
Cassiopeia Documentation

Release 3.0.x

Rob Rua

Sep 03, 2017

Contents

1	What is Cassiopeia?	1
2	Why use Cass?	3
3	An Example	5
4	Contributing	7
5	Overview	9
6	Top Level APIs	63
7	Index and Search	65
	Python Module Index	67

CHAPTER 1

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 2

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 3

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", id=21359666)
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 behind the scenes, you just get to write good code.)

```
kalturi = Summoner(name="Kalturi", id=21359666)
```

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

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

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

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

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

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

True for any champion mastery whose mastery level is greater than or equal to 6.

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

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

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

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

CHAPTER 4

Contributing

Contributions are welcome and we have an entire [page](#) devoted to ways in which you can help us with Cass.

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")

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

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")
kalturi.champion_masteries
kalturi.match_history
kalturi.mastery_pages
kalturi.rune_pages
kalturi.current_match
kalturi.leagues
```

Methods and Class Constructors

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

Champions

```
cassiopeia.get_champions(region: typing.Union[cassiopeia.data.Region, str] = None) → typing.List[cassiopeia.core.staticdata.champion.Champion]
cassiopeia.get_champion(key: typing.Union[str, int], region: typing.Union[cassiopeia.data.Region, str] = None) → cassiopeia.core.staticdata.champion.Champion
class cassiopeia.Champions(*args, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None)
    Bases: cassiopeia.core.common.CassiopeiaGhostList
    append(object) → None – append object to end
    clear() → None – remove all items from L
    contains(item: typing.Any) → bool
    copy() → list – a shallow copy of L
    count(value) → integer – return number of occurrences of value
    delete(item: typing.Any) → None
    enumerate(item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]
    extend(iterable) → None – extend list by appending elements from the iterable
    filter(function)
    find(item: typing.Any, reverse: bool = False) → typing.Any
    from_data(data: typing.Union[list, cassiopeia.core.common.CoreData])
    included_data
        A set of tags to return additional information for this champion when it's loaded.
    index(value[, start[, stop]]) → integer – return first index of value.
        Raises ValueError if the value is not present.
    insert()
        L.insert(index, object) – insert object before index
    locale
        The locale for this champion.
    platform
        The platform for this champion.
    pop([index]) → item – remove and return item at index (default last).
        Raises IndexError if list is empty or index is out of range.
    property(load_group_or_method: typing.Any) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]
    region
        The region for this champion.
    remove(value) → None – remove first occurrence of value.
        Raises ValueError if the value is not present.
    reverse()
        L.reverse() – reverse IN PLACE
    search(item: typing.Any, streaming: bool = False, reverse: bool = False) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]
```

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

version

The version for this champion.

```
class cassiopeia.Champion(*, id: int = None, name: str = None, key: str = None, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None)
```

Bases: `cassiopeia.core.common.CassiopeiaGhost`

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

ally_tips

The tips for playing with this champion.

blurb

A short blurb about this champion.

coop_ai_enabled

Whether or not the champion is currently enabled in coop and AI games.

custom_enabled

Whether or not the champion is currently enabled in custom games.

enabled

Whether or not the champion is currently enabled.

enemy_tips

The tips for playing against this champion.

free_to_play

Whether or not the champion is currently free to play.

id

The champion's ID.

image

The image information for this champion.

included_data

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

info

Info about this champion.

key

The champion's key.

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.

ranked_enabled

Whether or not the champion is currently enabled in ranked games.

recommended_itemsets

The champion's recommended itemsets.

region

The region for this champion.

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.

```
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 ['Item', 'GameMode', 'str']
    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 ['Item', 'str']
    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']
    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.

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.

Champion Masteries

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

```
cassiopeia.get_champion_masteries (summoner: typing.Union[cassiopeia.core.summoner.Summoner,
                                                         int, str], region: typing.Union[cassiopeia.data.Region, str] = None) →
cassiopeia.core.championmastery.ChampionMasteries

class cassiopeia.ChampionMasteries (*args, summoner: typing.Union[cassiopeia.core.summoner.Summoner,
                                                                    int, str], region: typing.Union[cassiopeia.data.Region, str] =
                                                                    None)
    Bases: cassiopeia.core.common.CassiopeiaGhostList
    append (object) → None – append object to end
    clear () → None – remove all items from L
    contains (item: typing.Any) → bool
    copy () → list – a shallow copy of L
    count (value) → integer – return number of occurrences of value
    delete (item: typing.Any) → None
    enumerate (item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typ-
ing.Any], NoneType], NoneType]
    extend (iterable) → None – extend list by appending elements from the iterable
    filter (function)
    find (item: typing.Any, reverse: bool = False) → typing.Any
    from_data (data: typing.Union[list, cassiopeia.core.common.CoreData])
    index (value[, start[, stop]]) → integer – return first index of value.
        Raises ValueError if the value is not present.
    insert ()
        L.insert(index, object) – insert object before index
    platform
    pop ([index]) → item – remove and return item at index (default last).
        Raises IndexError if list is empty or index is out of range.
    property (load_group_or_method: typing.Any) → typing.Union[property, typ-
ing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]
    region
    remove (value) → None – remove first occurrence of value.
        Raises ValueError if the value is not present.
    reverse ()
        L.reverse() – reverse IN PLACE
    search (item: typing.Any, streaming: bool = False, reverse: bool = False) → typ-
ing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]
    sort (key=None, reverse=False) → None – stable sort *IN PLACE*
    summoner
```

```

class cassiopeia.ChampionMastery (*, summoner: typing.Union[cassiopeia.core.summoner.Summoner,
                                                             int, str] = None, champion: typing.
                                                             Union[cassiopeia.core.staticdata.champion.Champion,
                                                             int, str] = None, region: typing.Union[cassiopeia.data.Region,
                                                             str] = None)
Bases: cassiopeia.core.common.CassiopeiaGhost
Searchable by ['str', 'bool', 'Summoner', 'Champion', 'datetime', 'int']

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.

