors\_lost** (*cassiopeia.core.match.ParticipantStats attribute*), 46  
**INNER** (*cassiopeia.data.Tower attribute*), 18  
**insert()** (*cassiopeia.ChampionMasteries method*), 27  
**insert()** (*cassiopeia.Champions method*), 19  
**insert()** (*cassiopeia.core.league.LeagueEntries method*), 36  
**insert()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 35  
**insert()** (*cassiopeia.core.match.MatchHistory method*), 42  
**insert()** (*cassiopeia.FeaturedMatches method*), 57  
**insert()** (*cassiopeia.Items method*), 30  
**insert()** (*cassiopeia.Locales method*), 38  
**insert()** (*cassiopeia.Maps method*), 40  
**insert()** (*cassiopeia.ProfileIcons method*), 51  
**insert()** (*cassiopeia.Runes method*), 54  
**insert()** (*cassiopeia.SummonerSpells method*), 60  
**insert()** (*cassiopeia.Versions method*), 63  
**inspiration** (*cassiopeia.Runes attribute*), 54

iron (*cassiopeia.data.Tier attribute*), 17  
is\_bot (*cassiopeia.core.match.Participant attribute*), 44  
is\_bot (*cassiopeia.core.spectator.Participant attribute*), 58  
is\_keystone (*cassiopeia.Rune attribute*), 55  
is\_remake (*cassiopeia.Match attribute*), 43  
Item (*class in cassiopeia*), 30  
item\_id (*cassiopeia.core.match.Event attribute*), 49  
item\_sets (*cassiopeia.core.staticdata.champion.Recommendation attribute*), 23  
items (*cassiopeia.core.match.ParticipantState attribute*), 50  
items (*cassiopeia.core.match.ParticipantStats attribute*), 46  
items (*cassiopeia.core.staticdata.champion.ItemSet attribute*), 23  
Items (*class in cassiopeia*), 29  
items\_purchased (*cassiopeia.core.match.ParticipantStats attribute*), 46  
ItemSet (*class in cassiopeia.core.staticdata.champion*), 23  
ItemStats (*class in cassiopeia.core.staticdata.item*), 31

**J**

japan (*cassiopeia.data.Platform attribute*), 11  
japan (*cassiopeia.data.Region attribute*), 15  
jungle (*cassiopeia.data.Lane attribute*), 10  
jungle (*cassiopeia.data.Position attribute*), 12  
jungle\_bot\_blue (*cassiopeia.data.SummonersRiftArea attribute*), 17  
jungle\_bot\_red (*cassiopeia.data.SummonersRiftArea attribute*), 17  
jungle\_top\_blue (*cassiopeia.data.SummonersRiftArea attribute*), 17  
jungle\_top\_red (*cassiopeia.data.SummonersRiftArea attribute*), 17

**K**

kda (*cassiopeia.core.match.ParticipantState attribute*), 50  
kda (*cassiopeia.core.match.ParticipantStats attribute*), 46  
key (*cassiopeia.Champion attribute*), 20  
key (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
key (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 24  
key (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 62  
key (*cassiopeia.SummonerSpell attribute*), 61  
Key (*class in cassiopeia.data*), 10  
keyboard\_key (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24

keystones (*cassiopeia.Runes attribute*), 54  
keywords (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
keywords (*cassiopeia.Item attribute*), 31  
killer\_id (*cassiopeia.core.match.Event attribute*), 49  
killing\_sprees (*cassiopeia.core.match.ParticipantStats attribute*), 46  
killings\_sprees (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26  
kills (*cassiopeia.core.match.ParticipantState attribute*), 50  
kills (*cassiopeia.core.match.ParticipantStats attribute*), 46  
kills (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26  
kills (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
kills\_heatmap() (*cassiopeia.Match method*), 43  
korea (*cassiopeia.data.Platform attribute*), 11  
korea (*cassiopeia.data.Region attribute*), 15

**L**

lane (*cassiopeia.core.match.Participant attribute*), 44  
Lane (*class in cassiopeia.data*), 10  
lane\_type (*cassiopeia.core.match.Event attribute*), 49  
language (*cassiopeia.Realms attribute*), 53  
LanguageStrings (*class in cassiopeia*), 33  
largest\_critical\_strike (*cassiopeia.core.match.ParticipantStats attribute*), 46  
largest\_killing\_spree (*cassiopeia.core.match.ParticipantStats attribute*), 46  
largest\_multi\_kill (*cassiopeia.core.match.ParticipantStats attribute*), 46  
last\_played (*cassiopeia.ChampionMastery attribute*), 28  
latest() (*cassiopeia.Patch class method*), 50  
latest\_data\_dragon (*cassiopeia.Realms attribute*), 53  
latest\_versions (*cassiopeia.Realms attribute*), 53  
latin\_america\_north (*cassiopeia.data.Platform attribute*), 11  
latin\_america\_north (*cassiopeia.data.Region attribute*), 15  
latin\_america\_south (*cassiopeia.data.Platform attribute*), 11  
latin\_america\_south (*cassiopeia.data.Region attribute*), 15  
league (*cassiopeia.core.league.LeagueEntry attribute*), 37  
League (*class in cassiopeia.core.league*), 33

