Cassiopeia Documentation

Release 3.0.x

Rob Rua

Apr 15, 2022

CONTENTS

1 What is Cassiopeia? 1

2 Why use Cass? 3

3 An Example 5

4 Django web Framework 7
- 4.1 Contributing . . . 7
- 4.2 Overview . . . 7
- 4.3 Top Level APIs . . . 72

5 Index and Search 73

Python Module Index 75

Index 77

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 py 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).

CHAPTER

FOUR

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-ocurrence 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_FRECUENCY* 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-ocurrence 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_FRECUENCY* 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[cassiopeia._configuration.settings.T]] = None*)

expire_sinks(*type: Optional[Type[cassiopeia._configuration.settings.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.Enum

An enumeration.

americas = 'AMERICAS'

asia = 'ASIA'

europe = 'EUROPE'

class cassiopeia.data.**Division**(*value*)
Bases: enum.Enum

An enumeration.

four = 'IV'

one = 'I'

three = 'III'

two = 'II'

class cassiopeia.data.**GameMode**(*value*)
Bases: enum.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.Enum

An enumeration.

custom = 'CUSTOM_GAME'

matched = 'MATCHED_GAME'

tutorial = 'TUTORIAL_GAME'

class cassiopeia.data.Key(*value*)
Bases: enum.Enum

An enumeration.

E = 'E'

Q = 'Q'

R = 'R'

W = 'W'

class cassiopeia.data.Lane(*value*)
Bases: enum.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.Enum

An enumeration.

cunning = 'Cunning'

ferocity = 'Ferocity'

resolve = 'Resolve'

class cassiopeia.data.MatchType(*value*)
Bases: enum.Enum

An enumeration.

normal = 'normal'

ranked = 'ranked'

tourney = 'tourney'

tutorial = 'tutorial'

class cassiopeia.data.Platform(*value*)
Bases: enum.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.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.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'

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:* cassiopeia.data.Tier, *division:* cassiopeia.data.Division)
Bases: object

class cassiopeia.data.**Region**(*value*)
Bases: enum.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.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.Enum

An enumeration.

duo = 'DUO'

duo_carry = 'DUO_CARRY'

duo_support = 'DUO_SUPPORT'

from_match_naming_scheme()

none = 'NONE'

solo = 'SOLO'

class cassiopeia.data.**Season**(*value*)

Bases: enum.Enum

An enumeration.

end(*region:* cassiopeia.data.Region) → arrow.arrow.Arrow

from_id()

id

preseason_3 = 'PRESEASON3'

preseason_4 = 'PRESEASON2014'

preseason_5 = 'PRESEASON2015'

preseason_6 = 'PRESEASON2016'

preseason_7 = 'PRESEASON2017'

preseason_8 = 'PRESEASON2018'

preseason_9 = 'PRESEASON2019'

season_3 = 'SEASON3'

season_4 = 'SEASON2014'

season_5 = 'SEASON2015'

season_6 = 'SEASON2016'

season_7 = 'SEASON2017'

season_8 = 'SEASON2018'

season_9 = 'SEASON2019'

start(*region:* cassiopeia.data.Region) → arrow.arrow.Arrow

class cassiopeia.data.Side(*value*)
Bases: enum.Enum

An enumeration.

blue = 100

red = 200

class cassiopeia.data.SummonersRiftArea(*value*)
Bases: enum.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:* cassiopeia.data.Position) → *cassiopeia.data.SummonersRiftArea*

get_lane() → *cassiopeia.data.Lane*

get_side() → *cassiopeia.data.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.Enum

An enumeration.

bronze = 'BRONZE'

challenger = 'CHALLENGER'

diamond = 'DIAMOND'

gold = 'GOLD'

grandmaster = 'GRANDMASTER'

iron = 'IRON'

master = 'MASTER'

platinum = 'PLATINUM'

silver = 'SILVER'

unranked = 'UNRANKED'

class cassiopeia.data.**Tower**(*value*)

Bases: enum.Enum

An enumeration.

BASE = 'BASE_TURRET'

INNER = 'INNER_TURRET'

NEXUS = 'NEXUS_TURRET'

OUTER = 'OUTER_TURRET'

UNDEFINED = 'UNDEFINED_TURRET'

Champions

cassiopeia.**get_champions**() → *cassiopeia.core.staticdata.champion.Champions*

cassiopeia.**get_champion**(*region: Optional[Union[*cassiopeia.data.Region*, str]] = None*) → *cassiopeia.core.staticdata.champion.Champion*

class cassiopeia.**Champions**(**args*, ***kwargs*)

Bases: cassiopeia.core.common.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 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.

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[merakicommons.container.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: cassiopeia.core.common.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*) → *cassiopeia.core.staticdata.champion.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: cassiopeia.core.common.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.champion.**Stats**(***kwargs*)
Bases: cassiopeia.core.common.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: cassiopeia.core.common.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: cassiopeia.core.common.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: cassiopeia.core.common.CassiopeiaObject

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

classmethod from_data(*data: cassiopeia.core.common.CoreData*, *region:* cassiopeia.data.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: cassiopeia.core.common.CassiopeiaObject

Searchable by ['str', 'Item']

classmethod from_data(*data: cassiopeia.core.common.CoreData*, *region:* cassiopeia.data.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: cassiopeia.core.common.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: cassiopeia.core.common.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: cassiopeia.core.common.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: cassiopeia.core.common.CassiopeiaLazyList

class cassiopeia_championgg.core.**ChampionGGMatchup**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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 Masteries

cassiopeia.**get_champion_mastery**(*champion: Union[*cassiopeia.core.staticdata.champion.Champion*, int, str]*, *region: Optional[Union[*cassiopeia.data.Region*, str]] = None*) → *cassiopeia.core.championmastery.ChampionMastery*

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

class cassiopeia.**ChampionMasteries**(**args*, ***kwargs*)

Bases: cassiopeia.core.common.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[merakicommons.container.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: cassiopeia.core.common.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**() → *cassiopeia.core.staticdata.item.Items*

class cassiopeia.**Items**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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[merakicommons.container.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: cassiopeia.core.common.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 additonal 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: cassiopeia.core.common.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**() → *cassiopeia.core.staticdata.languagestrings.LanguageStrings*

class cassiopeia.**LanguageStrings**(**args*, ***kwargs*)

Bases: cassiopeia.core.common.CassiopeiaGhost

Searchable by []

locale

platform

region

strings

type

version

Leagues

cassiopeia.Summoner.**leagues**

cassiopeia.**get_challenger_league**(*region: Optional[Union[*cassiopeia.data.Region*, str]] = None*) → *cassiopeia.core.league.ChallengerLeague*

cassiopeia.**get_master_league**(*region: Optional[Union[*cassiopeia.data.Region*, str]] = None*) → *cassiopeia.core.league.MasterLeague*

class cassiopeia.core.league.**League**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.CassiopeiaGhost

Searchable by ['str', 'Queue', 'Tier']

entries

id

name

platform

queue

region

tier

class cassiopeia.core.**ChallengerLeague**(**args*, ***kwargs*)
Bases: *cassiopeia.core.league.League*

entries

id

name

platform

queue

region

tier

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

entries

id

name

platform

queue

region

tier

class cassiopeia.core.league.**LeagueSummonerEntries**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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[merakicommons.container.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: cassiopeia.core.common.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: cassiopeia.core.common.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[*cassiopeia.data.Region*, str]] = None*, *queue: Optional[*cassiopeia.data.Queue*] = None*, *tier: Optional[*cassiopeia.data.Tier*] = None*, *division: Optional[*cassiopeia.data.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[merakicommons.container.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: cassiopeia.core.common.CassiopeiaGhost

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

division

fresh_blood

classmethod from_data(*data: cassiopeia.core.league.LeagueEntryData*, *loaded_groups: Optional[Set[Type[cassiopeia.core.common.CoreData]]] = None*, *league: Optional[*cassiopeia.core.league.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: cassiopeia.core.common.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[merakicommons.container.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**() → *cassiopeia.core.staticdata.map.Maps*

class cassiopeia.**Maps**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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[merakicommons.container.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: cassiopeia.core.common.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**(*region: Optional[cassiopeia.data.Region] = None*, *platform: Optional[cassiopeia.data.Platform] = None*, *puuid: Optional[str] = None*, *begin_index: Optional[int] = None*, *end_index: Optional[int] = None*, *begin_time: Optional[arrow.arrow.Arrow] = None*, *end_time: Optional[arrow.arrow.Arrow] = None*, *queue: Optional[cassiopeia.data.Queue] = None*, *type: Optional[cassiopeia.data.MatchType] = None*)

cassiopeia.**get_match**(*region: Optional[Union[cassiopeia.data.Region, str]] = None*) → *cassiopeia.core.match.Match*

class cassiopeia.core.match.**MatchHistory**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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.

begin_index

begin_time

clear()
Remove all items from list.

contains(*item: Any*) → bool

continent() → *cassiopeia.data.Continent*

copy()
Return a shallow copy of the list.

count(*object*)
Return number of occurrences of value.

delete(*item: Any*) → None

end_index

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() → *cassiopeia.data.MatchType*

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() → *cassiopeia.data.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[merakicommons.container.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.**Match**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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: cassiopeia.core.match.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: cassiopeia.core.common.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: cassiopeia.core.common.CoreData*, *match:* cassiopeia.core.match.Match)

inhibitor_kills

participants

rift_herald_kills

side

tower_kills

win

class cassiopeia.core.match.Participant(**kwargs)

Bases: cassiopeia.core.common.CassiopeiaObject

Searchable by ['str', 'Summoner', 'Champion', 'Side', 'Rune', 'SummonerSpell']

champion

cumulative_timeline

ended_in_early_surrender

enemy_team

classmethod from_data(*data: cassiopeia.core.common.CoreData*, *match:* cassiopeia.core.match.Match)

id

individual_position

is_bot

lane

match_history_uri

rank_last_season

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: cassiopeia.core.common.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: cassiopeia.core.match.ParticipantStatsData*, *match:* cassiopeia.core.match.Match, *participant:* cassiopeia.core.match.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:* cassiopeia.core.match.Match)

class cassiopeia.core.match.**Timeline**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.CassiopeiaGhost

continent

first_tower_fallen

frame_interval

frames

id

platform

region

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

events

participant_frames

timestamp

class cassiopeia.core.match.**ParticipantFrame**(***kwargs*)
Bases: cassiopeia.core.common.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: cassiopeia.core.common.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, ASCENDED_EVENT, CAPTURE_POINT, PORO_KING_SUMMON

Type Legal values

victim_id

ward_type

class cassiopeia.core.match.**Position**(***kwargs*)

Bases: cassiopeia.core.common.CassiopeiaObject

location

x

y

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

Bases: object

class cassiopeia.core.match.**ParticipantState**(*id: int*, *time: datetime.timedelta*, *participant_timeline:* cassiopeia.core.match.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,* cassiopeia.data.Region*]*, *season:* cassiopeia.data.Season, *name: str*, *start: Union[arrow.arrow.Arrow, float]*, *end: Optional[Union[arrow.arrow.Arrow, float]]*)

Bases: object

end

classmethod from_date(*date: arrow.arrow.Arrow*, *region: Union[*cassiopeia.data.Region*, str]*) → *cassiopeia.core.patch.Patch*

classmethod from_str(*string: str*, *region: Union[*cassiopeia.data.Region*, str]*) → *cassiopeia.core.patch.Patch*

classmethod latest(*region: Optional[Union[*cassiopeia.data.Region*, str]] = None*) → *cassiopeia.core.patch.Patch*

major

majorminor

minor

name

region

revision

season

start

Profile Icons

cassiopeia.Summoner.**profile_icon**

cassiopeia.**get_profile_icons**() → *cassiopeia.core.staticdata.profileicon.ProfileIcons*