```

Items

```

cassiopeia.get_items (region: typing.Union[cassiopeia.data.Region, str] = None) → typing.
List[cassiopeia.core.staticdata.item.Item]

class cassiopeia.Items (*args, region: typing.Union[cassiopeia.data.Region, str] = None, version: str =
                        None, locale: str = None, included_data: typing.Set[str] = None)
Bases: cassiopeia.core.common.CassiopeiaGhostList

append (object) → None – append object to end

clear () → None – remove all items from L

contains (item: typing.Any) → bool

copy () → list – a shallow copy of L

count (value) → integer – return number of occurrences of value

delete (item: typing.Any) → None

enumerate (item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typ-
ing.Any], NoneType], NoneType]

```

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

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

index (*value[, start[, stop]]*) → integer – return first index of value.
Raises ValueError if the value is not present.

insert ()
L.insert(index, object) – insert object before index

locale
The locale for this item.

platform

pop ([*index*]) → item – remove and return item at index (default last).
Raises IndexError if list is empty or index is out of range.

property (*load_group_or_method: typing.Any*) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (*value*) → None – remove first occurrence of value.
Raises ValueError if the value is not present.

reverse ()
L.reverse() – reverse *IN PLACE*

search (*item: typing.Any, streaming: bool = False, reverse: bool = False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

version

class cassiopeia.Item(*, *id: int = None, name: str = None, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None*)
Bases: cassiopeia.core.common.CassiopeiaGhost
Searchable by ['str', 'Region', 'Map', 'Platform', 'int']

builds_from

builds_into

champion

consume_on_full

consumed

description

effect

gold

group

hide

id
The item's ID.

image
The image information for this item.

in_store

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

keywords

locale
The locale for this item.

maps

max_stacks

name

plaintext

platform
The platform for this item.

region
The region for this item.

sanitized_description

special_recipe

sprite

stats

tags

tier

version
The version for this item.

```
class cassiopeia.core.staticdata.item.ItemStats(**kwargs)
    Bases: 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` (*region: typing.Union[cassiopeia.data.Region, str] = None*) → `cassiopeia.core.staticdata.languagestrings.LanguageStrings`

`class cassiopeia.LanguageStrings` (*, *strings: typing.Dict[str, str] = None, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None*)

Bases: `cassiopeia.core.common.CassiopeiaGhost`

Searchable by []

locale

platform

region

strings

type

version

Leagues

```
cassiopeia.Summoner.leagues

cassiopeia.get_leagues (summoner: typing.Union[cassiopeia.core.summoner.Summoner, int, str],
                        region: typing.Union[cassiopeia.data.Region, str] = None) → cassiopeia.core.league.Leagues

cassiopeia.get_challenger_league (queue: typing.Union[cassiopeia.data.Queue, int, str], region:
                                  typing.Union[cassiopeia.data.Region, str] = None) → cassiopeia.core.league.ChallengerLeague

cassiopeia.get_master_league (queue: typing.Union[cassiopeia.data.Queue, int, str], region:
                              typing.Union[cassiopeia.data.Region, str] = None) → cassiopeia.core.league.MasterLeague

class cassiopeia.core.league.Leagues (*args, summoner: typing.Union[cassiopeia.core.summoner.Summoner, int, str],
                                       region: typing.Union[cassiopeia.data.Region, str] = None)
    Bases: cassiopeia.core.common.CassiopeiaGhostList
    append (object) → None – append object to end
    clear () → None – remove all items from L
    contains (item: typing.Any) → bool
    copy () → list – a shallow copy of L
    count (value) → integer – return number of occurrences of value
    delete (item: typing.Any) → None
    enumerate (item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]
    extend (iterable) → None – extend list by appending elements from the iterable
    filter (function)
    find (item: typing.Any, reverse: bool = False) → typing.Any
    fives
    flex
    from_data (data: typing.Union[list, cassiopeia.core.common.CoreData])
    index (value[, start[, stop]]) → integer – return first index of value.
        Raises ValueError if the value is not present.
    insert ()
        L.insert(index, object) – insert object before index
    platform
    pop ([index]) → item – remove and return item at index (default last).
        Raises IndexError if list is empty or index is out of range.
    property (load_group_or_method: typing.Any) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]
    region
    remove (value) → None – remove first occurrence of value.
        Raises ValueError if the value is not present.
```

reverse ()

L.reverse() – reverse *IN PLACE*

search (item: typing.Any, streaming: bool = False, reverse: bool = False) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (key=None, reverse=False) → None – stable sort **IN PLACE**

threes

class cassiopeia.core.league.**League** (**kwargs)

Bases: cassiopeia.core.common.CassiopeiaObject

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

entries

name

platform

queue

region

tier

class cassiopeia.**ChallengerLeague** (*, queue: typing.Union[cassiopeia.data.Queue, str, int] = None, region: typing.Union[cassiopeia.data.Region, str] = None)

Bases: [cassiopeia.core.league.League](#), cassiopeia.core.common.CassiopeiaGhost

entries

name

queue

region

tier

class cassiopeia.**MasterLeague** (*, queue: typing.Union[cassiopeia.data.Queue, str, int] = None, region: typing.Union[cassiopeia.data.Region, str] = None)

Bases: [cassiopeia.core.league.League](#), cassiopeia.core.common.CassiopeiaGhost

entries

name

queue

region

tier

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, summoner: typing.
    Union[cassiopeia.core.summoner.Summoner,
    int, str], region: typing.
    Union[cassiopeia.data.Region, str] = None)
    Bases: cassiopeia.core.common.CassiopeiaGhostList
```