**league\_entries** (*cassiopeia.Summoner attribute*), 59  
**league\_points** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**LeagueEntries** (*class in cassiopeia.core.league*), 35  
**LeagueEntry** (*class in cassiopeia.core.league*), 37  
**leagues** (*cassiopeia.cassiopeia.Summoner attribute*), 33  
**LeagueSummonerEntries** (*class in cassiopeia.core.league*), 34  
**legacy\_mode** (*cassiopeia.Realms attribute*), 53  
**level** (*cassiopeia.ChampionMastery attribute*), 28  
**level** (*cassiopeia.core.match.ParticipantFrame attribute*), 48  
**level** (*cassiopeia.core.match.ParticipantState attribute*), 50  
**level** (*cassiopeia.core.match.ParticipantStats attribute*), 46  
**level** (*cassiopeia.Summoner attribute*), 59  
**level\_up\_type** (*cassiopeia.core.match.Event attribute*), 49  
**life\_steal** (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
**link** (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 24  
**link** (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 62  
**load()** (*cassiopeia.Champion method*), 20  
**loading\_image** (*cassiopeia.core.staticdata.champion.Skin attribute*), 22  
**loading\_image\_url** (*cassiopeia.core.staticdata.champion.Skin attribute*), 22  
**locale** (*cassiopeia.Champion attribute*), 20  
**locale** (*cassiopeia.Champions attribute*), 19  
**locale** (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 52  
**locale** (*cassiopeia.Item attribute*), 31  
**locale** (*cassiopeia.Items attribute*), 30  
**locale** (*cassiopeia.LanguageStrings attribute*), 33  
**locale** (*cassiopeia.Map attribute*), 40  
**locale** (*cassiopeia.Maps attribute*), 40  
**locale** (*cassiopeia.ProfileIcons attribute*), 52  
**locale** (*cassiopeia.Realms attribute*), 53  
**locale** (*cassiopeia.Rune attribute*), 55  
**locale** (*cassiopeia.Runes attribute*), 54  
**locale** (*cassiopeia.SummonerSpell attribute*), 61  
**locale** (*cassiopeia.SummonerSpells attribute*), 60  
**locales** (*cassiopeia.ShardStatus attribute*), 56  
**Locales** (*class in cassiopeia*), 38  
**location** (*cassiopeia.core.match.Position attribute*), 49  
**long\_description** (*cassiopeia.Rune attribute*), 55  
**longest\_time\_spent\_living** (*cassiopeia.core.match.ParticipantStats attribute*), 46  
**lore** (*cassiopeia.Champion attribute*), 20  
**losses** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**losses** (*cassiopeia.core.league.MiniSeries attribute*), 35

## M

**magic** (*cassiopeia.core.staticdata.champion.Info attribute*), 21  
**magic\_damage\_dealt** (*cassiopeia.core.match.ParticipantStats attribute*), 46  
**magic\_damage\_dealt\_to\_champions** (*cassiopeia.core.match.ParticipantStats attribute*), 46  
**magic\_damage\_taken** (*cassiopeia.core.match.ParticipantStats attribute*), 46  
**magic\_resist** (*cassiopeia.core.staticdata.champion.Stats attribute*), 22  
**magic\_resist** (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
**magic\_resist\_per\_level** (*cassiopeia.core.staticdata.champion.Stats attribute*), 22  
**major** (*cassiopeia.Patch attribute*), 50  
**majorminor** (*cassiopeia.Patch attribute*), 51  
**mana** (*cassiopeia.core.staticdata.champion.Stats attribute*), 22  
**mana** (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
**mana** (*cassiopeia.data.Resource attribute*), 16  
**mana\_per\_level** (*cassiopeia.core.staticdata.champion.Stats attribute*), 22  
**mana\_regen** (*cassiopeia.core.staticdata.champion.Stats attribute*), 22  
**mana\_regen** (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
**mana\_regen\_per\_level** (*cassiopeia.core.staticdata.champion.Stats attribute*), 22  
**map** (*cassiopeia.core.spectator.CurrentMatch attribute*), 58  
**map** (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 23  
**map** (*cassiopeia.Match attribute*), 43  
**Map** (*class in cassiopeia*), 40  
**maps** (*cassiopeia.Item attribute*), 31  
**Maps** (*class in cassiopeia*), 39  
**master** (*cassiopeia.data.Tier attribute*), 17  
**MasterLeague** (*class in cassiopeia.core*), 34  
**MasteryTree** (*class in cassiopeia.data*), 11  
**Match** (*class in cassiopeia*), 42  
**match\_history** (*cassiopeia.cassiopeia.Summoner attribute*), 41

**N**

name (*cassiopeia.Champion* attribute), 20  
name (*cassiopeia.core.ChallengerLeague* attribute), 33  
name (*cassiopeia.core.league.League* attribute), 33  
name (*cassiopeia.core.MasterLeague* attribute), 34  
name (*cassiopeia.core.staticdata.champion.ChampionSpell* attribute), 24  
name (*cassiopeia.core.staticdata.champion.Passive* attribute), 22  
name (*cassiopeia.core.staticdata.champion.Skin* attribute), 22  
name (*cassiopeia.core.staticdata.profileicon.ProfileIcon* attribute), 52  
name (*cassiopeia.Item* attribute), 31  
name (*cassiopeia.Map* attribute), 41  
name (*cassiopeia.Patch* attribute), 51  
name (*cassiopeia.Rune* attribute), 55  
name (*cassiopeia.ShardStatus* attribute), 56  
name (*cassiopeia.Summoner* attribute), 59  
name (*cassiopeia.SummonerSpell* attribute), 61  
nemesis\_draft (*cassiopeia.data.Queue* attribute), 14  
neutral\_minions\_killed (*cassiopeia.core.match.ParticipantFrame* attribute), 48  
neutral\_minions\_killed (*cassiopeia.core.match.ParticipantState* attribute), 50  
neutral\_minions\_killed (*cassiopeia.core.match.ParticipantStats* attribute), 46  
neutral\_minions\_killed\_in\_enemy\_jungle (*cassiopeia\_championgg.core.ChampionGGStats* attribute), 25  
neutral\_minions\_killed\_in\_team\_jungle (*cassiopeia\_championgg.core.ChampionGGStats* attribute), 25  
neutral\_minions\_killed\_team\_jungle (*cassiopeia\_championgg.core.ChampionGGMatchupStats* attribute), 26  
NEXUS (*cassiopeia.data.Tower* attribute), 18  
nexus\_blitz (*cassiopeia.data.GameMode* attribute), 9  
nexus\_blitz (*cassiopeia.data.Queue* attribute), 14  
nexus\_blue (*cassiopeia.data.SummonersRiftArea* attribute), 17  
nexus\_kills (*cassiopeia.core.match.ParticipantStats* attribute), 46  
nexus\_lost (*cassiopeia.core.match.ParticipantStats* attribute), 46  
nexus\_red (*cassiopeia.data.SummonersRiftArea* attribute), 17  
nexus\_siege (*cassiopeia.data.GameMode* attribute), 9  
nexus\_siege (*cassiopeia.data.Queue* attribute), 14  
nexus\_takedowns (*cassiopeia.core.match.ParticipantStats* attribute),