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

Bases: cassiopeia.core.common.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[merakicommons.container.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: cassiopeia.core.common.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**() → *cassiopeia.core.staticdata.realm.Realms*

class cassiopeia.**Realms**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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**() → *cassiopeia.core.staticdata.rune.Runes*

class cassiopeia.**Runes**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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[merakicommons.container.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: cassiopeia.core.common.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**() → *cassiopeia.core.status.ShardStatus*

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

Searchable by []

hostname

locales

name

platform

region

services

slug

Spectator

cassiopeia.Summoner.**current_match**

cassiopeia.**get_current_match**(*region: Optional[Union[*cassiopeia.data.Region*, str]] = None*) → *cassiopeia.core.spectator.CurrentMatch*

cassiopeia.**get_featured_matches**() → *cassiopeia.core.spectator.FeaturedMatches*

class cassiopeia.**FeaturedMatches**(**args*, ***kwargs*)

Bases: cassiopeia.core.common.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[merakicommons.container.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: cassiopeia.core.common.CassiopeiaGhost

Searchable by []

blue_team

creation

duration

exists

classmethod from_data(*data: cassiopeia.core.spectator.CurrentGameInfoData*, *summoner: Union[*cassiopeia.core.summoner.Summoner*, str]*)

id

map

mode

observer_key

participants

platform

queue

red_team

region

teams

type

class cassiopeia.core.spectator.**Team**(**kwargs)
Bases: cassiopeia.core.common.CassiopeiaObject

Searchable by []

bans

classmethod from_data(*data: cassiopeia.core.common.CoreData*, *match:* cassiopeia.core.spectator.CurrentMatch)

participants

side

class cassiopeia.core.spectator.**Participant**(***kwargs*)
Bases: cassiopeia.core.common.CassiopeiaObject

Searchable by ['str', 'Summoner', 'Champion']

champion

classmethod from_data(*data: cassiopeia.core.common.CoreData*, *match:* cassiopeia.core.spectator.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[*cassiopeia.data.Region*, str]] = None*) → *cassiopeia.core.summoner.Summoner*

class cassiopeia.**Summoner**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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

rank_last_season

ranks

region
The region for this summoner.

revision_date

sanitized_name

verification_string

Summoner Spells

cassiopeia.**get_summoner_spells**() → *cassiopeia.core.staticdata.summonerspell.SummonerSpells*

class cassiopeia.**SummonerSpells**(**args*, ***kwargs*)
Bases: cassiopeia.core.common.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 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.

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[merakicommons.container.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: cassiopeia.core.common.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: cassiopeia.core.common.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[*cassiopeia.data.Region*, str]] = None*) → Union[None, str]

class cassiopeia.**Versions**(**args*, ***kwargs*)

Bases: cassiopeia.core.common.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[merakicommons.container.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 your 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 a 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
            },
            "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": {
        "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 `merakicommons` 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) snippit 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 `DataSource`s and `DataSink`s. 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": {
```

(continues on next page)

(continued from previous page)

```
      "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),
```

(continues on next page)

(continued from previous page)

```
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),
FfeaturedMatches: 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
    }
    ...
```

(continues on next page)

(continued from previous page)

```
  }
}
```

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,
```

(continues on next page)