append (*object*) → None – append object to end

clear () → None – remove all items from L

contains (*item: typing.Any*) → bool

copy () → list – a shallow copy of L

count (*value*) → integer – return number of occurrences of value

delete (*item: typing.Any*) → None

enumerate (*item: typing.Any, reverse: bool = False*) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

fives

flex

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

index (*value[, start[, stop]]*) → integer – return first index of value.
 Raises ValueError if the value is not present.

insert ()
 L.insert(index, object) – insert object before index

platform

pop ([*index*]) → item – remove and return item at index (default last).
 Raises IndexError if list is empty or index is out of range.

property (*load_group_or_method: typing.Any*) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (*value*) → None – remove first occurrence of value.
 Raises ValueError if the value is not present.

reverse ()
 L.reverse() – reverse *IN PLACE*

search (*item: typing.Any, streaming: bool = False, reverse: bool = False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

threes

```
class cassiopeia.core.league.LeagueEntry (*, region: typing.Union[cassiopeia.data.Region, str]
    = None)
    Bases: cassiopeia.core.common.CassiopeiaGhost
```

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

division

fresh_blood

hot_streak

league_points

losses

name

platform

The platform for this champion.

promos

queue

region

The region for this champion.

summoner

tier

veteran

wins

Locales

`cassiopeia.get_locales` (*region*: `typing.Union[cassiopeia.data.Region, str] = None`) → `typing.List[str]`

class `cassiopeia.Locales` (**args, region*: `typing.Union[cassiopeia.data.Region, str] = None`)
Bases: `cassiopeia.core.common.CassiopeiaGhostList`

append (*object*) → None – append object to end

clear () → None – remove all items from L

contains (*item*: `typing.Any`) → bool

copy () → list – a shallow copy of L

count (*value*) → integer – return number of occurrences of value

delete (*item*: `typing.Any`) → None

enumerate (*item*: `typing.Any`, *reverse*: `bool = False`) → `typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]`

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data*: `typing.Union[list, cassiopeia.core.common.CoreData]`)

index (*value* [, *start* [, *stop*]]) → integer – return first index of value.
Raises `ValueError` if the value is not present.

insert ()
L.insert(index, object) – insert object before index

platform

pop (*[index]*) → item – remove and return item at index (default last).
 Raises IndexError if list is empty or index is out of range.

property (*load_group_or_method: typing.Any*) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (*value*) → None – remove first occurrence of value.
 Raises ValueError if the value is not present.

reverse ()
 L.reverse() – reverse *IN PLACE*

search (*item: typing.Any, streaming: bool = False, reverse: bool = False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

Maps

cassiopeia.get_maps (*region: typing.Union[cassiopeia.data.Region, str] = None*) → typing.List[cassiopeia.core.staticdata.map.Map]

class cassiopeia.Maps (**args, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None*)
 Bases: cassiopeia.core.common.CassiopeiaGhostList

append (*object*) → None – append object to end

clear () → None – remove all items from L

contains (*item: typing.Any*) → bool

copy () → list – a shallow copy of L

count (*value*) → integer – return number of occurrences of value

delete (*item: typing.Any*) → None

enumerate (*item: typing.Any, reverse: bool = False*) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

index (*value[, start[, stop]]*) → integer – return first index of value.
 Raises ValueError if the value is not present.

insert ()
 L.insert(index, object) – insert object before index

locale**platform**

pop (*[index]*) → item – remove and return item at index (default last).
 Raises IndexError if list is empty or index is out of range.

```
property (load_group_or_method: typing.Any) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (value) → None – remove first occurrence of value.
    Raises ValueError if the value is not present.

reverse ()
    L.reverse() – reverse IN PLACE

search (item: typing.Any, streaming: bool = False, reverse: bool = False) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (key=None, reverse=False) → None – stable sort *IN PLACE*

version

class cassiopeia.Map (*, id: int = None, name: str = None, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None)
    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.
```

Masteries

```
cassiopeia.Summoner.mastery_pages

cassiopeia.get_masteries (region: typing.Union[cassiopeia.data.Region, str] = None) → typing.List[cassiopeia.core.staticdata.mastery.Mastery]

class cassiopeia.Masteries (*args, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None)
    Bases: cassiopeia.core.common.CassiopeiaGhostList

    append (object) → None – append object to end

    clear () → None – remove all items from L

    contains (item: typing.Any) → bool

    copy () → list – a shallow copy of L

    count (value) → integer – return number of occurrences of value

    delete (item: typing.Any) → None
```

enumerate (*item: typing.Any, reverse: bool = False*) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

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

index (*value[, start[, stop]]*) → integer – return first index of value.
Raises ValueError if the value is not present.

insert ()
L.insert(index, object) – insert object before index

locale

platform

pop (*[index]*) → item – remove and return item at index (default last).
Raises IndexError if list is empty or index is out of range.

property (*load_group_or_method: typing.Any*) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (*value*) → None – remove first occurrence of value.
Raises ValueError if the value is not present.

reverse ()
L.reverse() – reverse *IN PLACE*

search (*item: typing.Any, streaming: bool = False, reverse: bool = False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

version

class cassiopeia.**Mastery** (*, *id: int = None, name: str = None, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None*)
Bases: cassiopeia.core.common.CassiopeiaGhost
Searchable by ['str', 'MasteryTree', 'Region', 'int', 'Platform']

description
The mastery's description.

id
The mastery's ID.

image
The image information for this mastery.