46  
**nmatches** (*cassiopeia\_championgg.core.ChampionGGMatchup* attribute), 26  
**no\_cost** (*cassiopeia.data.Resource* attribute), 16  
**none** (*cassiopeia.data.Position* attribute), 12  
**none** (*cassiopeia.data.Resource* attribute), 16  
**none** (*cassiopeia.data.Role* attribute), 16  
**none** (*cassiopeia.data.SummonersRiftArea* attribute), 17  
**normal** (*cassiopeia.data.MatchType* attribute), 11  
**normal\_draft\_fives** (*cassiopeia.data.Queue* attribute), 14  
**normal\_tft** (*cassiopeia.data.Queue* attribute), 14  
**north\_america** (*cassiopeia.data.Platform* attribute), 11  
**north\_america** (*cassiopeia.data.Region* attribute), 15  
**not\_played** (*cassiopeia.core.league.MiniSeries* attribute), 35  
**number** (*cassiopeia.core.staticdata.champion.Skin* attribute), 22

**O**

**objectives** (*cassiopeia.core.match.ParticipantState* attribute), 50  
**objectives\_stolen** (*cassiopeia.core.match.ParticipantStats* attribute), 46  
**objectives\_stolen\_assists** (*cassiopeia.core.match.ParticipantStats* attribute), 46  
**observer\_key** (*cassiopeia.core.spectator.CurrentMatch* attribute), 58  
**oceania** (*cassiopeia.data.Platform* attribute), 11  
**oceania** (*cassiopeia.data.Region* attribute), 15  
**odyssey** (*cassiopeia.data.GameMode* attribute), 9  
**odyssey\_cadet** (*cassiopeia.data.Queue* attribute), 14  
**odyssey\_captain** (*cassiopeia.data.Queue* attribute), 14  
**odyssey\_crewmember** (*cassiopeia.data.Queue* attribute), 14  
**odyssey\_intro** (*cassiopeia.data.Queue* attribute), 14  
**odyssey\_onslaught** (*cassiopeia.data.Queue* attribute), 14  
**one** (*cassiopeia.data.Division* attribute), 9  
**one\_for\_all** (*cassiopeia.data.GameMode* attribute), 9  
**one\_for\_all** (*cassiopeia.data.Queue* attribute), 14  
**one\_for\_all\_rapid** (*cassiopeia.data.Queue* attribute), 14  
**OUTER** (*cassiopeia.data.Tower* attribute), 18  
**overcharge** (*cassiopeia.data.GameMode* attribute), 9  
**overcharge** (*cassiopeia.data.Queue* attribute), 14

**P**

**Participant** (*class in cassiopeia.core.match*), 44  
**Participant** (*class in cassiopeia.core.spectator*), 58  
**participant\_frames** (*cassiopeia.core.match.Frame* attribute), 48  
**participant\_id** (*cassiopeia.core.match.Event* attribute), 49  
**participant\_id** (*cassiopeia.core.match.ParticipantFrame* attribute), 48  
**ParticipantFrame** (*class in cassiopeia.core.match*), 48  
**participants** (*cassiopeia.core.match.Team* attribute), 44  
**participants** (*cassiopeia.core.spectator.CurrentMatch* attribute), 58  
**participants** (*cassiopeia.core.spectator.Team* attribute), 58  
**participants** (*cassiopeia.Match* attribute), 43  
**ParticipantState** (*class in cassiopeia.core.match*), 50  
**ParticipantStats** (*class in cassiopeia.core.match*), 45  
**ParticipantTimeline** (*class in cassiopeia.core.match*), 47  
**passive** (*cassiopeia.Champion* attribute), 20  
**Passive** (*class in cassiopeia.core.staticdata.champion*), 22  
**patch** (*cassiopeia.Match* attribute), 43  
**patch** (*cassiopeia\_championgg.core.ChampionGGMatchup* attribute), 26  
**patch** (*cassiopeia\_championgg.core.ChampionGGStats* attribute), 25  
**Patch** (*class in cassiopeia*), 50  
**path** (*cassiopeia.Rune* attribute), 55  
**penta\_kills** (*cassiopeia.core.match.ParticipantStats* attribute), 46  
**percent\_ability\_power** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**percent\_armor** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**percent\_attack\_damage** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**percent\_attack\_speed** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**percent\_attack\_speed\_per\_level** (*cassiopeia.core.staticdata.champion.Stats* attribute), 22  
**percent\_block** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**percent\_critical\_strike\_damage** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**percent\_health** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32  
**percent\_health\_regen** (*cassiopeia.core.staticdata.item.ItemStats* attribute), 32

percent\_magic\_resist (cassiopeia.core.staticdata.item.ItemStats tribute), 32

percent\_mana\_regen (cassiopeia.core.staticdata.item.ItemStats tribute), 32

percent\_movespeed (cassiopeia.core.staticdata.item.ItemStats tribute), 32

percent\_xp\_bonus (cassiopeia.core.staticdata.item.ItemStats tribute), 32

performance\_score (cassiopeia\_championgg.core.ChampionGGStats attribute), 25

physical\_damage\_dealt (cassiopeia.core.match.ParticipantStats attribute), 46

physical\_damage\_dealt\_to\_champions (cassiopeia.core.match.ParticipantStats attribute), 46