(continued from previous page)

```
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 datasources 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*), 29
account_id (*cassiopeia.Summoner attribute*), 54
after_id (*cassiopeia.core.match.Event attribute*), 45
all_random_summoners_rift (*cassiopeia.data.GameMode attribute*), 9
all_random_summoners_rift (*cassiopeia.data.Queue attribute*), 11
all_random_urf (*cassiopeia.data.Queue attribute*), 11
all_random_urf_snow (*cassiopeia.data.GameMode attribute*), 9
all_random_urf_snow (*cassiopeia.data.Queue attribute*), 11
ally_tips (*cassiopeia.Champion attribute*), 18
alternative_images (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
alternative_images (*cassiopeia.SummonerSpell attribute*), 56
americas (*cassiopeia.data.Continent attribute*), 9
apex (*cassiopeia.data.Position attribute*), 11
append() (*cassiopeia.ChampionMasteries method*), 25
append() (*cassiopeia.Champions method*), 17
append() (*cassiopeia.core.league.LeagueEntries method*), 33
append() (*cassiopeia.core.league.LeagueSummonerEntries method*), 31
append() (*cassiopeia.core.match.MatchHistory method*), 38
append() (*cassiopeia.FeaturedMatches method*), 52
append() (*cassiopeia.Items method*), 27
append() (*cassiopeia.Locales method*), 35
append() (*cassiopeia.Maps method*), 36
append() (*cassiopeia.ProfileIcons method*), 47
append() (*cassiopeia.Runes method*), 49
append() (*cassiopeia.SummonerSpells method*), 55
append() (*cassiopeia.Versions method*), 58
aram (*cassiopeia.data.GameMode attribute*), 9
aram (*cassiopeia.data.Queue attribute*), 12
aram_butchers_bridge (*cassiopeia.data.Queue attribute*), 12
armor (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
armor (*cassiopeia.core.staticdata.item.ItemStats attribute*), 29
armor_per_level (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
ascended_type (*cassiopeia.core.match.Event attribute*), 45
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*), 45
assists (*cassiopeia.core.match.ParticipantState attribute*), 46
assists (*cassiopeia.core.match.ParticipantStats attribute*), 41
assists (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
assists (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
attack (*cassiopeia.core.staticdata.champion.Info attribute*), 20
attack_damage (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
attack_damage (*cassiopeia.core.staticdata.item.ItemStats attribute*), 29
attack_damage_per_level (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
attack_range (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
attack_speed (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
attack_speed (*cassiopeia.core.staticdata.item.ItemStats attribute*), 29

B

ban_rate (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23

ban_rates (*cassiopeia.Champion attribute*), 18
bans (*cassiopeia.core.match.Team attribute*), 40
bans (*cassiopeia.core.spectator.Team attribute*), 53
baron_kills (*cassiopeia.core.match.ParticipantStats attribute*), 41
baron_kills (*cassiopeia.core.match.Team attribute*), 40
BASE (*cassiopeia.data.Tower attribute*), 17
before_id (*cassiopeia.core.match.Event attribute*), 45
begin_index (*cassiopeia.core.match.MatchHistory attribute*), 38
begin_time (*cassiopeia.core.match.MatchHistory attribute*), 38
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*), 29
blood_hunt_assassin (*cassiopeia.data.Queue attribute*), 12
blood_well (*cassiopeia.data.Resource attribute*), 14
blue (*cassiopeia.data.Side attribute*), 16
blue_team (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
blue_team (*cassiopeia.Match attribute*), 39
blurb (*cassiopeia.Champion attribute*), 18
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*), 11
bounty_level (*cassiopeia.core.match.ParticipantStats attribute*), 41
brazil (*cassiopeia.data.Platform attribute*), 11
brazil (*cassiopeia.data.Region attribute*), 14
bronze (*cassiopeia.data.Tier attribute*), 16
building_type (*cassiopeia.core.match.Event attribute*), 45
builds_from (*cassiopeia.Item attribute*), 28
builds_into (*cassiopeia.Item attribute*), 28

C

captured_point (*cassiopeia.core.match.Event attribute*), 45
cassiopeia.data
    module, 9
cdn (*cassiopeia.Realms attribute*), 49
challenger (*cassiopeia.data.Tier attribute*), 16
ChallengerLeague (*class in cassiopeia.core*), 31
champion (*cassiopeia.ChampionMastery attribute*), 26
champion (*cassiopeia.core.match.Participant attribute*), 41
champion (*cassiopeia.core.spectator.Participant attribute*), 54
champion (*cassiopeia.Item attribute*), 28
champion (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
Champion (*class in cassiopeia*), 18
champion_assists (*cassiopeia.core.match.ParticipantTimeline attribute*), 44
champion_deaths (*cassiopeia.core.match.ParticipantTimeline attribute*), 44
champion_experience (*cassiopeia.core.match.ParticipantStats attribute*), 41
champion_key (*cassiopeia.core.staticdata.champion.Skin attribute*), 21
champion_kills (*cassiopeia.core.match.ParticipantTimeline attribute*), 44
champion_masteries (*cassiopeia.Summoner attribute*), 54
champion_transform (*cassiopeia.core.match.ParticipantStats attribute*), 41
championgg_metadata (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
ChampionGGMatchup (*class in cassiopeia_championgg.core*), 24
ChampionGGMatchups (*class in cassiopeia_championgg.core*), 24
ChampionGGMatchupStats (*class in cassiopeia_championgg.core*), 24
ChampionGGStats (*class in cassiopeia_championgg.core*), 23
ChampionMasteries (*class in cassiopeia*), 25
ChampionMastery (*class in cassiopeia*), 26
Champions (*class in cassiopeia*), 17
ChampionSpell (*class in cassiopeia.core.staticdata.champion*), 22
chest_granted (*cassiopeia.ChampionMastery attribute*), 26
clash (*cassiopeia.data.Queue attribute*), 12
classic (*cassiopeia.data.GameMode attribute*), 9
clear() (*cassiopeia.ChampionMasteries method*), 25
clear() (*cassiopeia.Champions method*), 17
clear() (*cassiopeia.core.league.LeagueEntries method*), 33
clear() (*cassiopeia.core.league.LeagueSummonerEntries method*), 31
clear() (*cassiopeia.core.match.MatchHistory method*),

38
clear() (*cassiopeia.FeaturedMatches method*), 52
clear() (*cassiopeia.Items method*), 27
clear() (*cassiopeia.Locales method*), 35
clear() (*cassiopeia.Maps method*), 36
clear() (*cassiopeia.ProfileIcons method*), 47
clear() (*cassiopeia.Runes method*), 49
clear() (*cassiopeia.SummonerSpells method*), 55
clear() (*cassiopeia.Versions method*), 58
clear_sinks() (*cassiopeia._configuration.settings.Settings method*), 8
client_refresh_interval (*cassiopeia.FeaturedMatches attribute*), 52
coefficients (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 23
coefficients (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 57
consumables_purchased (*cassiopeia.core.match.ParticipantStats attribute*), 42
consume_on_full (*cassiopeia.Item attribute*), 28
consumed (*cassiopeia.Item attribute*), 28
contains() (*cassiopeia.ChampionMasteries method*), 25
contains() (*cassiopeia.Champions method*), 17
contains() (*cassiopeia.core.league.LeagueEntries method*), 33
contains() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
contains() (*cassiopeia.core.match.MatchHistory method*), 38
contains() (*cassiopeia.FeaturedMatches method*), 52
contains() (*cassiopeia.Items method*), 27
contains() (*cassiopeia.Locales method*), 35
contains() (*cassiopeia.Maps method*), 36
contains() (*cassiopeia.ProfileIcons method*), 47
contains() (*cassiopeia.Runes method*), 49
contains() (*cassiopeia.SummonerSpells method*), 55
contains() (*cassiopeia.Versions method*), 58
continent (*cassiopeia.core.match.Timeline attribute*), 44
continent (*cassiopeia.data.Platform attribute*), 11
continent (*cassiopeia.data.Region attribute*), 14
continent (*cassiopeia.Match attribute*), 39
Continent (*class in cassiopeia.data*), 9
continent() (*cassiopeia.core.match.MatchHistory method*), 38
cooldowns (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
cooldowns (*cassiopeia.SummonerSpell attribute*), 56
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*), 25
copy() (*cassiopeia.Champions method*), 17
copy() (*cassiopeia.core.league.LeagueEntries method*), 33
copy() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
copy() (*cassiopeia.core.match.MatchHistory method*), 38
copy() (*cassiopeia.FeaturedMatches method*), 52
copy() (*cassiopeia.Items method*), 27
copy() (*cassiopeia.Locales method*), 35
copy() (*cassiopeia.Maps method*), 36
copy() (*cassiopeia.ProfileIcons method*), 47
copy() (*cassiopeia.Runes method*), 49
copy() (*cassiopeia.SummonerSpells method*), 55
copy() (*cassiopeia.Versions method*), 58
costs (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
costs (*cassiopeia.SummonerSpell attribute*), 56
count() (*cassiopeia.ChampionMasteries method*), 25
count() (*cassiopeia.Champions method*), 17
count() (*cassiopeia.core.league.LeagueEntries method*), 33
count() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
count() (*cassiopeia.core.match.MatchHistory method*), 38
count() (*cassiopeia.FeaturedMatches method*), 52
count() (*cassiopeia.Items method*), 27
count() (*cassiopeia.Locales method*), 35
count() (*cassiopeia.Maps method*), 36
count() (*cassiopeia.ProfileIcons method*), 47
count() (*cassiopeia.Runes method*), 49
count() (*cassiopeia.SummonerSpells method*), 55
count() (*cassiopeia.Versions method*), 58
courage (*cassiopeia.data.Resource attribute*), 14
creation (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
creation (*cassiopeia.Match attribute*), 39
creator_id (*cassiopeia.core.match.Event attribute*), 45
creep_score (*cassiopeia.core.match.ParticipantFrame attribute*), 44
creep_score (*cassiopeia.core.match.ParticipantState attribute*), 46
crimson_rush (*cassiopeia.data.Resource attribute*), 14
critical_strike_chance (*cas-*

siopeia.core.staticdata.champion.Stats attribute), 20
critical_strike_chance (*cassiopeia.core.staticdata.item.ItemStats attribute*), 29
critical_strike_chance_per_level (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
critical_strike_damage (*cassiopeia.core.staticdata.item.ItemStats attribute*), 29
css_version (*cassiopeia.Realms attribute*), 49
cumulative_timeline (*cassiopeia.core.match.Participant attribute*), 41
CumulativeTimeline (*class in cassiopeia.core.match*), 45
cunning (*cassiopeia.data.MasteryTree attribute*), 10
current_gold (*cassiopeia.core.match.ParticipantFrame attribute*), 44
current_gold (*cassiopeia.core.match.ParticipantState attribute*), 46
current_match (*cassiopeia.cassiopeia.Summoner attribute*), 52
current_match (*cassiopeia.Summoner attribute*), 54
CurrentMatch (*class in cassiopeia.core.spectator*), 53
custom (*cassiopeia.data.GameType attribute*), 10
custom (*cassiopeia.data.Queue attribute*), 12

D

damage_composition (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
damage_dealt_to_buildings (*cassiopeia.core.match.ParticipantStats attribute*), 42
damage_dealt_to_objectives (*cassiopeia.core.match.ParticipantStats attribute*), 42
damage_dealt_to_turrets (*cassiopeia.core.match.ParticipantStats attribute*), 42
damage_self_mitigated (*cassiopeia.core.match.ParticipantStats attribute*), 42
dark_star (*cassiopeia.data.GameMode attribute*), 9
dark_star (*cassiopeia.data.Queue attribute*), 12
deaths (*cassiopeia.core.match.ParticipantState attribute*), 46
deaths (*cassiopeia.core.match.ParticipantStats attribute*), 42
deaths (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
deaths (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
default_locale (*cassiopeia.data.Platform attribute*), 11
default_locale (*cassiopeia.data.Region attribute*), 14
defense (*cassiopeia.core.staticdata.champion.Info attribute*), 20
definitely_not_dominion (*cassiopeia.data.Queue attribute*), 12
delete() (*cassiopeia.ChampionMasteries method*), 25
delete() (*cassiopeia.Champions method*), 17
delete() (*cassiopeia.core.league.LeagueEntries method*), 33
delete() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
delete() (*cassiopeia.core.match.MatchHistory method*), 38
delete() (*cassiopeia.FeaturedMatches method*), 52
delete() (*cassiopeia.Items method*), 27