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

locale
The locale for this mastery.

name
The mastery's name.

platform
The platform for this mastery.

points
The number of points that can be put into this mastery.

prerequisite
The prerequisite masteries.

region
The region for this mastery.

sanitized_description
The mastery's sanitized description.

sprite

tree
The mastery's tree (Cunning, Ferocity, Resolve).

version
The version for this mastery.

```
class cassiopeia.core.masterypage.MasteryPages (*args, summoner: typing.Union[cassiopeia.core.summoner.Summoner,
int, str], region: typing.Union[cassiopeia.data.Region, str]
= None)
Bases: cassiopeia.core.common.CassiopeiaGhostList
```

append (*object*) → None – append object to end

clear () → None – remove all items from L

contains (*item: typing.Any*) → bool

copy () → list – a shallow copy of L

count (*value*) → integer – return number of occurrences of value

delete (*item: typing.Any*) → None

enumerate (*item: typing.Any, reverse: bool = False*) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

index (*value* [, *start* [, *stop*]]) → integer – return first index of value.
Raises ValueError if the value is not present.

insert ()
L.insert(index, object) – insert object before index

platform

pop ([*index*]) → item – remove and return item at index (default last).
Raises IndexError if list is empty or index is out of range.

```

property (load_group_or_method: typing.Any) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (value) → None – remove first occurrence of value.
    Raises ValueError if the value is not present.

reverse ()
    L.reverse() – reverse IN PLACE

search (item: typing.Any, streaming: bool = False, reverse: bool = False) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (key=None, reverse=False) → None – stable sort *IN PLACE*

class cassiopeia.core.masterypage.MasteryPage (**kwargs)
    Bases: cassiopeia.core.common.CassiopeiaGhost
    Searchable by ['str', 'bool', 'Region', 'Platform', 'int', 'Mastery']

    current

    id

    locale
        The locale for this mastery page.

    masteries

    name

    platform
        The platform for this mastery page.

    region

    summoner

```

Matches

```

cassiopeia.Summoner.match_history

cassiopeia.get_match_history (summoner: typing.Union[cassiopeia.core.summoner.Summoner, str, int] = None, region: typing.Union[cassiopeia.data.Region, str] = None, begin_index: int = 0, end_index: int = None, begin_time: datetime.datetime = None, end_time: datetime.datetime = None, queues: typing.Set[cassiopeia.data.Queue] = None, seasons: typing.Set[cassiopeia.data.Season] = None, champions: typing.Set[cassiopeia.core.staticdata.champion.Champion] = None)

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

```

```
class cassiopeia.core.match.MatchHistory (summoner: typing.Union[cassiopeia.core.summoner.Summoner,  
                                                                    str, int] = None, account_id: int = None, re-  
                                                                    gion: typing.Union[cassiopeia.data.Region, str]  
                                                                    = None, begin_index: int = 0, end_index: int  
                                                                    = None, begin_time: datetime.datetime = None,  
                                                                    end_time: datetime.datetime = None, queues:  
                                                                    typing.Set[cassiopeia.data.Queue] = None,  
                                                                    seasons: typing.Set[cassiopeia.data.Season]  
                                                                    = None, champions: typing.  
                                                                    Set[cassiopeia.core.staticdata.champion.Champion]  
                                                                    = None)
```

Bases: cassiopeia.core.common.CassiopeiaGhostLazyList

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

append (*object*) → None – append object to end

begin_index

begin_time

champions

clear () → None – remove all items from L

contains (*item: typing.Any*) → bool

copy () → list – a shallow copy of L

count (*value*) → integer – return number of occurrences of value

delete (*item: typing.Any*) → None

end_index

end_time

enumerate (*item: typing.Any, reverse: bool = False*) → typing.Generator[[typing.Tuple[int, typ-
ing.Any], NoneType], NoneType]

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

index (*value* [, *start* [, *stop*]]) → integer – return first index of value.
Raises ValueError if the value is not present.

insert ()
L.insert(index, object) – insert object before index

platform

pop ([*index*]) → item – remove and return item at index (default last).
Raises IndexError if list is empty or index is out of range.

property (*load_group_or_method: typing.Any*) → typing.Union[property, typ-
ing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

queues

region

remove (*value*) → None – remove first occurrence of value.