physical\_damage\_taken (cassiopeia.core.match.ParticipantStats attribute), 46

pipeline (cassiopeia.\_configuration.settings.Settings attribute), 8

plaintext (cassiopeia.Item attribute), 31

platform (cassiopeia.Champion attribute), 20

platform (cassiopeia.ChampionMasteryes attribute), 28

platform (cassiopeia.ChampionMastery attribute), 28

platform (cassiopeia.Champions attribute), 19

platform (cassiopeia.core.ChallengerLeague attribute), 33

platform (cassiopeia.core.league.League attribute), 33

platform (cassiopeia.core.league.LeagueEntries attribute), 36

platform (cassiopeia.core.league.LeagueEntry attribute), 37

platform (cassiopeia.core.league.LeagueSummonerEntries attribute), 35

platform (cassiopeia.core.MasterLeague attribute), 34

platform (cassiopeia.core.match.Timeline attribute), 48

platform (cassiopeia.core.spectator.CurrentMatch attribute), 58

platform (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), 52

platform (cassiopeia.data.Region attribute), 15

platform (cassiopeia.FeaturedMatches attribute), 57

platform (cassiopeia.Item attribute), 31

platform (cassiopeia.Items attribute), 30

platform (cassiopeia.LanguageStrings attribute), 33

platform (cassiopeia.Locales attribute), 38

platform (cassiopeia.Map attribute), 41

platform (cassiopeia.Maps attribute), 40

(cas- at) platform (cassiopeia.Match attribute), 43

(cas- at) platform (cassiopeia.ProfileIcons attribute), 52

(cas- at) platform (cassiopeia.Realms attribute), 53

(cas- at) platform (cassiopeia.Rune attribute), 55

(cas- at) platform (cassiopeia.Runes attribute), 54

(cas- at) platform (cassiopeia.ShardStatus attribute), 56

(cas- at) platform (cassiopeia.Summoner attribute), 59

(cas- at) platform (cassiopeia.SummonerSpell attribute), 61

(cas- at) platform (cassiopeia.SummonerSpells attribute), 60

(cas- at) platform (cassiopeia.Versions attribute), 63

Platform (class in cassiopeia.data), 11

platinum (cassiopeia.data.Tier attribute), 17

play\_rate (cassiopeia\_championgg.core.ChampionGGStats attribute), 25

play\_rate\_by\_role (cassiopeia\_championgg.core.ChampionGGStats attribute), 25

play\_rates (cassiopeia.Champion attribute), 20

plugins (cassiopeia.\_configuration.settings.Settings attribute), 8

points (cassiopeia.ChampionMastery attribute), 28

points\_since\_last\_level (cassiopeia.ChampionMastery attribute), 29

points\_until\_next\_level (cassiopeia.ChampionMastery attribute), 29

pop() (cassiopeia.ChampionMasteryes method), 28

pop() (cassiopeia.Champions method), 19

pop() (cassiopeia.core.league.LeagueEntries method), 36

pop() (cassiopeia.core.league.LeagueSummonerEntries method), 35

pop() (cassiopeia.core.match.MatchHistory method), 42

pop() (cassiopeia.FeaturedMatches method), 57

pop() (cassiopeia.Items method), 30

pop() (cassiopeia.Locales method), 38

pop() (cassiopeia.Maps method), 40

pop() (cassiopeia.ProfileIcons method), 52

pop() (cassiopeia.Runes method), 54

pop() (cassiopeia.SummonerSpells method), 60

pop() (cassiopeia.Versions method), 63

poro\_king (cassiopeia.data.GameMode attribute), 10

poro\_king (cassiopeia.data.Queue attribute), 14

position (cassiopeia.core.match.Event attribute), 49

position (cassiopeia.core.match.ParticipantFrame attribute), 48

position (cassiopeia.core.match.ParticipantState attribute), 50

Position (class in cassiopeia.core.match), 49

Position (class in cassiopeia.data), 12

practice\_tool (cassiopeia.data.GameMode attribute), 10

precision (cassiopeia.Runes attribute), 54

priority (cassiopeia.core.staticdata.champion.RecommendedItems attribute), 23

**profile\_icon** (*cassiopeia.cassiopeia.Summoner attribute*), 51  
**profile\_icon** (*cassiopeia.Summoner attribute*), 59  
**ProfileIcon** (*class in cassiopeia.core.staticdata.profileicon*), 52  
**ProfileIcons** (*class in cassiopeia*), 51  
**progress** (*cassiopeia.core.league.MiniSeries attribute*), 35  
**project** (*cassiopeia.data.GameMode attribute*), 10  
**project** (*cassiopeia.data.Queue attribute*), 14  
**promos** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**puuid** (*cassiopeia.Summoner attribute*), 59

## Q

**Q** (*cassiopeia.data.Key attribute*), 10  
**quadra\_kills** (*cassiopeia.core.match.ParticipantStats attribute*), 46  
**queue** (*cassiopeia.core.ChallengerLeague attribute*), 33  
**queue** (*cassiopeia.core.league.League attribute*), 33  
**queue** (*cassiopeia.core.league.LeagueEntries attribute*), 36  
**queue** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**queue** (*cassiopeia.core.MasterLeague attribute*), 34  
**queue** (*cassiopeia.core.spectator.CurrentMatch attribute*), 58  
**queue** (*cassiopeia.Match attribute*), 43  
**Queue** (*class in cassiopeia.data*), 12  
**queue()** (*cassiopeia.core.match.MatchHistory method*), 42

## R

**R** (*cassiopeia.data.Key attribute*), 10  
**rage** (*cassiopeia.data.Resource attribute*), 16  
**range** (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
**range** (*cassiopeia.SummonerSpell attribute*), 62  
**Rank** (*class in cassiopeia.data*), 15  
**ranked** (*cassiopeia.data.MatchType attribute*), 11  
**ranked\_flex\_fives** (*cassiopeia.data.Queue attribute*), 14  
**ranked\_flex\_threes** (*cassiopeia.data.Queue attribute*), 14  
**ranked\_solo\_fives** (*cassiopeia.data.Queue attribute*), 14  
**ranked\_tft** (*cassiopeia.data.Queue attribute*), 14  
**ranked\_tft\_double\_up** (*cassiopeia.data.Queue attribute*), 14  
**ranked\_tft\_pairs** (*cassiopeia.data.Queue attribute*), 14  
**ranks** (*cassiopeia.Summoner attribute*), 59  
**ranks\_with** (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 24