delete() (*cassiopeia.Locales method*), 35
delete() (*cassiopeia.Maps method*), 36
delete() (*cassiopeia.ProfileIcons method*), 47
delete() (*cassiopeia.Runes method*), 49
delete() (*cassiopeia.SummonerSpells method*), 55
delete() (*cassiopeia.Versions method*), 58
delta_assists (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_deaths (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_gold_earned (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_killing_sprees (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_kills (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_minions_killed (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_neutral_minions_killed_team_jungle (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_ten_to_twenty (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_thirty_to_end (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_total_damage_dealt_to_champions (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 24
delta_twenty_to_thirty (*cassiopeia_championgg.core.ChampionGGMatchupStats*

attribute), 24
delta_weighted_score (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
delta_wins (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
delta_zero_to_ten (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
deprecated_all_random_urf (*cassiopeia.data.Queue attribute*), 12
deprecated_aram (*cassiopeia.data.Queue attribute*), 12
deprecated_blind_dominion (*cassiopeia.data.Queue attribute*), 12
deprecated_blind_fives (*cassiopeia.data.Queue attribute*), 12
deprecated_blind_threes (*cassiopeia.data.Queue attribute*), 12
deprecated_coop_ai_beginner_fives (*cassiopeia.data.Queue attribute*), 12
deprecated_coop_ai_dominion (*cassiopeia.data.Queue attribute*), 12
deprecated_coop_ai_fives (*cassiopeia.data.Queue attribute*), 12
deprecated_coop_ai_intermediate_fives (*cassiopeia.data.Queue attribute*), 12
deprecated_coop_ai_intro_fives (*cassiopeia.data.Queue attribute*), 12
deprecated_coop_ai_threes (*cassiopeia.data.Queue attribute*), 12
deprecated_doom_bots_rank_1 (*cassiopeia.data.Queue attribute*), 12
deprecated_doom_bots_rank_2 (*cassiopeia.data.Queue attribute*), 12
deprecated_doom_bots_rank_5 (*cassiopeia.data.Queue attribute*), 12
deprecated_draft_dominion (*cassiopeia.data.Queue attribute*), 12
deprecated_draft_fives (*cassiopeia.data.Queue attribute*), 12
deprecated_nexus_blitz (*cassiopeia.data.Queue attribute*), 12
deprecated_nexus_siege (*cassiopeia.data.Queue attribute*), 12
deprecated_poro_king (*cassiopeia.data.Queue attribute*), 12
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
deprecated_team_builder_fives (*cassiopeia.data.Queue attribute*), 13
description (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
description (*cassiopeia.core.staticdata.champion.Passive attribute*), 21
description (*cassiopeia.Item attribute*), 28
description (*cassiopeia.SummonerSpell attribute*), 56
diamond (*cassiopeia.data.Tier attribute*), 16
difficulty (*cassiopeia.core.staticdata.champion.Info attribute*), 20
division (*cassiopeia.core.league.LeagueEntries attribute*), 33
division (*cassiopeia.core.league.LeagueEntry attribute*), 34
Division (*class in cassiopeia.data*), 9
dodge (*cassiopeia.core.staticdata.item.ItemStats attribute*), 29
domination (*cassiopeia.Runes attribute*), 49
dominion (*cassiopeia.data.GameMode attribute*), 9
dominion_score (*cassiopeia.core.match.ParticipantFrame attribute*), 44
dominion_score (*cassiopeia.core.match.ParticipantState attribute*), 46
dominion_score (*cassiopeia.core.match.Team attribute*), 40
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*), 42
dragon_kills (*cassiopeia.core.match.ParticipantStats attribute*), 42
dragon_kills (*cassiopeia.core.match.Team attribute*), 40
duo (*cassiopeia.data.Role attribute*), 15
duo_carry (*cassiopeia.data.Role attribute*), 15
duo_support (*cassiopeia.data.Role attribute*), 15
duration (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
duration (*cassiopeia.Match attribute*), 39
dynamic (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 23
dynamic (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 57

E

E (*cassiopeia.data.Key attribute*), 10
effect (*cassiopeia.Item attribute*), 28
effects (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
effects (*cassiopeia.SummonerSpell attribute*), 56
effects_by_level (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
elo (*cassiopeia_championgg.core.ChampionGGMatchup attribute*), 24
elo (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
end (*cassiopeia.Patch attribute*), 46
end() (*cassiopeia.data.Season method*), 15
end_index (*cassiopeia.core.match.MatchHistory attribute*), 38
end_time (*cassiopeia.core.match.MatchHistory attribute*), 38
ended_in_early_surrender (*cassiopeia.core.match.Participant attribute*), 41
enemy (*cassiopeia_championgg.core.ChampionGGMatchup attribute*), 24
enemy_team (*cassiopeia.core.match.Participant attribute*), 41
enemy_tips (*cassiopeia.Champion attribute*), 19
energy (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
energy (*cassiopeia.data.Resource attribute*), 14
energy_regen (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
entries (*cassiopeia.core.ChallengerLeague attribute*), 31
entries (*cassiopeia.core.league.League attribute*), 31
entries (*cassiopeia.core.MasterLeague attribute*), 31
enumerate() (*cassiopeia.ChampionMasteries method*), 25
enumerate() (*cassiopeia.Champions method*), 17
enumerate() (*cassiopeia.core.league.LeagueEntries method*), 33
enumerate() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
enumerate() (*cassiopeia.core.match.MatchHistory method*), 38
enumerate() (*cassiopeia.FeaturedMatches method*), 52
enumerate() (*cassiopeia.Items method*), 27
enumerate() (*cassiopeia.Locales method*), 35
enumerate() (*cassiopeia.Maps method*), 36
enumerate() (*cassiopeia.ProfileIcons method*), 47
enumerate() (*cassiopeia.Runes method*), 49
enumerate() (*cassiopeia.SummonerSpells method*), 55
enumerate() (*cassiopeia.Versions method*), 58
europe (*cassiopeia.data.Continent attribute*), 9
europe_north_east (*cassiopeia.data.Platform attribute*), 11
europe_north_east (*cassiopeia.data.Region attribute*), 14
europe_west (*cassiopeia.data.Platform attribute*), 11
europe_west (*cassiopeia.data.Region attribute*), 14
Event (*class in cassiopeia.core.match*), 45
events (*cassiopeia.core.match.Frame attribute*), 44
events (*cassiopeia.core.match.ParticipantTimeline attribute*), 44
exists (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
exists (*cassiopeia.Match attribute*), 39
exists (*cassiopeia.Summoner attribute*), 54
experience (*cassiopeia.core.match.ParticipantFrame attribute*), 44
experience (*cassiopeia.core.match.ParticipantState attribute*), 46
expire_sinks() (*cassiopeia._configuration.settings.Settings method*), 8
extend() (*cassiopeia.ChampionMasteries method*), 25
extend() (*cassiopeia.Champions method*), 17
extend() (*cassiopeia.core.league.LeagueEntries method*), 33
extend() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
extend() (*cassiopeia.core.match.MatchHistory method*), 38
extend() (*cassiopeia.FeaturedMatches method*), 52
extend() (*cassiopeia.Items method*), 27
extend() (*cassiopeia.Locales method*), 35
extend() (*cassiopeia.Maps method*), 36
extend() (*cassiopeia.ProfileIcons method*), 47
extend() (*cassiopeia.Runes method*), 49
extend() (*cassiopeia.SummonerSpells method*), 55
extend() (*cassiopeia.Versions method*), 58

F

FeaturedMatches (*class in cassiopeia*), 52
ferocity (*cassiopeia.data.MasteryTree attribute*), 10
ferocity (*cassiopeia.data.Resource attribute*), 14
filter() (*cassiopeia.ChampionMasteries method*), 26
filter() (*cassiopeia.Champions method*), 17
filter() (*cassiopeia.core.league.LeagueEntries method*), 33
filter() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
filter() (*cassiopeia.core.match.MatchHistory method*), 38
filter() (*cassiopeia.FeaturedMatches method*), 52
filter() (*cassiopeia.Items method*), 27
filter() (*cassiopeia.Locales method*), 35
filter() (*cassiopeia.Maps method*), 36

filter() (*cassiopeia.ProfileIcons method*), 47
filter() (*cassiopeia.Runes method*), 50
filter() (*cassiopeia.SummonerSpells method*), 55
filter() (*cassiopeia.Versions method*), 58
find() (*cassiopeia.ChampionMasteries method*), 26
find() (*cassiopeia.Champions method*), 17
find() (*cassiopeia.core.league.LeagueEntries method*), 33
find() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
find() (*cassiopeia.core.match.MatchHistory method*), 38
find() (*cassiopeia.FeaturedMatches method*), 52
find() (*cassiopeia.Items method*), 27
find() (*cassiopeia.Locales method*), 35
find() (*cassiopeia.Maps method*), 36
find() (*cassiopeia.ProfileIcons method*), 47
find() (*cassiopeia.Runes method*), 50
find() (*cassiopeia.SummonerSpells method*), 55
find() (*cassiopeia.Versions method*), 58
first_baron (*cassiopeia.core.match.Team attribute*), 40
first_blood (*cassiopeia.core.match.Team attribute*), 40
first_blood_assist (*cassiopeia.core.match.ParticipantStats attribute*), 42
first_blood_kill (*cassiopeia.core.match.ParticipantStats attribute*), 42
first_dragon (*cassiopeia.core.match.Team attribute*), 40
first_inhibitor (*cassiopeia.core.match.Team attribute*), 40
first_rift_herald (*cassiopeia.core.match.Team attribute*), 40
first_tower (*cassiopeia.core.match.Team attribute*), 40
first_tower_assist (*cassiopeia.core.match.ParticipantStats attribute*), 42
first_tower_fallen (*cassiopeia.core.match.Timeline attribute*), 44
first_tower_kill (*cassiopeia.core.match.ParticipantStats attribute*), 42
fives (*cassiopeia.core.league.LeagueSummonerEntries attribute*), 32
flex (*cassiopeia.core.league.LeagueSummonerEntries attribute*), 32
flow (*cassiopeia.data.Resource attribute*), 15
four (*cassiopeia.data.Division attribute*), 9
Frame (*class in cassiopeia.core.match*), 44
frame_interval (*cassiopeia.core.match.Timeline attribute*), 44
frames (*cassiopeia.core.match.ParticipantTimeline attribute*), 44
frames (*cassiopeia.core.match.Timeline attribute*), 44
free_to_play (*cassiopeia.Champion attribute*), 19
free_to_play_new_players (*cassiopeia.Champion attribute*), 19
fresh_blood (*cassiopeia.core.league.LeagueEntry attribute*), 34
from_data() (*cassiopeia.ChampionMasteries class method*), 26
from_data() (*cassiopeia.Champions class method*), 17
from_data() (*cassiopeia.core.league.LeagueEntries class method*), 33
from_data() (*cassiopeia.core.league.LeagueEntry class method*), 34
from_data() (*cassiopeia.core.league.LeagueSummonerEntries class method*), 32
from_data() (*cassiopeia.core.match.MatchHistory class method*), 38
from_data() (*cassiopeia.core.match.Participant class method*), 41
from_data() (*cassiopeia.core.match.ParticipantStats class method*), 42
from_data() (*cassiopeia.core.match.ParticipantTimeline class method*), 44
from_data() (*cassiopeia.core.match.Team class method*), 40
from_data() (*cassiopeia.core.spectator.CurrentMatch class method*), 53
from_data() (*cassiopeia.core.spectator.Participant class method*), 54
from_data() (*cassiopeia.core.spectator.Team class method*), 53
from_data() (*cassiopeia.core.staticdata.champion.ItemSet class method*), 22
from_data() (*cassiopeia.core.staticdata.champion.RecommendedItems class method*), 21
from_data() (*cassiopeia.FeaturedMatches class method*), 52
from_data() (*cassiopeia.Items class method*), 27
from_data() (*cassiopeia.Locales class method*), 35
from_data() (*cassiopeia.Maps class method*), 36
from_data() (*cassiopeia.ProfileIcons class method*), 47
from_data() (*cassiopeia.Runes class method*), 50
from_data() (*cassiopeia.SummonerSpells class method*), 55
from_data() (*cassiopeia.Versions class method*), 58
from_date() (*cassiopeia.Patch class method*), 46
from_generator() (*cassiopeia.ChampionMasteries class method*), 26
from_generator() (*cassiopeia.Champions class method*), 17
from_generator() (*cassiopeia.core.league.LeagueEntries class method*), 33
from_generator() (*cas-*

siopeia.core.league.LeagueSummonerEntries class method), 32
from_generator() (*cassiopeia.core.match.MatchHistory class method*), 39
from_generator() (*cassiopeia.FeaturedMatches class method*), 52