Raises ValueError if the value is not present.

reverse ()

L.reverse() – reverse *IN PLACE*

search (*item: typing.Any, streaming: bool = False, reverse: bool = False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

seasons

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

summoner

class cassiopeia.**Match** (*, *id: int = None, region: typing.Union[cassiopeia.data.Region, str] = None*)

Bases: cassiopeia.core.common.CassiopeiaGhost

Searchable by ['Item', 'str', 'Map', 'Region', 'Season', 'GameMode', 'Platform', 'GameType', 'Champion', 'Queue']

blue_team

creation

duration

classmethod **from_match_reference** (*ref*)

id

map

mode

participants

platform

The platform for this match.

queue

red_team

region

The region for this match.

season

teams

timeline

type

version

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

Bases: cassiopeia.core.common.CassiopeiaObject

Searchable by ['str', 'bool']

bans

baron_kills

dominion_victory_score

dragon_kills

```
first_baron
first_blood
first_dragon
first_inhibitor
first_rift_herald
first_tower
classmethod from_data (data: cassiopeia.core.common.CoreData, participants: typing.List[cassiopeia.core.match.Participant])
inhibitor_kills
participants
rift_herald_kills
side
tower_kills
vilemaw_kills
win

class cassiopeia.core.match.Participant (**kwargs)
    Bases: cassiopeia.core.common.CassiopeiaObject
    Searchable by ['Side', 'str', 'Summoner', 'Champion', 'SummonerSpell', 'Rune', 'Mastery']
    champion
    classmethod from_data (data: cassiopeia.core.common.CoreData, match: cassiopeia.core.match.Match)
    id
    masteries
    rank_last_season
    runes
    side
    stats
    summoner
    summoner_spell_d
    summoner_spell_f
    team
    timeline
    version

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

`assists`
`champion_level`
`combat_player_score`
`damage_dealt_to_objectives`
`damage_dealt_to_turrets`
`damage_self_mitigated`
`deaths`
`double_kills`
`first_blood_assist`
`first_blood_kill`
`first_inhibitor_assist`
`first_inhibitor_kill`
`first_tower_assist`
`first_tower_kill`
`classmethod from_data` (*data:* `cassiopeia.core.match.ParticipantStatsData`, *match:* `cassiopeia.core.match.Match`)
`gold_earned`
`gold_spent`
`id`
`inhibitor_kills`
`items`
`killling_sprees`
`kills`
`largest_critical_strike`
`largest_killing_spree`
`largest_multi_kill`
`longest_time_spent_living`
`magic_damage_dealt`
`magic_damage_dealt_to_champions`
`magical_damage_taken`
`neutral_minions_killed`
`neutral_minions_killed_enemy_jungle`
`neutral_minions_killed_team_jungle`
`node_capture`
`node_capture_assist`
`node_neutralize`
`node_neutralize_assist`

objective_player_score
penta_kills
physical_damage_dealt
physical_damage_dealt_to_champions
physical_damage_taken
quadra_kills
sight_wards_bought_in_game
team_objective
time_CCing_others
total_damage_dealt
total_damage_dealt_to_champions
total_damage_taken
total_heal
total_minions_killed
total_player_score
total_score_rank
total_time_crowd_control_dealt
total_units_healed
triple_kills
true_damage_dealt
true_damage_dealt_to_champions
true_damage_taken
turret_kills
unreal_kills
vision_score
vision_wards_bought_in_game
wards_killed
wards_placed
win

```
class cassiopeia.core.match.ParticipantTimeline(**kwargs)
    Bases: cassiopeia.core.common.CassiopeiaObject
    creeps_per_min_deltas
    cs_diff_per_min_deltas
    damage_taken_diff_per_min_deltas
    damage_taken_per_min_deltas
    gold_per_min_deltas
```

```

    id
    lane
    role
    xp_diff_per_min_deltas
    xp_per_min_deltas
class cassiopeia.core.match.Timeline(*, id: int = None, region: typing.Union[cassiopeia.data.Region, str] = None)
    Bases: cassiopeia.core.common.CassiopeiaGhost
    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
    current_gold
    dominion_score
    jungle_minions_killed
    level
    minions_killed
    participant_id
    position
    team_score
    total_gold
    xp
class cassiopeia.core.match.Event(**kwargs)
    Bases: cassiopeia.core.common.CassiopeiaObject
    Searchable by ['str']
    after_id
    ascended_type
    assisting_participant_ids
    before_id
    building_type

```

creator_id
event_type
item_id
killer_id
lane_type
level_up_type
monster_sub_type
monster_type
participant_id
point_captured
position
skill_slot
team_id
timestamp
tower_type
type
 Legal values – 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
victim_id
ward_type

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

 x

 y

Profile Icons

cassiopeia.Summoner.**profile_icon**

cassiopeia.**get_profile_icons**(region: typing.Union[cassiopeia.data.Region, str] = None) → typing.List[cassiopeia.core.staticdata.profileicon.ProfileIcon]

class cassiopeia.**ProfileIcons**(*args, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None)
 Bases: cassiopeia.core.common.CassiopeiaGhostList

 append(object) → None – append object to end

 clear() → None – remove all items from L

 contains(item: typing.Any) → bool

 copy() → list – a shallow copy of L

 count(value) → integer – return number of occurrences of value

 delete(item: typing.Any) → None

```

enumerate (item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]
extend (iterable) → None – extend list by appending elements from the iterable
filter (function)
find (item: typing.Any, reverse: bool = False) → typing.Any
from_data (data: typing.Union[list, cassiopeia.core.common.CoreData])
index (value[, start[, stop]]) → integer – return first index of value.
    Raises ValueError if the value is not present.
insert ()
    L.insert(index, object) – insert object before index
locale
platform
pop ([index]) → item – remove and return item at index (default last).
    Raises IndexError if list is empty or index is out of range.
property (load_group_or_method: typing.Any) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]
region
remove (value) → None – remove first occurrence of value.
    Raises ValueError if the value is not present.
reverse ()
    L.reverse() – reverse IN PLACE
search (item: typing.Any, streaming: bool = False, reverse: bool = False) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]
sort (key=None, reverse=False) → None – stable sort *IN PLACE*
version
class cassiopeia.core.staticdata.profileicon.ProfileIcon (*, id: int, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None)