**ranks\_with** (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 62  
**Realms** (*class in cassiopeia*), 53  
**rec\_math** (*cassiopeia.core.staticdata.champion.ItemSet attribute*), 23  
**recommended\_itemsets** (*cassiopeia.Champion attribute*), 20  
**RecommendedItems** (*class in cassiopeia.core.staticdata.champion*), 23  
**red** (*cassiopeia.data.Side attribute*), 16  
**red\_team** (*cassiopeia.core.spectator.CurrentMatch attribute*), 58  
**red\_team** (*cassiopeia.Match attribute*), 43  
**region** (*cassiopeia.Champion attribute*), 20  
**region** (*cassiopeia.ChampionMasteries attribute*), 28  
**region** (*cassiopeia.ChampionMastery attribute*), 29  
**region** (*cassiopeia.Champions attribute*), 19  
**region** (*cassiopeia.core.ChallengerLeague attribute*), 34  
**region** (*cassiopeia.core.league.League attribute*), 33  
**region** (*cassiopeia.core.league.LeagueEntries attribute*), 36  
**region** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**region** (*cassiopeia.core.league.LeagueSummonerEntries attribute*), 35  
**region** (*cassiopeia.core.MasterLeague attribute*), 34  
**region** (*cassiopeia.core.match.Timeline attribute*), 48  
**region** (*cassiopeia.core.spectator.CurrentMatch attribute*), 58  
**region** (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 52  
**region** (*cassiopeia.data.Platform attribute*), 11  
**region** (*cassiopeia.FeaturedMatches attribute*), 57  
**region** (*cassiopeia.Item attribute*), 31  
**region** (*cassiopeia.Items attribute*), 30  
**region** (*cassiopeia.LanguageStrings attribute*), 33  
**region** (*cassiopeia.Locales attribute*), 38  
**region** (*cassiopeia.Map attribute*), 41  
**region** (*cassiopeia.Maps attribute*), 40  
**region** (*cassiopeia.Match attribute*), 43  
**region** (*cassiopeia.Patch attribute*), 51  
**region** (*cassiopeia.ProfileIcons attribute*), 52  
**region** (*cassiopeia.Realms attribute*), 53  
**region** (*cassiopeia.Rune attribute*), 55  
**region** (*cassiopeia.Runes attribute*), 54  
**region** (*cassiopeia.ShardStatus attribute*), 56  
**region** (*cassiopeia.Summoner attribute*), 59  
**region** (*cassiopeia.SummonerSpell attribute*), 62  
**region** (*cassiopeia.SummonerSpells attribute*), 60  
**region** (*cassiopeia.Versions attribute*), 63  
**region** (*cassiopeia\_championgg.core.ChampionGGMatchup attribute*), 26  
**Region** (*class in cassiopeia.data*), 15  
**release\_date** (*cassiopeia.Champion attribute*), 20

**remove()** (*cassiopeia.ChampionMasteryes method*), 28  
**remove()** (*cassiopeia.Champions method*), 19  
**remove()** (*cassiopeia.core.league.LeagueEntries method*), 36  
**remove()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 35  
**remove()** (*cassiopeia.core.match.MatchHistory method*), 42  
**remove()** (*cassiopeia.FeaturedMatches method*), 57  
**remove()** (*cassiopeia.Items method*), 30  
**remove()** (*cassiopeia.Locales method*), 38  
**remove()** (*cassiopeia.Maps method*), 40  
**remove()** (*cassiopeia.ProfileIcons method*), 52  
**remove()** (*cassiopeia.Runes method*), 54  
**remove()** (*cassiopeia.SummonerSpells method*), 60  
**remove()** (*cassiopeia.Versions method*), 63  
**resolve** (*cassiopeia.data.MasteryTree attribute*), 11  
**resolve** (*cassiopeia.Runes attribute*), 54  
**resource** (*cassiopeia.Champion attribute*), 20  
**resource** (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
**resource** (*cassiopeia.SummonerSpell attribute*), 62  
**Resource** (*class in cassiopeia.data*), 15  
**reverse()** (*cassiopeia.ChampionMasteryes method*), 28  
**reverse()** (*cassiopeia.Champions method*), 19  
**reverse()** (*cassiopeia.core.league.LeagueEntries method*), 36  
**reverse()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 35  
**reverse()** (*cassiopeia.core.match.MatchHistory method*), 42  
**reverse()** (*cassiopeia.FeaturedMatches method*), 57  
**reverse()** (*cassiopeia.Items method*), 30  
**reverse()** (*cassiopeia.Locales method*), 39  
**reverse()** (*cassiopeia.Maps method*), 40  
**reverse()** (*cassiopeia.ProfileIcons method*), 52  
**reverse()** (*cassiopeia.Runes method*), 54  
**reverse()** (*cassiopeia.SummonerSpells method*), 60  
**reverse()** (*cassiopeia.Versions method*), 63  
**revision** (*cassiopeia.Patch attribute*), 51  
**revision\_date** (*cassiopeia.Summoner attribute*), 59  
**rift\_herald\_kills** (*cassiopeia.core.match.Team attribute*), 44  
**river\_bot** (*cassiopeia.data.SummonersRiftArea attribute*), 17  
**river\_top** (*cassiopeia.data.SummonersRiftArea attribute*), 17  
**role** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**role** (*cassiopeia.core.match.Participant attribute*), 44  
**role** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 26  
**role** (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
**Role** (*class in cassiopeia.data*), 16

**Rune** (*class in cassiopeia*), 55  
**runes** (*cassiopeia.core.match.Participant attribute*), 44  
**runes** (*cassiopeia.core.spectator.Participant attribute*), 58  
**Runes** (*class in cassiopeia*), 53  
**russia** (*cassiopeia.data.Platform attribute*), 11  
**russia** (*cassiopeia.data.Region attribute*), 15