from_generator() (*cassiopeia.Items class method*), 27
from_generator() (*cassiopeia.Locales class method*), 35
from_generator() (*cassiopeia.Maps class method*), 37
from_generator() (*cassiopeia.ProfileIcons class method*), 47
from_generator() (*cassiopeia.Runes class method*), 50
from_generator() (*cassiopeia.SummonerSpells class method*), 55
from_generator() (*cassiopeia.Versions class method*), 58
from_id() (*cassiopeia.data.Queue method*), 13
from_id() (*cassiopeia.data.Season method*), 15
from_league_naming_scheme() (*cassiopeia.data.Position method*), 11
from_match_naming_scheme() (*cassiopeia.data.Lane method*), 10
from_match_naming_scheme() (*cassiopeia.data.Role method*), 15
from_match_reference() (*cassiopeia.Match class method*), 39
from_platform() (*cassiopeia.data.Region static method*), 14
from_position() (*cassiopeia.data.SummonersRiftArea static method*), 16
from_region() (*cassiopeia.data.Platform static method*), 11
from_str() (*cassiopeia.Patch class method*), 46
fury (*cassiopeia.data.Resource attribute*), 15

G

game_type (*cassiopeia.Match attribute*), 40
GameMode (*class in cassiopeia.data*), 9
games_played (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
GameType (*class in cassiopeia.data*), 10
get_challenger_league() (*cassiopeia.cassiopeia method*), 31
get_champion() (*cassiopeia.cassiopeia method*), 17
get_champion_masteries() (*cassiopeia.cassiopeia method*), 25
get_champion_mastery() (*cassiopeia.cassiopeia method*), 25
get_champions() (*cassiopeia.cassiopeia method*), 17
get_current_match() (*cassiopeia.cassiopeia method*), 52
get_featured_matches() (*cassiopeia.cassiopeia method*), 52
get_items() (*cassiopeia.cassiopeia method*), 27
get_lane() (*cassiopeia.data.SummonersRiftArea method*), 16
get_language_strings() (*cassiopeia.cassiopeia method*), 30
get_locales() (*cassiopeia.cassiopeia method*), 35
get_maps() (*cassiopeia.cassiopeia method*), 36
get_master_league() (*cassiopeia.cassiopeia method*), 31
get_match() (*cassiopeia.cassiopeia method*), 38
get_match_history() (*cassiopeia.cassiopeia method*), 38
get_profile_icons() (*cassiopeia.cassiopeia method*), 47
get_realms() (*cassiopeia.cassiopeia method*), 49
get_runes() (*cassiopeia.cassiopeia method*), 49
get_side() (*cassiopeia.data.SummonersRiftArea method*), 16
get_status() (*cassiopeia.cassiopeia method*), 51
get_summoner() (*cassiopeia.cassiopeia method*), 54
get_summoner_spells() (*cassiopeia.cassiopeia method*), 55
get_version() (*cassiopeia.cassiopeia method*), 58
get_versions() (*cassiopeia.cassiopeia method*), 58
gold (*cassiopeia.data.Tier attribute*), 16
gold (*cassiopeia.Item attribute*), 29
gold_earned (*cassiopeia.core.match.ParticipantFrame attribute*), 44
gold_earned (*cassiopeia.core.match.ParticipantState attribute*), 46
gold_earned (*cassiopeia.core.match.ParticipantStats attribute*), 42
gold_earned (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
gold_earned (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
gold_spent (*cassiopeia.core.match.ParticipantStats attribute*), 42
grandmaster (*cassiopeia.data.Tier attribute*), 17
group (*cassiopeia.Item attribute*), 29
guardian_invasion_normal (*cassiopeia.data.Queue attribute*), 13
guardian_invasion_onslaught (*cassiopeia.data.Queue attribute*), 13

H

health (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
health (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
health_per_level (*cassiopeia.core.staticdata.champion.Stats at-*

tribute), 20
health_regen (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
health_regen (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
health_regen_per_level (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
heat (*cassiopeia.data.Resource attribute*), 15
hexakill_summoners_rift (*cassiopeia.data.Queue attribute*), 13
hexakill_twisted_treeline (*cassiopeia.data.Queue attribute*), 13
hide (*cassiopeia.Item attribute*), 29
hostname (*cassiopeia.ShardStatus attribute*), 51
hot_streak (*cassiopeia.core.league.LeagueEntry attribute*), 34

I

id (*cassiopeia.Champion attribute*), 19
id (*cassiopeia.core.ChallengerLeague attribute*), 31
id (*cassiopeia.core.league.League attribute*), 31
id (*cassiopeia.core.MasterLeague attribute*), 31
id (*cassiopeia.core.match.Participant attribute*), 41
id (*cassiopeia.core.match.Timeline attribute*), 44
id (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
id (*cassiopeia.core.staticdata.champion.Skin attribute*), 21
id (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
id (*cassiopeia.data.Queue attribute*), 13
id (*cassiopeia.data.Season attribute*), 15
id (*cassiopeia.Item attribute*), 29
id (*cassiopeia.Map attribute*), 37
id (*cassiopeia.Match attribute*), 40
id (*cassiopeia.Rune attribute*), 51
id (*cassiopeia.Summoner attribute*), 54
id (*cassiopeia.SummonerSpell attribute*), 56
id (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
id (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
image (*cassiopeia.Champion attribute*), 19
image (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
image (*cassiopeia.Item attribute*), 29
image (*cassiopeia.Map attribute*), 37
image (*cassiopeia.Rune attribute*), 51
image (*cassiopeia.SummonerSpell attribute*), 56
image_info (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
image_info (*cassiopeia.core.staticdata.champion.Passive attribute*), 21
in_store (*cassiopeia.Item attribute*), 29
inactive (*cassiopeia.core.league.LeagueEntry attribute*), 34
included_data (*cassiopeia.Champion attribute*), 19
included_data (*cassiopeia.Champions attribute*), 18
included_data (*cassiopeia.Item attribute*), 29
included_data (*cassiopeia.Items attribute*), 27
included_data (*cassiopeia.Rune attribute*), 51
included_data (*cassiopeia.Runes attribute*), 50
included_data (*cassiopeia.SummonerSpell attribute*), 56
included_data (*cassiopeia.SummonerSpells attribute*), 55
index() (*cassiopeia.ChampionMasteries method*), 26
index() (*cassiopeia.Champions method*), 18
index() (*cassiopeia.core.league.LeagueEntries method*), 33
index() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
index() (*cassiopeia.core.match.MatchHistory method*), 39
index() (*cassiopeia.FeaturedMatches method*), 52
index() (*cassiopeia.Items method*), 28
index() (*cassiopeia.Locales method*), 35
index() (*cassiopeia.Maps method*), 37
index() (*cassiopeia.ProfileIcons method*), 47
index() (*cassiopeia.Runes method*), 50
index() (*cassiopeia.SummonerSpells method*), 55
index() (*cassiopeia.Versions method*), 58
individual_position (*cassiopeia.core.match.Participant attribute*), 41
info (*cassiopeia.Champion attribute*), 19
Info (*class in cassiopeia.core.staticdata.champion*), 20
inhibitor_kills (*cassiopeia.core.match.ParticipantStats attribute*), 42
inhibitor_kills (*cassiopeia.core.match.Team attribute*), 40
inhibitor_takedowns (*cassiopeia.core.match.ParticipantStats attribute*), 42
inhibitors_lost (*cassiopeia.core.match.ParticipantStats attribute*), 42
INNER (*cassiopeia.data.Tower attribute*), 17
insert() (*cassiopeia.ChampionMasteries method*), 26
insert() (*cassiopeia.Champions method*), 18
insert() (*cassiopeia.core.league.LeagueEntries method*), 33
insert() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
insert() (*cassiopeia.core.match.MatchHistory method*), 39
insert() (*cassiopeia.FeaturedMatches method*), 52

insert() (*cassiopeia.Items method*), 28
insert() (*cassiopeia.Locales method*), 35
insert() (*cassiopeia.Maps method*), 37
insert() (*cassiopeia.ProfileIcons method*), 47
insert() (*cassiopeia.Runes method*), 50
insert() (*cassiopeia.SummonerSpells method*), 55
insert() (*cassiopeia.Versions method*), 58
inspiration (*cassiopeia.Runes attribute*), 50
iron (*cassiopeia.data.Tier attribute*), 17
is_bot (*cassiopeia.core.match.Participant attribute*), 41
is_bot (*cassiopeia.core.spectator.Participant attribute*), 54
is_keystone (*cassiopeia.Rune attribute*), 51
is_remake (*cassiopeia.Match attribute*), 40
Item (*class in cassiopeia*), 28
item_id (*cassiopeia.core.match.Event attribute*), 45
item_sets (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 21
items (*cassiopeia.core.match.ParticipantState attribute*), 46
items (*cassiopeia.core.match.ParticipantStats attribute*), 42
items (*cassiopeia.core.staticdata.champion.ItemSet attribute*), 22
Items (*class in cassiopeia*), 27
items_purchased (*cassiopeia.core.match.ParticipantStats attribute*), 42
ItemSet (*class in cassiopeia.core.staticdata.champion*), 22
ItemStats (*class in cassiopeia.core.staticdata.item*), 29

J

japan (*cassiopeia.data.Platform attribute*), 11
japan (*cassiopeia.data.Region attribute*), 14
jungle (*cassiopeia.data.Lane attribute*), 10
jungle (*cassiopeia.data.Position attribute*), 11
jungle_bot_blue (*cassiopeia.data.SummonersRiftArea attribute*), 16
jungle_bot_red (*cassiopeia.data.SummonersRiftArea attribute*), 16
jungle_top_blue (*cassiopeia.data.SummonersRiftArea attribute*), 16
jungle_top_red (*cassiopeia.data.SummonersRiftArea attribute*), 16

K

kda (*cassiopeia.core.match.ParticipantState attribute*), 46
kda (*cassiopeia.core.match.ParticipantStats attribute*), 42
key (*cassiopeia.Champion attribute*), 19
key (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
key (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 23
key (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 57
key (*cassiopeia.SummonerSpell attribute*), 56
Key (*class in cassiopeia.data*), 10
keyboard_key (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
keystones (*cassiopeia.Runes attribute*), 50
keywords (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
keywords (*cassiopeia.Item attribute*), 29
killer_id (*cassiopeia.core.match.Event attribute*), 45
killing_sprees (*cassiopeia.core.match.ParticipantStats attribute*), 42
killing_sprees (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
kills (*cassiopeia.core.match.ParticipantState attribute*), 46
kills (*cassiopeia.core.match.ParticipantStats attribute*), 42
kills (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
kills (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
kills_heatmap() (*cassiopeia.Match method*), 40
korea (*cassiopeia.data.Platform attribute*), 11
korea (*cassiopeia.data.Region attribute*), 14

L

lane (*cassiopeia.core.match.Participant attribute*), 41
Lane (*class in cassiopeia.data*), 10
lane_type (*cassiopeia.core.match.Event attribute*), 45
language (*cassiopeia.Realms attribute*), 49
LanguageStrings (*class in cassiopeia*), 30
largest_critical_strike (*cassiopeia.core.match.ParticipantStats attribute*), 42
largest_killing_spree (*cassiopeia.core.match.ParticipantStats attribute*), 42
largest_multi_kill (*cassiopeia.core.match.ParticipantStats attribute*), 42
last_played (*cassiopeia.ChampionMastery attribute*), 26
latest() (*cassiopeia.Patch class method*), 46
latest_data_dragon (*cassiopeia.Realms attribute*), 49
latest_versions (*cassiopeia.Realms attribute*), 49
latin_america_north (*cassiopeia.data.Platform attribute*), 11

latin_america_north (*cassiopeia.data.Region attribute*), 14
latin_america_south (*cassiopeia.data.Platform attribute*), 11
latin_america_south (*cassiopeia.data.Region attribute*), 14
league (*cassiopeia.core.league.LeagueEntry attribute*), 34
League (*class in cassiopeia.core.league*), 31
league_entries (*cassiopeia.Summoner attribute*), 54
league_points (*cassiopeia.core.league.LeagueEntry attribute*), 34
LeagueEntries (*class in cassiopeia.core.league*), 33
LeagueEntry (*class in cassiopeia.core.league*), 34
leagues (*cassiopeia.cassiopeia.Summoner attribute*), 31
LeagueSummonerEntries (*class in cassiopeia.core.league*), 31
legacy_mode (*cassiopeia.Realms attribute*), 49
level (*cassiopeia.ChampionMastery attribute*), 27
level (*cassiopeia.core.match.ParticipantFrame attribute*), 44
level (*cassiopeia.core.match.ParticipantState attribute*), 46
level (*cassiopeia.core.match.ParticipantStats attribute*), 42
level (*cassiopeia.Summoner attribute*), 54
level_up_type (*cassiopeia.core.match.Event attribute*), 45
life_steal (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
link (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 23