    Bases: cassiopeia.core.common.CassiopeiaGhost
    Searchable by ['str', 'Image', 'int']
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` (*region: typing.Union[cassiopeia.data.Region, str] = None*) → `cassiopeia.core.staticdata.realm.Realms`

class `cassiopeia.Realms` (*region: typing.Union[cassiopeia.data.Region, str] = None*)
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.Summoner.rune_pages`

`cassiopeia.get_runes` (*region: typing.Union[cassiopeia.data.Region, str] = None*) → `typing.List[cassiopeia.core.staticdata.rune.Rune]`

class `cassiopeia.Runes` (**args, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None*)
Bases: `cassiopeia.core.common.CassiopeiaGhostList`

append (*object*) → None – append object to end

clear () → None – remove all items from L

contains (*item: typing.Any*) → bool

copy () → list – a shallow copy of L

count (*value*) → integer – return number of occurrences of value

delete (*item: typing.Any*) → None

enumerate (*item: typing.Any, reverse: bool = False*) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

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

index (*value[, start[, stop]]*) → integer – return first index of value.
Raises ValueError if the value is not present.

insert ()
L.insert(index, object) – insert object before index

locale
The locale for this champion.

platform

pop (*[index]*) → item – remove and return item at index (default last).
Raises IndexError if list is empty or index is out of range.

property (*load_group_or_method: typing.Any*) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (*value*) → None – remove first occurrence of value.
Raises ValueError if the value is not present.

reverse ()
L.reverse() – reverse *IN PLACE*

search (*item: typing.Any, streaming: bool = False, reverse: bool = False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

version

class cassiopeia.Rune (*, *id: int = None, name: str = None, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None*)
Bases: cassiopeia.core.common.CassiopeiaGhost
Searchable by ['RuneType', 'str', 'Region', 'Platform', 'int']

description
The rune's description.

id
The rune's ID.

image
The image information for this rune.

included_data
The region for this rune.

locale
The locale for this rune.

name
The rune's name.

platform
The platform for this rune.

region
The region for this rune.

sanitized_description
The rune's sanitized description.

sprite

stats

tags

tier

type

version
The version for this rune.

```
class cassiopeia.core.runepage.RunePages (*args, summoner: typing.Union[cassiopeia.core.summoner.Summoner, int, str], region: typing.Union[cassiopeia.data.Region, str] = None)
    Bases: cassiopeia.core.common.CassiopeiaGhostList

    append(object) → None – append object to end
    clear() → None – remove all items from L
    contains(item: typing.Any) → bool
    copy() → list – a shallow copy of L
    count(value) → integer – return number of occurrences of value
    delete(item: typing.Any) → None
    enumerate(item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]
    extend(iterable) → None – extend list by appending elements from the iterable
    filter(function)
    find(item: typing.Any, reverse: bool = False) → typing.Any
    from_data(data: typing.Union[list, cassiopeia.core.common.CoreData])
    index(value[, start[, stop]]) → integer – return first index of value.
        Raises ValueError if the value is not present.
    insert()
        L.insert(index, object) – insert object before index
    platform
    pop([index]) → item – remove and return item at index (default last).
        Raises IndexError if list is empty or index is out of range.
    property(load_group_or_method: typing.Any) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]
```

region

remove (*value*) → None – remove first occurrence of value.
Raises ValueError if the value is not present.

reverse ()

L.reverse() – reverse *IN PLACE*

search (*item: typing.Any, streaming: bool = False, reverse: bool = False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (*key=None, reverse=False*) → None – stable sort **IN PLACE**

class cassiopeia.core.runepage.**RunePage** (***kwargs*)
Bases: cassiopeia.core.common.CassiopeiaGhost

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

current

id

locale

The locale for this rune page.

name

platform

The platform for this rune page.

region

runes

summoner

Status

cassiopeia.**get_status** (*region: typing.Union[cassiopeia.data.Region, str] = None*) → cassiopeia.core.status.ShardStatus

class cassiopeia.**ShardStatus** (*region: typing.Union[cassiopeia.data.Region, str] = None*)
Bases: cassiopeia.core.common.CassiopeiaGhost

Searchable by []