## S

**sanitized\_description** (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
**sanitized\_description** (*cassiopeia.core.staticdata.champion.Passive attribute*), 22  
**sanitized\_description** (*cassiopeia.Item attribute*), 31  
**sanitized\_description** (*cassiopeia.SummonerSpell attribute*), 62  
**sanitized\_name** (*cassiopeia.Summoner attribute*), 59  
**sanitized\_tooltip** (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
**sanitized\_tooltip** (*cassiopeia.SummonerSpell attribute*), 62  
**sea** (*cassiopeia.data.Continent attribute*), 9  
**search()** (*cassiopeia.ChampionMasteryes method*), 28  
**search()** (*cassiopeia.Champions method*), 19  
**search()** (*cassiopeia.core.league.LeagueEntries method*), 37  
**search()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 35  
**search()** (*cassiopeia.core.match.MatchHistory method*), 42  
**search()** (*cassiopeia.FeaturedMatches method*), 57  
**search()** (*cassiopeia.Items method*), 30  
**search()** (*cassiopeia.Locales method*), 39  
**search()** (*cassiopeia.Maps method*), 40  
**search()** (*cassiopeia.ProfileIcons method*), 52  
**search()** (*cassiopeia.Runes method*), 54  
**search()** (*cassiopeia.SummonerSpells method*), 60  
**search()** (*cassiopeia.Versions method*), 64  
**services** (*cassiopeia.ShardStatus attribute*), 56  
**set\_riot\_api\_key()** (*cassiopeia.\_configuration.settings.Settings method*), 8  
**Settings** (*class in cassiopeia.\_configuration.settings*), 8  
**ShardStatus** (*class in cassiopeia*), 56  
**shield** (*cassiopeia.data.Resource attribute*), 16  
**short\_description** (*cassiopeia.Rune attribute*), 55  
**showdown** (*cassiopeia.data.GameMode attribute*), 10  
**showdown\_1v1** (*cassiopeia.data.Queue attribute*), 14  
**showdown\_2v2** (*cassiopeia.data.Queue attribute*), 14  
**side** (*cassiopeia.core.match.Event attribute*), 49

side (*cassiopeia.core.match.Participant attribute*), 44  
 side (*cassiopeia.core.match.Team attribute*), 44  
 side (*cassiopeia.core.spectator.Participant attribute*), 58  
 side (*cassiopeia.core.spectator.Team attribute*), 58  
 Side (*class in cassiopeia.data*), 16  
 sight\_wards\_bought (*cassiopeia.core.match.ParticipantStats attribute*), 46  
 silver (*cassiopeia.data.Tier attribute*), 17  
 skill (*cassiopeia.core.match.Event attribute*), 49  
 skill\_order (*cassiopeia.core.match.Participant attribute*), 44  
 skills (*cassiopeia.core.match.ParticipantState attribute*), 50  
 Skin (*class in cassiopeia.core.staticdata.champion*), 22  
 skins (*cassiopeia.Champion attribute*), 20  
 slug (*cassiopeia.ShardStatus attribute*), 56  
 solo (*cassiopeia.data.Role attribute*), 16  
 sorcery (*cassiopeia.Runes attribute*), 55  
 sort() (*cassiopeia.ChampionMasteryes method*), 28  
 sort() (*cassiopeia.Champions method*), 19  
 sort() (*cassiopeia.core.league.LeagueEntries method*), 37  
 sort() (*cassiopeia.core.league.LeagueSummonerEntries method*), 35  
 sort() (*cassiopeia.core.match.MatchHistory method*), 42  
 sort() (*cassiopeia.FeaturedMatches method*), 57  
 sort() (*cassiopeia.Items method*), 30  
 sort() (*cassiopeia.Locales method*), 39  
 sort() (*cassiopeia.Maps method*), 40  
 sort() (*cassiopeia.ProfileIcons method*), 52  
 sort() (*cassiopeia.Runes method*), 55  
 sort() (*cassiopeia.SummonerSpells method*), 60  
 sort() (*cassiopeia.Versions method*), 64  
 special\_recipe (*cassiopeia.Item attribute*), 31  
 spell\_1\_casts (*cassiopeia.core.match.ParticipantStats attribute*), 46  
 spell\_2\_casts (*cassiopeia.core.match.ParticipantStats attribute*), 46  
 spell\_3\_casts (*cassiopeia.core.match.ParticipantStats attribute*), 46  
 spell\_4\_casts (*cassiopeia.core.match.ParticipantStats attribute*), 46  
 spell\_vamp (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32  
 spells (*cassiopeia.Champion attribute*), 20  
 SpellVars (*class in cassiopeia.core.staticdata.champion*), 24  
 SpellVars (*class in cassiopeia.core.staticdata.summonerspell*), 62  
 splash (*cassiopeia.core.staticdata.champion.Skin attribute*), 22  
 splash\_url (*cassiopeia.core.staticdata.champion.Skin attribute*), 22  
 sprite (*cassiopeia.Champion attribute*), 21  
 sprite (*cassiopeia.Item attribute*), 31  
 sprite (*cassiopeia.Map attribute*), 41  
 sprite (*cassiopeia.SummonerSpell attribute*), 62  
 star\_guardian (*cassiopeia.data.GameMode attribute*), 10  
 start (*cassiopeia.core.match.MatchHistory attribute*), 42  
 start (*cassiopeia.Match attribute*), 43  
 start (*cassiopeia.Patch attribute*), 51  
 start\_time (*cassiopeia.core.match.MatchHistory attribute*), 42  
 stat\_runes (*cassiopeia.core.match.Participant attribute*), 44  
 stats (*cassiopeia.Champion attribute*), 21  
 stats (*cassiopeia.core.match.Participant attribute*), 44  
 stats (*cassiopeia.Item attribute*), 31  
 Stats (*class in cassiopeia.core.staticdata.champion*), 21  
 store (*cassiopeia.Realms attribute*), 53  
 strings (*cassiopeia.LanguageStrings attribute*), 33  
 summoner (*cassiopeia.ChampionMasteries attribute*), 28  
 summoner (*cassiopeia.ChampionMastery attribute*), 29  
 summoner (*cassiopeia.core.league.LeagueEntry attribute*), 37  
 summoner (*cassiopeia.core.match.Participant attribute*), 44  
 summoner (*cassiopeia.core.spectator.Participant attribute*), 58  
 Summoner (*class in cassiopeia*), 59  
 summoner\_spell\_1\_casts (*cassiopeia.core.match.ParticipantStats attribute*), 46  
 summoner\_spell\_2\_casts (*cassiopeia.core.match.ParticipantStats attribute*), 47  
 summoner\_spell\_d (*cassiopeia.core.match.Participant attribute*), 45  
 summoner\_spell\_d (*cassiopeia.core.spectator.Participant attribute*), 58  
 summoner\_spell\_f (*cassiopeia.core.match.Participant attribute*), 45  
 summoner\_spell\_f (*cassiopeia.core.spectator.Participant attribute*), 58  
 SummonerSpell (*class in cassiopeia*), 61  
 SummonerSpells (*class in cassiopeia*), 59  
 SummonersRiftArea (*class in cassiopeia.data*), 16