link (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 57
load() (*cassiopeia.Champion method*), 19
loading_image (*cassiopeia.core.staticdata.champion.Skin attribute*), 21
loading_image_url (*cassiopeia.core.staticdata.champion.Skin attribute*), 21
locale (*cassiopeia.Champion attribute*), 19
locale (*cassiopeia.Champions attribute*), 18
locale (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
locale (*cassiopeia.Item attribute*), 29
locale (*cassiopeia.Items attribute*), 28
locale (*cassiopeia.LanguageStrings attribute*), 30
locale (*cassiopeia.Map attribute*), 37
locale (*cassiopeia.Maps attribute*), 37
locale (*cassiopeia.ProfileIcons attribute*), 47
locale (*cassiopeia.Realms attribute*), 49
locale (*cassiopeia.Rune attribute*), 51
locale (*cassiopeia.Runes attribute*), 50
locale (*cassiopeia.SummonerSpell attribute*), 56
locale (*cassiopeia.SummonerSpells attribute*), 55
locales (*cassiopeia.ShardStatus attribute*), 51
Locales (*class in cassiopeia*), 35
location (*cassiopeia.core.match.Position attribute*), 45
long_description (*cassiopeia.Rune attribute*), 51
longest_time_spent_living (*cassiopeia.core.match.ParticipantStats attribute*), 42
lore (*cassiopeia.Champion attribute*), 19
losses (*cassiopeia.core.league.LeagueEntry attribute*), 34
losses (*cassiopeia.core.league.MiniSeries attribute*), 33

M

magic (*cassiopeia.core.staticdata.champion.Info attribute*), 20
magic_damage_dealt (*cassiopeia.core.match.ParticipantStats attribute*), 42
magic_damage_dealt_to_champions (*cassiopeia.core.match.ParticipantStats attribute*), 42
magic_damage_taken (*cassiopeia.core.match.ParticipantStats attribute*), 42
magic_resist (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
magic_resist (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
magic_resist_per_level (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
major (*cassiopeia.Patch attribute*), 46
majorminor (*cassiopeia.Patch attribute*), 46
mana (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
mana (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
mana (*cassiopeia.data.Resource attribute*), 15
mana_per_level (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
mana_regen (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
mana_regen (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
mana_regen_per_level (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
map (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
map (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 21
map (*cassiopeia.Match attribute*), 40

Map (*class in cassiopeia*), 37
maps (*cassiopeia.Item attribute*), 29
Maps (*class in cassiopeia*), 36
master (*cassiopeia.data.Tier attribute*), 17
MasterLeague (*class in cassiopeia.core*), 31
MasteryTree (*class in cassiopeia.data*), 10
Match (*class in cassiopeia*), 39
match_history (*cassiopeia.cassiopeia.Summoner attribute*), 38
match_history (*cassiopeia.Summoner attribute*), 54
match_history_uri (*cassiopeia.core.match.Participant attribute*), 41
match_history_uri (*cassiopeia.Summoner attribute*), 54
match_type() (*cassiopeia.core.match.MatchHistory method*), 39
matched (*cassiopeia.data.GameType attribute*), 10
MatchHistory (*class in cassiopeia.core.match*), 38
MatchType (*class in cassiopeia.data*), 10
matchups (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
max_profile_icon_id (*cassiopeia.Realms attribute*), 49
max_rank (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
max_rank (*cassiopeia.SummonerSpell attribute*), 56
max_stacks (*cassiopeia.Item attribute*), 29
me (*cassiopeia_championgg.core.ChampionGGMatchup attribute*), 24
mid_lane (*cassiopeia.data.Lane attribute*), 10
mid_lane_blue (*cassiopeia.data.SummonersRiftArea attribute*), 16
mid_lane_purple (*cassiopeia.data.SummonersRiftArea attribute*), 16
mid_lane_red (*cassiopeia.data.SummonersRiftArea attribute*), 16
middle (*cassiopeia.data.Position attribute*), 11
minions_killed (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
minions_killed (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
MiniSeries (*class in cassiopeia.core.league*), 33
minor (*cassiopeia.Patch attribute*), 46
mirror_mode_fives (*cassiopeia.data.Queue attribute*), 13
mode (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
mode (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 21
mode (*cassiopeia.Match attribute*), 40
modes (*cassiopeia.SummonerSpell attribute*), 56
module
    cassiopeia.data, 9
monster_sub_type (*cassiopeia.core.match.Event attribute*), 45
monster_type (*cassiopeia.core.match.Event attribute*), 45
movespeed (*cassiopeia.core.staticdata.champion.Stats attribute*), 20
movespeed (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30

N

name (*cassiopeia.Champion attribute*), 19
name (*cassiopeia.core.ChallengerLeague attribute*), 31
name (*cassiopeia.core.league.League attribute*), 31
name (*cassiopeia.core.MasterLeague attribute*), 31
name (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
name (*cassiopeia.core.staticdata.champion.Passive attribute*), 21
name (*cassiopeia.core.staticdata.champion.Skin attribute*), 21
name (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
name (*cassiopeia.Item attribute*), 29
name (*cassiopeia.Map attribute*), 37
name (*cassiopeia.Patch attribute*), 47
name (*cassiopeia.Rune attribute*), 51
name (*cassiopeia.ShardStatus attribute*), 51
name (*cassiopeia.Summoner attribute*), 54
name (*cassiopeia.SummonerSpell attribute*), 57
nemesis_draft (*cassiopeia.data.Queue attribute*), 13
neutral_minions_killed (*cassiopeia.core.match.ParticipantFrame attribute*), 44
neutral_minions_killed (*cassiopeia.core.match.ParticipantState attribute*), 46
neutral_minions_killed (*cassiopeia.core.match.ParticipantStats attribute*), 42
neutral_minions_killed_in_enemy_jungle (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
neutral_minions_killed_in_team_jungle (*cassiopeia_championgg.core.ChampionGGStats attribute*), 23
neutral_minions_killed_team_jungle (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
NEXUS (*cassiopeia.data.Tower attribute*), 17
nexus_blitz (*cassiopeia.data.GameMode attribute*), 9
nexus_blitz (*cassiopeia.data.Queue attribute*), 13

nexus_blue (*cassiopeia.data.SummonersRiftArea attribute*), 16
nexus_kills (*cassiopeia.core.match.ParticipantStats attribute*), 42
nexus_lost (*cassiopeia.core.match.ParticipantStats attribute*), 42
nexus_red (*cassiopeia.data.SummonersRiftArea attribute*), 16
nexus_siege (*cassiopeia.data.GameMode attribute*), 9
nexus_siege (*cassiopeia.data.Queue attribute*), 13
nexus_takedowns (*cassiopeia.core.match.ParticipantStats attribute*), 42
nmatches (*cassiopeia_championgg.core.ChampionGGMatchup attribute*), 24
no_cost (*cassiopeia.data.Resource attribute*), 15
none (*cassiopeia.data.Position attribute*), 11
none (*cassiopeia.data.Resource attribute*), 15
none (*cassiopeia.data.Role attribute*), 15
none (*cassiopeia.data.SummonersRiftArea attribute*), 16
normal (*cassiopeia.data.MatchType attribute*), 10
normal_draft_fives (*cassiopeia.data.Queue attribute*), 13
normal_tft (*cassiopeia.data.Queue attribute*), 13
north_america (*cassiopeia.data.Platform attribute*), 11
north_america (*cassiopeia.data.Region attribute*), 14
not_played (*cassiopeia.core.league.MiniSeries attribute*), 33
number (*cassiopeia.core.staticdata.champion.Skin attribute*), 21

O

objectives (*cassiopeia.core.match.ParticipantState attribute*), 46
objectives_stolen (*cassiopeia.core.match.ParticipantStats attribute*), 43
objectives_stolen_assists (*cassiopeia.core.match.ParticipantStats attribute*), 43
observer_key (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
oceania (*cassiopeia.data.Platform attribute*), 11
oceania (*cassiopeia.data.Region attribute*), 14
odyssey (*cassiopeia.data.GameMode attribute*), 9
odyssey_cadet (*cassiopeia.data.Queue attribute*), 13
odyssey_captain (*cassiopeia.data.Queue attribute*), 13
odyssey_crewmember (*cassiopeia.data.Queue attribute*), 13
odyssey_intro (*cassiopeia.data.Queue attribute*), 13
odyssey_onslaught (*cassiopeia.data.Queue attribute*), 13
one (*cassiopeia.data.Division attribute*), 9
one_for_all (*cassiopeia.data.GameMode attribute*), 9
one_for_all (*cassiopeia.data.Queue attribute*), 13
one_for_all_rapid (*cassiopeia.data.Queue attribute*), 13
OUTER (*cassiopeia.data.Tower attribute*), 17
overcharge (*cassiopeia.data.GameMode attribute*), 9
overcharge (*cassiopeia.data.Queue attribute*), 13

P

Participant (*class in cassiopeia.core.match*), 41
Participant (*class in cassiopeia.core.spectator*), 53
participant_frames (*cassiopeia.core.match.Frame attribute*), 44
participant_id (*cassiopeia.core.match.Event attribute*), 45
participant_id (*cassiopeia.core.match.ParticipantFrame attribute*), 44
ParticipantFrame (*class in cassiopeia.core.match*), 44
participants (*cassiopeia.core.match.Team attribute*), 40
participants (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
participants (*cassiopeia.core.spectator.Team attribute*), 53
participants (*cassiopeia.Match attribute*), 40
ParticipantState (*class in cassiopeia.core.match*), 46
ParticipantStats (*class in cassiopeia.core.match*), 41
ParticipantTimeline (*class in cassiopeia.core.match*), 44
passive (*cassiopeia.Champion attribute*), 19
Passive (*class in cassiopeia.core.staticdata.champion*), 21
patch (*cassiopeia.Match attribute*), 40
patch (*cassiopeia_championgg.core.ChampionGGMatchup attribute*), 24
patch (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
Patch (*class in cassiopeia*), 46
path (*cassiopeia.Rune attribute*), 51
penta_kills (*cassiopeia.core.match.ParticipantStats attribute*), 43
percent_ability_power (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_armor (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_attack_damage (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_attack_speed (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_attack_speed_per_level (*cassiopeia.core.staticdata.champion.Stats at-*

tribute), 21
percent_block (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_critical_strike_damage (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_health (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_health_regen (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_magic_resist (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_mana_regen (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_movespeed (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
percent_xp_bonus (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
performance_score (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
physical_damage_dealt (*cassiopeia.core.match.ParticipantStats attribute*), 43
physical_damage_dealt_to_champions (*cassiopeia.core.match.ParticipantStats attribute*), 43
physical_damage_taken (*cassiopeia.core.match.ParticipantStats attribute*), 43
pipeline (*cassiopeia._configuration.settings.Settings attribute*), 8
plaintext (*cassiopeia.Item attribute*), 29
platform (*cassiopeia.Champion attribute*), 19
platform (*cassiopeia.ChampionMasteries attribute*), 26
platform (*cassiopeia.ChampionMastery attribute*), 27
platform (*cassiopeia.Champions attribute*), 18
platform (*cassiopeia.core.ChallengerLeague attribute*), 31
platform (*cassiopeia.core.league.League attribute*), 31
platform (*cassiopeia.core.league.LeagueEntries attribute*), 34
platform (*cassiopeia.core.league.LeagueEntry attribute*), 34
platform (*cassiopeia.core.league.LeagueSummonerEntries attribute*), 32
platform (*cassiopeia.core.MasterLeague attribute*), 31
platform (*cassiopeia.core.match.MatchHistory attribute*), 39
platform (*cassiopeia.core.match.Timeline attribute*), 44
platform (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
platform (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
platform (*cassiopeia.data.Region attribute*), 14
platform (*cassiopeia.FeaturedMatches attribute*), 52
platform (*cassiopeia.Item attribute*), 29
platform (*cassiopeia.Items attribute*), 28
platform (*cassiopeia.LanguageStrings attribute*), 30
platform (*cassiopeia.Locales attribute*), 35
platform (*cassiopeia.Map attribute*), 37