hostname

locales

name

platform

region

region_tag

services

slug

Spectator

```
cassiopeia.Summoner.current_match

cassiopeia.get_current_match (summoner: typing.Union[cassiopeia.core.summoner.Summoner,
int, str], region: typing.Union[cassiopeia.data.Region, str] =
None) → cassiopeia.core.spectator.CurrentMatch

cassiopeia.get_featured_matches (region: typing.Union[cassiopeia.data.Region, str] = None) →
cassiopeia.core.spectator.FeaturedMatches

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

    append (object) → None – append object to end

    clear () → None – remove all items from L

    client_refresh_interval

    contains (item: typing.Any) → bool

    copy () → list – a shallow copy of L

    count (value) → integer – return number of occurrences of value

    delete (item: typing.Any) → None

    enumerate (item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typ-
ing.Any], NoneType], NoneType]

    extend (iterable) → None – extend list by appending elements from the iterable

    filter (function)

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

    from_data (data: typing.Union[list, cassiopeia.core.common.CoreData])

    index (value[, start[, stop]]) → integer – return first index of value.
        Raises ValueError if the value is not present.

    insert ()
        L.insert(index, object) – insert object before index

    platform

    pop ([index]) → item – remove and return item at index (default last).
        Raises IndexError if list is empty or index is out of range.

    property (load_group_or_method: typing.Any) → typing.Union[property, typ-
ing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

    region

    remove (value) → None – remove first occurrence of value.
        Raises ValueError if the value is not present.

    reverse ()
        L.reverse() – reverse IN PLACE

    search (item: typing.Any, streaming: bool = False, reverse: bool = False) → typ-
ing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

    sort (key=None, reverse=False) → None – stable sort *IN PLACE*
```

```

class cassiopeia.core.spectator.CurrentMatch (*,          summoner:          typ-
                                                    ing.Union[cassiopeia.core.summoner.Summoner,
                                                    int, str] = None, region: typ-
                                                    ing.Union[cassiopeia.data.Region, str] =
                                                    None)
    Bases: cassiopeia.core.common.CassiopeiaGhost
    Searchable by []
    blue_team
    creation
    duration
    exists
    classmethod from_data (data: cassiopeia.core.spectator.CurrentGameInfoData, summoner: typ-
                            ing.Union[cassiopeia.core.summoner.Summoner, int, str])
    id
    map
    mode
    observer_key
    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, region: cassiopeia.data.Region)
    participants
class cassiopeia.core.spectator.Participant (**kwargs)
    Bases: cassiopeia.core.common.CassiopeiaObject
    Searchable by []
    champion
    classmethod from_data (data: cassiopeia.core.common.CoreData, region: cassiopeia.data.Region)
    is_bot
    masteries
    runes
    summoner
    summoner_spell_d

```

```
summoner_spell_f
team
```

Summoners

```
cassiopeia.get_summoner(*, id: int = None, account: typing.Union[cassiopeia.core.summoner.Account, int] = None, name: str = None, region: typing.Union[cassiopeia.data.Region, str] = None) → cassiopeia.core.summoner.Summoner
```

```
class cassiopeia.Summoner(*, id: int = None, account: typing.Union[cassiopeia.core.summoner.Account, int] = None, name: str = None, region: typing.Union[cassiopeia.data.Region, str] = None)
```

```
    Bases: cassiopeia.core.common.CassiopeiaGhost
```

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

```
    account
```

```
    champion_masters
```

```
    current_match
```

```
    exists
```

```
    id
```

```
    league_positions
```

```
    leagues
```

```
    level
```

```
    mastery_pages
```

```
    match_history
```

```
    match_history_uri
```

```
    name
```

```
    platform
```

```
        The platform for this summoner.
```

```
    profile_icon
```

```
    region
```

```
        The region for this summoner.
```

```
    revision_date
```

```
    rune_pages
```

```
class cassiopeia.core.summoner.Account(**kwargs)
```

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

```
    Searchable by ['int']
```

```
    id
```

Summoner Spells

`cassiopeia.get_summoner_spells` (*region: typing.Union[cassiopeia.data.Region, str] = None*) → `typing.List[cassiopeia.core.staticdata.summonerspell.SummonerSpell]`

class `cassiopeia.SummonerSpells` (*args, *region: typing.Union[cassiopeia.data.Region, str] = None*, *version: str = None*, *locale: str = None*, *included_data: typing.Set[str] = None*)

Bases: `cassiopeia.core.common.CassiopeiaGhostList`

append (*object*) → None – append object to end

clear () → None – remove all items from L

contains (*item: typing.Any*) → bool

copy () → list – a shallow copy of L

count (*value*) → integer – return number of occurrences of value

delete (*item: typing.Any*) → None

enumerate (*item: typing.Any*, *reverse: bool = False*) → `typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]`

extend (*iterable*) → None – extend list by appending elements from the iterable

filter (*function*)

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

from_data (*data: typing.Union[list, cassiopeia.core.common.CoreData]*)

included_data

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

index (*value* [, *start* [, *stop*]]) → integer – return first index of value.
Raises `ValueError` if the value is not present.

insert ()

L.insert(index, object) – insert object before index

locale

The locale for this champion.

platform

pop ([*index*]) → item – remove and return item at index (default last).
Raises `IndexError` if list is empty or index is out of range.

property (*load_group_or_method: typing.Any*) → `typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]`

region

remove (*value*) → None – remove first occurrence of value.
Raises `ValueError` if the value is not present.

reverse ()

L.reverse() – reverse *IN PLACE*

search (*item: typing.Any*, *streaming: bool = False*, *reverse: bool = False*) → `typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]`

sort (*key=None*, *reverse=False*) → None – stable sort **IN PLACE**

version

```
class cassiopeia.SummonerSpell(*, id: int = None, name: str = None, region: typing.Union[cassiopeia.data.Region, str] = None, version: str = None, locale: str = None, included_data: typing.Set[str] = None)
```

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 (region: typing.Union[cassiopeia.data.Region, str] = None) → typing.List[str]
```

```
cassiopeia.get_version (date: datetime.date = None, region: typing.Union[cassiopeia.data.Region, str] = None) → typing.Union[NoneType, str]
```