## T

tags (*cassiopeia.Champion attribute*), 21  
 tags (*cassiopeia.Item attribute*), 31  
 team (*cassiopeia.core.match.Participant attribute*), 45

**team** (*cassiopeia.core.spectator.Participant attribute*), 58  
**Team** (*class in cassiopeia.core.match*), 43  
**Team** (*class in cassiopeia.core.spectator*), 58  
**team\_position** (*cassiopeia.core.match.Participant attribute*), 45  
**team\_score** (*cassiopeia.core.match.ParticipantFrame attribute*), 48  
**team\_score** (*cassiopeia.core.match.ParticipantState attribute*), 50  
**teams** (*cassiopeia.core.spectator.CurrentMatch attribute*), 58  
**teams** (*cassiopeia.Match attribute*), 43  
**thirty\_to\_end** (*cassiopeia\_championgg.core.ChampionGGMatchupStats method*), 30  
**three** (*cassiopeia.data.Division attribute*), 9  
**threes** (*cassiopeia.core.league.LeagueSummonerEntries attribute*), 35  
**tier** (*cassiopeia.core.ChallengerLeague attribute*), 34  
**tier** (*cassiopeia.core.league.League attribute*), 33  
**tier** (*cassiopeia.core.league.LeagueEntries attribute*), 37  
**tier** (*cassiopeia.core.league.LeagueEntry attribute*), 37  
**tier** (*cassiopeia.core.MasterLeague attribute*), 34  
**tier** (*cassiopeia.Item attribute*), 31  
**tier** (*cassiopeia.Rune attribute*), 55  
**Tier** (*class in cassiopeia.data*), 17  
**time\_CCing\_others** (*cassiopeia.core.match.ParticipantStats attribute*), 47  
**time\_played** (*cassiopeia.core.match.ParticipantStats attribute*), 47  
**timeline** (*cassiopeia.core.match.Participant attribute*), 45  
**timeline** (*cassiopeia.Match attribute*), 43  
**Timeline** (*class in cassiopeia.core.match*), 48  
**timestamp** (*cassiopeia.core.match.Event attribute*), 49  
**timestamp** (*cassiopeia.core.match.Frame attribute*), 48  
**timezone** (*cassiopeia.data.Region attribute*), 15  
**title** (*cassiopeia.Champion attribute*), 21  
**title** (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 23  
**to\_dict()** (*cassiopeia.ChampionMasteries method*), 28  
**to\_dict()** (*cassiopeia.Champions method*), 19  
**to\_dict()** (*cassiopeia.core.league.LeagueEntries method*), 37  
**to\_dict()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 35  
**to\_dict()** (*cassiopeia.core.match.MatchHistory method*), 42  
**to\_dict()** (*cassiopeia.FeaturedMatches method*), 57  
**to\_dict()** (*cassiopeia.Items method*), 30  
**to\_dict()** (*cassiopeia.Locales method*), 39  
**to\_dict()** (*cassiopeia.Maps method*), 40  
**to\_dict()** (*cassiopeia.ProfileIcons method*), 52  
**to\_dict()** (*cassiopeia.Runes method*), 55  
**to\_dict()** (*cassiopeia.SummonerSpells method*), 61  
**to\_dict()** (*cassiopeia.Versions method*), 64  
**to\_json()** (*cassiopeia.ChampionMasteries method*), 28  
**to\_json()** (*cassiopeia.Champions method*), 19  
**to\_json()** (*cassiopeia.core.league.LeagueEntries method*), 37  
**to\_json()** (*cassiopeia.core.league.LeagueSummonerEntries method*), 35  
**to\_json()** (*cassiopeia.core.match.MatchHistory method*), 42  
**to\_json()** (*cassiopeia.FeaturedMatches method*), 57  
**to\_json()** (*cassiopeia.Items method*), 30  
**to\_json()** (*cassiopeia.Locales method*), 39  
**to\_json()** (*cassiopeia.Maps method*), 40  
**to\_json()** (*cassiopeia.ProfileIcons method*), 52  
**to\_json()** (*cassiopeia.Runes method*), 55  
**to\_json()** (*cassiopeia.SummonerSpells method*), 61  
**to\_json()** (*cassiopeia.Versions method*), 64  
**tokens** (*cassiopeia.ChampionMastery attribute*), 29  
**tooltip** (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 24  
**tooltip** (*cassiopeia.SummonerSpell attribute*), 62  
**top** (*cassiopeia.data.Position attribute*), 12  
**top\_lane** (*cassiopeia.data.Lane attribute*), 10  
**top\_lane\_blue** (*cassiopeia.data.SummonersRiftArea attribute*), 17  
**top\_lane\_purple** (*cassiopeia.data.SummonersRiftArea attribute*), 17  
**top\_lane\_red** (*cassiopeia.data.SummonersRiftArea attribute*), 17  
**total\_damage\_dealt** (*cassiopeia.core.match.ParticipantStats attribute*), 47  
**total\_damage\_dealt\_to\_champions** (*cassiopeia.core.match.ParticipantStats attribute*), 47  
**total\_damage\_dealt\_to\_champions** (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 27  
**total\_damage\_shielded\_on\_teammates** (*cassiopeia.core.match.ParticipantStats attribute*), 47  
**total\_damage\_taken** (*cassiopeia.core.match.ParticipantStats attribute*), 47  
**total\_damage\_taken** (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
**total\_heal** (*cassiopeia.core.match.ParticipantStats attribute*), 47  
**total\_healed** (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
**total\_heals\_on\_teammates** (*cassiopeia.core.match.ParticipantStats attribute*), 47