platform (*cassiopeia.Maps attribute*), 37
platform (*cassiopeia.Match attribute*), 40
platform (*cassiopeia.ProfileIcons attribute*), 47
platform (*cassiopeia.Realms attribute*), 49
platform (*cassiopeia.Rune attribute*), 51
platform (*cassiopeia.Runes attribute*), 50
platform (*cassiopeia.ShardStatus attribute*), 51
platform (*cassiopeia.Summoner attribute*), 54
platform (*cassiopeia.SummonerSpell attribute*), 57
platform (*cassiopeia.SummonerSpells attribute*), 55
platform (*cassiopeia.Versions attribute*), 58
Platform (*class in cassiopeia.data*), 11
platinum (*cassiopeia.data.Tier attribute*), 17
play_rate (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
play_rate_by_role (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
play_rates (*cassiopeia.Champion attribute*), 19
plugins (*cassiopeia._configuration.settings.Settings attribute*), 8
points (*cassiopeia.ChampionMastery attribute*), 27
points_since_last_level (*cassiopeia.ChampionMastery attribute*), 27
points_until_next_level (*cassiopeia.ChampionMastery attribute*), 27
pop() (*cassiopeia.ChampionMasteries method*), 26
pop() (*cassiopeia.Champions method*), 18
pop() (*cassiopeia.core.league.LeagueEntries method*), 34
pop() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
pop() (*cassiopeia.core.match.MatchHistory method*), 39
pop() (*cassiopeia.FeaturedMatches method*), 52
pop() (*cassiopeia.Items method*), 28
pop() (*cassiopeia.Locales method*), 35
pop() (*cassiopeia.Maps method*), 37
pop() (*cassiopeia.ProfileIcons method*), 47
pop() (*cassiopeia.Runes method*), 50
pop() (*cassiopeia.SummonerSpells method*), 55
pop() (*cassiopeia.Versions method*), 58

poro_king (*cassiopeia.data.GameMode attribute*), 9
poro_king (*cassiopeia.data.Queue attribute*), 13
position (*cassiopeia.core.match.Event attribute*), 45
position (*cassiopeia.core.match.ParticipantFrame attribute*), 44
position (*cassiopeia.core.match.ParticipantState attribute*), 46
Position (*class in cassiopeia.core.match*), 45
Position (*class in cassiopeia.data*), 11
practice_tool (*cassiopeia.data.GameMode attribute*), 9
precision (*cassiopeia.Runes attribute*), 50
preseason_3 (*cassiopeia.data.Season attribute*), 15
preseason_4 (*cassiopeia.data.Season attribute*), 15
preseason_5 (*cassiopeia.data.Season attribute*), 15
preseason_6 (*cassiopeia.data.Season attribute*), 15
preseason_7 (*cassiopeia.data.Season attribute*), 15
preseason_8 (*cassiopeia.data.Season attribute*), 15
preseason_9 (*cassiopeia.data.Season attribute*), 15
priority (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 21
profile_icon (*cassiopeia.cassiopeia.Summoner attribute*), 47
profile_icon (*cassiopeia.Summoner attribute*), 54
ProfileIcon (*class in cassiopeia.core.staticdata.profileicon*), 48
ProfileIcons (*class in cassiopeia*), 47
progress (*cassiopeia.core.league.MiniSeries attribute*), 33
project (*cassiopeia.data.GameMode attribute*), 9
project (*cassiopeia.data.Queue attribute*), 13
promos (*cassiopeia.core.league.LeagueEntry attribute*), 34
puuid (*cassiopeia.Summoner attribute*), 54

Q

Q (*cassiopeia.data.Key attribute*), 10
quadra_kills (*cassiopeia.core.match.ParticipantStats attribute*), 43
queue (*cassiopeia.core.ChallengerLeague attribute*), 31
queue (*cassiopeia.core.league.League attribute*), 31
queue (*cassiopeia.core.league.LeagueEntries attribute*), 34
queue (*cassiopeia.core.league.LeagueEntry attribute*), 35
queue (*cassiopeia.core.MasterLeague attribute*), 31
queue (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
queue (*cassiopeia.Match attribute*), 40
Queue (*class in cassiopeia.data*), 11
queue() (*cassiopeia.core.match.MatchHistory method*), 39

R

R (*cassiopeia.data.Key attribute*), 10
rage (*cassiopeia.data.Resource attribute*), 15
range (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 22
range (*cassiopeia.SummonerSpell attribute*), 57
Rank (*class in cassiopeia.data*), 14
rank_last_season (*cassiopeia.core.match.Participant attribute*), 41
rank_last_season (*cassiopeia.Summoner attribute*), 54
ranked (*cassiopeia.data.MatchType attribute*), 10
ranked_flex_fives (*cassiopeia.data.Queue attribute*), 13
ranked_flex_threes (*cassiopeia.data.Queue attribute*), 13
ranked_solo_fives (*cassiopeia.data.Queue attribute*), 13
ranked_tft (*cassiopeia.data.Queue attribute*), 13
ranks (*cassiopeia.Summoner attribute*), 54
ranks_with (*cassiopeia.core.staticdata.champion.SpellVars attribute*), 23
ranks_with (*cassiopeia.core.staticdata.summonerspell.SpellVars attribute*), 57
Realms (*class in cassiopeia*), 49
rec_math (*cassiopeia.core.staticdata.champion.ItemSet attribute*), 22
recommended_itemsets (*cassiopeia.Champion attribute*), 19
RecommendedItems (*class in cassiopeia.core.staticdata.champion*), 21
red (*cassiopeia.data.Side attribute*), 16
red_team (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
red_team (*cassiopeia.Match attribute*), 40
region (*cassiopeia.Champion attribute*), 19
region (*cassiopeia.ChampionMasteries attribute*), 26
region (*cassiopeia.ChampionMastery attribute*), 27
region (*cassiopeia.Champions attribute*), 18
region (*cassiopeia.core.ChallengerLeague attribute*), 31
region (*cassiopeia.core.league.League attribute*), 31
region (*cassiopeia.core.league.LeagueEntries attribute*), 34
region (*cassiopeia.core.league.LeagueEntry attribute*), 35
region (*cassiopeia.core.league.LeagueSummonerEntries attribute*), 32
region (*cassiopeia.core.MasterLeague attribute*), 31
region (*cassiopeia.core.match.MatchHistory attribute*), 39
region (*cassiopeia.core.match.Timeline attribute*), 44
region (*cassiopeia.core.spectator.CurrentMatch attribute*), 53

region (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
region (*cassiopeia.data.Platform attribute*), 11
region (*cassiopeia.FeaturedMatches attribute*), 52
region (*cassiopeia.Item attribute*), 29
region (*cassiopeia.Items attribute*), 28
region (*cassiopeia.LanguageStrings attribute*), 30
region (*cassiopeia.Locales attribute*), 36
region (*cassiopeia.Map attribute*), 37
region (*cassiopeia.Maps attribute*), 37
region (*cassiopeia.Match attribute*), 40
region (*cassiopeia.Patch attribute*), 47
region (*cassiopeia.ProfileIcons attribute*), 47
region (*cassiopeia.Realms attribute*), 49
region (*cassiopeia.Rune attribute*), 51
region (*cassiopeia.Runes attribute*), 50
region (*cassiopeia.ShardStatus attribute*), 51
region (*cassiopeia.Summoner attribute*), 54
region (*cassiopeia.SummonerSpell attribute*), 57
region (*cassiopeia.SummonerSpells attribute*), 55
region (*cassiopeia.Versions attribute*), 58
region (*cassiopeia_championgg.core.ChampionGGMatchup attribute*), 24
Region (*class in cassiopeia.data*), 14
release_date (*cassiopeia.Champion attribute*), 19
remove() (*cassiopeia.ChampionMasteries method*), 26
remove() (*cassiopeia.Champions method*), 18
remove() (*cassiopeia.core.league.LeagueEntries method*), 34
remove() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
remove() (*cassiopeia.core.match.MatchHistory method*), 39
remove() (*cassiopeia.FeaturedMatches method*), 52
remove() (*cassiopeia.Items method*), 28
remove() (*cassiopeia.Locales method*), 36
remove() (*cassiopeia.Maps method*), 37
remove() (*cassiopeia.ProfileIcons method*), 48
remove() (*cassiopeia.Runes method*), 50
remove() (*cassiopeia.SummonerSpells method*), 56
remove() (*cassiopeia.Versions method*), 58
resolve (*cassiopeia.data.MasteryTree attribute*), 10
resolve (*cassiopeia.Runes attribute*), 50
resource (*cassiopeia.Champion attribute*), 19
resource (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 23
resource (*cassiopeia.SummonerSpell attribute*), 57
Resource (*class in cassiopeia.data*), 14
reverse() (*cassiopeia.ChampionMasteries method*), 26
reverse() (*cassiopeia.Champions method*), 18
reverse() (*cassiopeia.core.league.LeagueEntries method*), 34
reverse() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
reverse() (*cassiopeia.core.match.MatchHistory method*), 39
reverse() (*cassiopeia.FeaturedMatches method*), 52
reverse() (*cassiopeia.Items method*), 28
reverse() (*cassiopeia.Locales method*), 36
reverse() (*cassiopeia.Maps method*), 37
reverse() (*cassiopeia.ProfileIcons method*), 48
reverse() (*cassiopeia.Runes method*), 50
reverse() (*cassiopeia.SummonerSpells method*), 56
reverse() (*cassiopeia.Versions method*), 58
revision (*cassiopeia.Patch attribute*), 47
revision_date (*cassiopeia.Summoner attribute*), 55
rift_herald_kills (*cassiopeia.core.match.Team attribute*), 40
river_bot (*cassiopeia.data.SummonersRiftArea attribute*), 16
river_top (*cassiopeia.data.SummonersRiftArea attribute*), 16
role (*cassiopeia.core.league.LeagueEntry attribute*), 35
role (*cassiopeia.core.match.Participant attribute*), 41
role (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
role (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
Role (*class in cassiopeia.data*), 15
Rune (*class in cassiopeia*), 51
runes (*cassiopeia.core.match.Participant attribute*), 41
runes (*cassiopeia.core.spectator.Participant attribute*), 54
Runes (*class in cassiopeia*), 49
russia (*cassiopeia.data.Platform attribute*), 11
russia (*cassiopeia.data.Region attribute*), 14

S

sanitized_description (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 23
sanitized_description (*cassiopeia.core.staticdata.champion.Passive attribute*), 21
sanitized_description (*cassiopeia.Item attribute*), 29
sanitized_description (*cassiopeia.SummonerSpell attribute*), 57
sanitized_name (*cassiopeia.Summoner attribute*), 55
sanitized_tooltip (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 23
sanitized_tooltip (*cassiopeia.SummonerSpell attribute*), 57
search() (*cassiopeia.ChampionMasteries method*), 26
search() (*cassiopeia.Champions method*), 18
search() (*cassiopeia.core.league.LeagueEntries method*), 34

search() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
search() (*cassiopeia.core.match.MatchHistory method*), 39
search() (*cassiopeia.FeaturedMatches method*), 52
search() (*cassiopeia.Items method*), 28
search() (*cassiopeia.Locales method*), 36
search() (*cassiopeia.Maps method*), 37
search() (*cassiopeia.ProfileIcons method*), 48
search() (*cassiopeia.Runes method*), 50
search() (*cassiopeia.SummonerSpells method*), 56
search() (*cassiopeia.Versions method*), 58
season (*cassiopeia.Patch attribute*), 47
Season (*class in cassiopeia.data*), 15
season_3 (*cassiopeia.data.Season attribute*), 15
season_4 (*cassiopeia.data.Season attribute*), 15
season_5 (*cassiopeia.data.Season attribute*), 15
season_6 (*cassiopeia.data.Season attribute*), 15
season_7 (*cassiopeia.data.Season attribute*), 15
season_8 (*cassiopeia.data.Season attribute*), 15
season_9 (*cassiopeia.data.Season attribute*), 15
services (*cassiopeia.ShardStatus attribute*), 51
set_riot_api_key() (*cassiopeia._configuration.settings.Settings method*), 8
Settings (*class in cassiopeia._configuration.settings*), 8
ShardStatus (*class in cassiopeia*), 51
shield (*cassiopeia.data.Resource attribute*), 15
short_description (*cassiopeia.Rune attribute*), 51
showdown (*cassiopeia.data.GameMode attribute*), 9
showdown_1v1 (*cassiopeia.data.Queue attribute*), 14
showdown_2v2 (*cassiopeia.data.Queue attribute*), 14
side (*cassiopeia.core.match.Event attribute*), 45
side (*cassiopeia.core.match.Participant attribute*), 41
side (*cassiopeia.core.match.Team attribute*), 40
side (*cassiopeia.core.spectator.Participant attribute*), 54
side (*cassiopeia.core.spectator.Team attribute*), 53
Side (*class in cassiopeia.data*), 16
sight_wards_bought (*cassiopeia.core.match.ParticipantStats attribute*), 43
silver (*cassiopeia.data.Tier attribute*), 17
skill (*cassiopeia.core.match.Event attribute*), 45
skill_order (*cassiopeia.core.match.Participant attribute*), 41
skills (*cassiopeia.core.match.ParticipantState attribute*), 46
Skin (*class in cassiopeia.core.staticdata.champion*), 21
skins (*cassiopeia.Champion attribute*), 19
slug (*cassiopeia.ShardStatus attribute*), 51
solo (*cassiopeia.data.Role attribute*), 15
sorcery (*cassiopeia.Runes attribute*), 50
sort() (*cassiopeia.ChampionMasteries method*), 26
sort() (*cassiopeia.Champions method*), 18
sort() (*cassiopeia.core.league.LeagueEntries method*), 34