siopeia.core.match.ParticipantStats attribute), 47  
**total\_minions\_killed** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**total\_time\_cc\_dealt** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**total\_time\_spent\_dead** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**total\_units\_healed** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**tourney** (cassiopeia.data.MatchType attribute), 11  
**Tower** (class in cassiopeia.data), 17  
**tower\_kills** (cassiopeia.core.match.Team attribute), 44  
**tower\_type** (cassiopeia.core.match.Event attribute), 49  
**triple\_kills** (cassiopeia.core.match.ParticipantStats  
 attribute), 47  
**true\_damage\_dealt** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**true\_damage\_dealt\_to\_champions** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**true\_damage\_taken** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**turkey** (cassiopeia.data.Platform attribute), 11  
**turkey** (cassiopeia.data.Region attribute), 15  
**turret\_kills** (cassiopeia.core.match.ParticipantStats  
 attribute), 47  
**turret\_takedowns** (casi-  
 siopeia.core.match.ParticipantStats attribute), 47  
**turrets\_lost** (cassiopeia.core.match.ParticipantStats  
 attribute), 47  
**tutorial** (cassiopeia.data.GameMode attribute), 10  
**tutorial** (cassiopeia.data.GameType attribute), 10  
**tutorial** (cassiopeia.data.MatchType attribute), 11  
**tutorial1** (cassiopeia.data.Queue attribute), 14  
**tutorial2** (cassiopeia.data.Queue attribute), 15  
**tutorial3** (cassiopeia.data.Queue attribute), 15  
**tutorial\_1** (cassiopeia.data.GameMode attribute), 10  
**tutorial\_2** (cassiopeia.data.GameMode attribute), 10  
**tutorial\_3** (cassiopeia.data.GameMode attribute), 10  
**twenty\_to\_thirty** (casi-  
 siopeia\_championgg.core.ChampionGGMatchupStats  
 attribute), 27  
**two** (cassiopeia.data.Division attribute), 9  
**type** (cassiopeia.core.match.Event attribute), 49  
**type** (cassiopeia.core.spectator.CurrentMatch attribute), 58  
**type** (cassiopeia.core.staticdata.champion.ItemSet  
 attribute), 23  
**type** (cassiopeia.core.staticdata.champion.RecommendedItems  
 attribute), 23  
**type** (cassiopeia.LanguageStrings attribute), 33  
**type** (cassiopeia.Match attribute), 43

**U**

**ultimate\_spellbook** (cassiopeia.data.Queue attribute), 15  
**UNDEFINED** (cassiopeia.data.Tower attribute), 18  
**unpurchasable\_items** (cassiopeia.Map attribute), 41  
**unranked** (cassiopeia.data.Tier attribute), 17  
**unreal\_kills** (cassiopeia.core.match.ParticipantStats  
 attribute), 47  
**urf** (cassiopeia.data.GameMode attribute), 10  
**urf** (cassiopeia.data.Queue attribute), 15  
**urf\_coop\_ai** (cassiopeia.data.Queue attribute), 15  
**url** (cassiopeia.core.staticdata.profileicon.ProfileIcon  
 attribute), 52  
**utility** (cassiopeia.data.Lane attribute), 11  
**utility** (cassiopeia.data.Position attribute), 12  
**utlbook** (cassiopeia.data.GameMode attribute), 10

**V**

**variables** (cassiopeia.core.staticdata.champion.ChampionSpell  
 attribute), 24  
**variables** (cassiopeia.SummonerSpell attribute), 62  
**verification\_string** (cassiopeia.Summoner attribute), 59  
**version** (cassiopeia.Champion attribute), 21  
**version** (cassiopeia.Champions attribute), 19  
**version** (cassiopeia.core.match.Participant attribute), 45  
**version** (cassiopeia.core.staticdata.profileicon.ProfileIcon  
 attribute), 53  
**version** (cassiopeia.Item attribute), 31  
**version** (cassiopeia.Items attribute), 30  
**version** (cassiopeia.LanguageStrings attribute), 33  
**version** (cassiopeia.Map attribute), 41  
**version** (cassiopeia.Maps attribute), 40  
**version** (cassiopeia.Match attribute), 43  
**version** (cassiopeia.ProfileIcons attribute), 52  
**version** (cassiopeia.Realms attribute), 53  
**version** (cassiopeia.Rune attribute), 55  
**version** (cassiopeia.Runes attribute), 55  
**version** (cassiopeia.SummonerSpell attribute), 62  
**version** (cassiopeia.SummonerSpells attribute), 61  
**version\_from\_match** (casi-  
 siopeia.\_configuration.settings.Settings attribute), 8

**Versions** (class in cassiopeia), 63

**veteran** (cassiopeia.core.league.LeagueEntry attribute), 37

`victim_id` (*cassiopeia.core.match.Event attribute*), 49  
`vision_score` (*cassiopeia.core.match.ParticipantStats attribute*), 47  
`vision_wards_bought` (*cassiopeia.core.match.ParticipantStats attribute*), 47  
`vision_wards_placed` (*cassiopeia.core.match.ParticipantStats attribute*), 47

## W

`W` (*cassiopeia.data.Key attribute*), 10  
`ward_type` (*cassiopeia.core.match.Event attribute*), 49  
`wards_killed` (*cassiopeia.core.match.ParticipantStats attribute*), 47  
`wards_killed` (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
`wards_placed` (*cassiopeia.core.match.ParticipantStats attribute*), 47  
`weighted_score` (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 27  
`win` (*cassiopeia.core.match.ParticipantStats attribute*), 47  
`win` (*cassiopeia.core.match.Team attribute*), 44  
`win_rate` (*cassiopeia\_championgg.core.ChampionGGStats attribute*), 25  
`win_rates` (*cassiopeia.Champion attribute*), 21  
`winrate` (*cassiopeia\_championgg.core.ChampionGGMatchup attribute*), 26  
`winrate` (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 27  
`wins` (*cassiopeia.core.league.LeagueEntry attribute*), 38  
`wins` (*cassiopeia.core.league.MiniSeries attribute*), 35  
`wins` (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 27  
`wins_required` (*cassiopeia.core.league.MiniSeries attribute*), 35

## X

`x` (*cassiopeia.core.match.Position attribute*), 49  
`xp_bonus` (*cassiopeia.core.staticdata.item.ItemStats attribute*), 32

## Y

`y` (*cassiopeia.core.match.Position attribute*), 49

## Z

`zero_to_ten` (*cassiopeia\_championgg.core.ChampionGGMatchupStats attribute*), 27