sort() (*cassiopeia.core.league.LeagueSummonerEntries method*), 32
sort() (*cassiopeia.core.match.MatchHistory method*), 39
sort() (*cassiopeia.FeaturedMatches method*), 53
sort() (*cassiopeia.Items method*), 28
sort() (*cassiopeia.Locales method*), 36
sort() (*cassiopeia.Maps method*), 37
sort() (*cassiopeia.ProfileIcons method*), 48
sort() (*cassiopeia.Runes method*), 50
sort() (*cassiopeia.SummonerSpells method*), 56
sort() (*cassiopeia.Versions method*), 58
special_recipe (*cassiopeia.Item attribute*), 29
spell_1_casts (*cassiopeia.core.match.ParticipantStats attribute*), 43
spell_2_casts (*cassiopeia.core.match.ParticipantStats attribute*), 43
spell_3_casts (*cassiopeia.core.match.ParticipantStats attribute*), 43
spell_4_casts (*cassiopeia.core.match.ParticipantStats attribute*), 43
spell_vamp (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30
spells (*cassiopeia.Champion attribute*), 19
SpellVars (*class in cassiopeia.core.staticdata.champion*), 23
SpellVars (*class in cassiopeia.core.staticdata.summonerspell*), 57
splash (*cassiopeia.core.staticdata.champion.Skin attribute*), 21
splash_url (*cassiopeia.core.staticdata.champion.Skin attribute*), 21
sprite (*cassiopeia.Champion attribute*), 19
sprite (*cassiopeia.Item attribute*), 29
sprite (*cassiopeia.Map attribute*), 38
sprite (*cassiopeia.SummonerSpell attribute*), 57
star_guardian (*cassiopeia.data.GameMode attribute*), 9
start (*cassiopeia.Match attribute*), 40
start (*cassiopeia.Patch attribute*), 47
start() (*cassiopeia.data.Season method*), 16
stat_runes (*cassiopeia.core.match.Participant attribute*), 41
stats (*cassiopeia.Champion attribute*), 19
stats (*cassiopeia.core.match.Participant attribute*), 41
stats (*cassiopeia.Item attribute*), 29
Stats (*class in cassiopeia.core.staticdata.champion*), 20
store (*cassiopeia.Realms attribute*), 49
strings (*cassiopeia.LanguageStrings attribute*), 30
summoner (*cassiopeia.ChampionMasteries attribute*), 26
summoner (*cassiopeia.ChampionMastery attribute*), 27

summoner (*cassiopeia.core.league.LeagueEntry attribute*), 35
summoner (*cassiopeia.core.match.Participant attribute*), 41
summoner (*cassiopeia.core.spectator.Participant attribute*), 54
Summoner (*class in cassiopeia*), 54
summoner_spell_1_casts (*cassiopeia.core.match.ParticipantStats attribute*), 43
summoner_spell_2_casts (*cassiopeia.core.match.ParticipantStats attribute*), 43
summoner_spell_d (*cassiopeia.core.match.Participant attribute*), 41
summoner_spell_d (*cassiopeia.core.spectator.Participant attribute*), 54
summoner_spell_f (*cassiopeia.core.match.Participant attribute*), 41
summoner_spell_f (*cassiopeia.core.spectator.Participant attribute*), 54
SummonerSpell (*class in cassiopeia*), 56
SummonerSpells (*class in cassiopeia*), 55
SummonersRiftArea (*class in cassiopeia.data*), 16

T

tags (*cassiopeia.Champion attribute*), 20
tags (*cassiopeia.Item attribute*), 29
team (*cassiopeia.core.match.Participant attribute*), 41
team (*cassiopeia.core.spectator.Participant attribute*), 54
Team (*class in cassiopeia.core.match*), 40
Team (*class in cassiopeia.core.spectator*), 53
team_position (*cassiopeia.core.match.Participant attribute*), 41
team_score (*cassiopeia.core.match.ParticipantFrame attribute*), 45
team_score (*cassiopeia.core.match.ParticipantState attribute*), 46
teams (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
teams (*cassiopeia.Match attribute*), 40
thirty_to_end (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
three (*cassiopeia.data.Division attribute*), 9
threes (*cassiopeia.core.league.LeagueSummonerEntries attribute*), 33
tier (*cassiopeia.core.ChallengerLeague attribute*), 31
tier (*cassiopeia.core.league.League attribute*), 31
tier (*cassiopeia.core.league.LeagueEntries attribute*), 34
tier (*cassiopeia.core.league.LeagueEntry attribute*), 35
tier (*cassiopeia.core.MasterLeague attribute*), 31
tier (*cassiopeia.Item attribute*), 29
tier (*cassiopeia.Rune attribute*), 51
Tier (*class in cassiopeia.data*), 16
time_CCing_others (*cassiopeia.core.match.ParticipantStats attribute*), 43
time_played (*cassiopeia.core.match.ParticipantStats attribute*), 43
timeline (*cassiopeia.core.match.Participant attribute*), 41
timeline (*cassiopeia.Match attribute*), 40
Timeline (*class in cassiopeia.core.match*), 44
timestamp (*cassiopeia.core.match.Event attribute*), 45
timestamp (*cassiopeia.core.match.Frame attribute*), 44
timezone (*cassiopeia.data.Region attribute*), 14
title (*cassiopeia.Champion attribute*), 20
title (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 22
to_dict() (*cassiopeia.ChampionMasteries method*), 26
to_dict() (*cassiopeia.Champions method*), 18
to_dict() (*cassiopeia.core.league.LeagueEntries method*), 34
to_dict() (*cassiopeia.core.league.LeagueSummonerEntries method*), 33
to_dict() (*cassiopeia.core.match.MatchHistory method*), 39
to_dict() (*cassiopeia.FeaturedMatches method*), 53
to_dict() (*cassiopeia.Items method*), 28
to_dict() (*cassiopeia.Locales method*), 36
to_dict() (*cassiopeia.Maps method*), 37
to_dict() (*cassiopeia.ProfileIcons method*), 48
to_dict() (*cassiopeia.Runes method*), 50
to_dict() (*cassiopeia.SummonerSpells method*), 56
to_dict() (*cassiopeia.Versions method*), 59
to_json() (*cassiopeia.ChampionMasteries method*), 26
to_json() (*cassiopeia.Champions method*), 18
to_json() (*cassiopeia.core.league.LeagueEntries method*), 34
to_json() (*cassiopeia.core.league.LeagueSummonerEntries method*), 33
to_json() (*cassiopeia.core.match.MatchHistory method*), 39
to_json() (*cassiopeia.FeaturedMatches method*), 53
to_json() (*cassiopeia.Items method*), 28
to_json() (*cassiopeia.Locales method*), 36
to_json() (*cassiopeia.Maps method*), 37
to_json() (*cassiopeia.ProfileIcons method*), 48
to_json() (*cassiopeia.Runes method*), 50
to_json() (*cassiopeia.SummonerSpells method*), 56
to_json() (*cassiopeia.Versions method*), 59
tokens (*cassiopeia.ChampionMastery attribute*), 27
tooltip (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 23
tooltip (*cassiopeia.SummonerSpell attribute*), 57

top (*cassiopeia.data.Position attribute*), 11
top_lane (*cassiopeia.data.Lane attribute*), 10
top_lane_blue (*cassiopeia.data.SummonersRiftArea attribute*), 16
top_lane_purple (*cassiopeia.data.SummonersRiftArea attribute*), 16
top_lane_red (*cassiopeia.data.SummonersRiftArea attribute*), 16
total_damage_dealt (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_damage_dealt_to_champions (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_damage_dealt_to_champions (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
total_damage_shielded_on_teammates (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_damage_taken (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_damage_taken (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
total_heal (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_healed (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
total_heals_on_teammates (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_minions_killed (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_time_cc_dealt (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_time_spent_dead (*cassiopeia.core.match.ParticipantStats attribute*), 43
total_units_healed (*cassiopeia.core.match.ParticipantStats attribute*), 43
tourney (*cassiopeia.data.MatchType attribute*), 11
Tower (*class in cassiopeia.data*), 17
tower_kills (*cassiopeia.core.match.Team attribute*), 41
tower_type (*cassiopeia.core.match.Event attribute*), 45
triple_kills (*cassiopeia.core.match.ParticipantStats attribute*), 43
true_damage_dealt (*cassiopeia.core.match.ParticipantStats attribute*), 43
true_damage_dealt_to_champions (*cassiopeia.core.match.ParticipantStats attribute*), 43
true_damage_taken (*cassiopeia.core.match.ParticipantStats attribute*), 43
turkey (*cassiopeia.data.Platform attribute*), 11
turkey (*cassiopeia.data.Region attribute*), 14
turret_kills (*cassiopeia.core.match.ParticipantStats attribute*), 43
turret_takedowns (*cassiopeia.core.match.ParticipantStats attribute*), 43
turrets_lost (*cassiopeia.core.match.ParticipantStats attribute*), 43
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*), 14
tutorial3 (*cassiopeia.data.Queue attribute*), 14
tutorial_1 (*cassiopeia.data.GameMode attribute*), 10
tutorial_2 (*cassiopeia.data.GameMode attribute*), 10
tutorial_3 (*cassiopeia.data.GameMode attribute*), 10
twenty_to_thirty (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
two (*cassiopeia.data.Division attribute*), 9
type (*cassiopeia.core.match.Event attribute*), 45
type (*cassiopeia.core.spectator.CurrentMatch attribute*), 53
type (*cassiopeia.core.staticdata.champion.ItemSet attribute*), 22
type (*cassiopeia.core.staticdata.champion.RecommendedItems attribute*), 22
type (*cassiopeia.LanguageStrings attribute*), 30
type (*cassiopeia.Match attribute*), 40

U

ultimate_spellbook (*cassiopeia.data.Queue attribute*), 14
UNDEFINED (*cassiopeia.data.Tower attribute*), 17
unpurchasable_items (*cassiopeia.Map attribute*), 38
unranked (*cassiopeia.data.Tier attribute*), 17
unreal_kills (*cassiopeia.core.match.ParticipantStats attribute*), 43
urf (*cassiopeia.data.GameMode attribute*), 10
urf (*cassiopeia.data.Queue attribute*), 14
urf_coop_ai (*cassiopeia.data.Queue attribute*), 14
url (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
utility (*cassiopeia.data.Lane attribute*), 10
utility (*cassiopeia.data.Position attribute*), 11
utlbook (*cassiopeia.data.GameMode attribute*), 10

V

variables (*cassiopeia.core.staticdata.champion.ChampionSpell attribute*), 23
variables (*cassiopeia.SummonerSpell attribute*), 57
verification_string (*cassiopeia.Summoner attribute*), 55
version (*cassiopeia.Champion attribute*), 20
version (*cassiopeia.Champions attribute*), 18
version (*cassiopeia.core.match.Participant attribute*), 41
version (*cassiopeia.core.staticdata.profileicon.ProfileIcon attribute*), 48
version (*cassiopeia.Item attribute*), 29
version (*cassiopeia.Items attribute*), 28
version (*cassiopeia.LanguageStrings attribute*), 30
version (*cassiopeia.Map attribute*), 38
version (*cassiopeia.Maps attribute*), 37
version (*cassiopeia.Match attribute*), 40
version (*cassiopeia.ProfileIcons attribute*), 48
version (*cassiopeia.Realms attribute*), 49
version (*cassiopeia.Rune attribute*), 51
version (*cassiopeia.Runes attribute*), 51
version (*cassiopeia.SummonerSpell attribute*), 57
version (*cassiopeia.SummonerSpells attribute*), 56
version_from_match (*cassiopeia._configuration.settings.Settings attribute*), 8
Versions (*class in cassiopeia*), 58
veteran (*cassiopeia.core.league.LeagueEntry attribute*), 35
victim_id (*cassiopeia.core.match.Event attribute*), 45
vision_score (*cassiopeia.core.match.ParticipantStats attribute*), 43
vision_wards_bought (*cassiopeia.core.match.ParticipantStats attribute*), 43
vision_wards_placed (*cassiopeia.core.match.ParticipantStats attribute*), 44

W

W (*cassiopeia.data.Key attribute*), 10
ward_type (*cassiopeia.core.match.Event attribute*), 45
wards_killed (*cassiopeia.core.match.ParticipantStats attribute*), 44
wards_killed (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
wards_placed (*cassiopeia.core.match.ParticipantStats attribute*), 44
weighted_score (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
win (*cassiopeia.core.match.ParticipantStats attribute*), 44
win (*cassiopeia.core.match.Team attribute*), 41
win_rate (*cassiopeia_championgg.core.ChampionGGStats attribute*), 24
win_rates (*cassiopeia.Champion attribute*), 20
winrate (*cassiopeia_championgg.core.ChampionGGMatchup attribute*), 24
winrate (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
wins (*cassiopeia.core.league.LeagueEntry attribute*), 35
wins (*cassiopeia.core.league.MiniSeries attribute*), 33
wins (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25
wins_required (*cassiopeia.core.league.MiniSeries attribute*), 33

X

x (*cassiopeia.core.match.Position attribute*), 45
xp_bonus (*cassiopeia.core.staticdata.item.ItemStats attribute*), 30

Y

y (*cassiopeia.core.match.Position attribute*), 45

Z

zero_to_ten (*cassiopeia_championgg.core.ChampionGGMatchupStats attribute*), 25