```
class cassiopeia.Versions (*args, region: typing.Union[cassiopeia.data.Region, str] = None)
```

Bases: cassiopeia.core.common.CassiopeiaGhostList

append (object) → None – append object to end

clear () → None – remove all items from L

contains (item: typing.Any) → bool

copy () → list – a shallow copy of L

count (value) → integer – return number of occurrences of value

delete (item: typing.Any) → None

enumerate (item: typing.Any, reverse: bool = False) → typing.Generator[[typing.Tuple[int, typing.Any], NoneType], NoneType]

extend (iterable) → None – extend list by appending elements from the iterable

filter (function)

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

from_data (data: typing.Union[list, cassiopeia.core.common.CoreData])

index (value[, start[, stop]]) → integer – return first index of value.

Raises ValueError if the value is not present.

insert ()
L.insert(index, object) – insert object before index

platform

pop ([*index*]) → item – remove and return item at index (default last).
Raises IndexError if list is empty or index is out of range.

property (*load_group_or_method*: *typing.Any*) → typing.Union[property, typing.Callable[[typing.Callable[[typing.Any], typing.Any]], property]]

region

remove (*value*) → None – remove first occurrence of value.
Raises ValueError if the value is not present.

reverse ()
L.reverse() – reverse *IN PLACE*

search (*item*: *typing.Any*, *streaming*: *bool* = *False*, *reverse*: *bool* = *False*) → typing.Union[typing.SearchableList, typing.Generator[[typing.Any, NoneType], NoneType]]

sort (*key*=*None*, *reverse*=*False*) → None – stable sort **IN PLACE**

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.Division
    Bases: enum.Enum

    An enumeration.

    five = 'V'
    four = 'IV'
    one = 'I'
    three = 'III'
    two = 'II'

class cassiopeia.data.GameMode
    Bases: enum.Enum

    An enumeration.

    aram = 'ARAM'
    arsr = 'ARSR'
    ascension = 'ASCENSION'
    assassinate = 'ASSASSINATE'
    classic = 'CLASSIC'
    dark_star = 'DARKSTAR'
    dominion = 'ODIN'
    doom_bots = 'DOOMBOTSTEEMO'
    nexus_siege = 'SIEGE'
```

```
one_for_all = 'ONEFORALL'
poro_king = 'KINGPORO'
showdown = 'FIRSTBLOOD'
star_guardian = 'STARGUARDIAN'
tutorial = 'TUTORIAL'
urf = 'URF'

class cassiopeia.data.GameType
    Bases: enum.Enum

    An enumeration.

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

class cassiopeia.data.Lane
    Bases: enum.Enum

    An enumeration.

    bot_lane = 'BOT_LANE'
    mid_lane = 'MID_LANE'
    top_lane = 'TOP_LANE'

class cassiopeia.data.MasteryTree
    Bases: enum.Enum

    An enumeration.

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

class cassiopeia.data.Platform
    Bases: enum.Enum

    An enumeration.

    brazil = 'BR1'
    default_locale
    europe_north_east = 'EUN1'
    europe_west = 'EUW1'
    japan = 'JP1'
    korea = 'KR'
    latin_america_north = 'LA1'
    latin_america_south = 'LA2'
    north_america = 'NA1'
    oceania = 'OC1'
```

```
pbe = 'PBE1'
region
russia = 'RU'
turkey = 'TR1'
class cassiopeia.data.Queue
    Bases: enum.Enum
    An enumeration.
    all_random_summoners_rift = 'ARSR_5x5'
    all_random_urf = 'ARURF_5X5'
    aram = 'ARAM_5x5'
    ascension = 'ASCENSION_5x5'
    black_market = 'BILGEWATER_5x5'
    blood_hunt = 'ASSASSINATE_5x5'
    bot_beginner_fives = 'BOT_5x5_BEGINNER'
    bot_dominion = 'BOT_ODIN_5x5'
    bot_fives = 'BOT_5x5'
    bot_intermediate_fives = 'BOT_5x5_INTERMEDIATE'
    bot_intro_fives = 'BOT_5x5_INTRO'
    bot_threes = 'BOT_TT_3x3'
    bot_urf = 'BOT_URF_5x5'
    butchers_bridge = 'BILGEWATER_ARAM_5x5'
    custom = 'CUSTOM'
    darkstar = 'DARKSTAR_3x3'
    definitely_not_dominion = 'DEFINITELY_NOT_DOMINION_5x5'
    dominion_blind = 'ODIN_5x5_BLIND'
    dominion_draft = 'ODIN_5x5_DRAFT'
    doom_bots_1 = 'NIGHTMARE_BOT_5x5_RANK1'
    doom_bots_2 = 'NIGHTMARE_BOT_5x5_RANK2'
    doom_bots_5 = 'NIGHTMARE_BOT_5x5_RANK5'
    dynamic_queue = 'TEAM_BUILDER_DRAFT_UNRANKED_5x5'
    flex = 'RANKED_FLEX_SR'
    flex_threes = 'RANKED_FLEX_TT'
    from_id(id: int)
    hexakill_summoners_rift = 'SR_6x6'
    hexakill_twisted_treeline = 'HEXAKILL'
    nemesis_draft = 'COUNTER_PICK'
```

```

nexus_siege = 'SIEGE'
normal_blind_fives = 'NORMAL_5x5_BLIND'
normal_blind_threes = 'NORMAL_3x3'
normal_draft_fives = 'NORMAL_5x5_DRAFT'
one_for_all = 'ONEFORALL_5x5'
one_for_all_mirror = 'ONEFORALL_MIRRORMODE_5x5'
poro_king = 'KING_PORO_5x5'
ranked_dynamic_queue = 'TEAM_BUILDER_DRAFT_RANKED_5x5'
ranked_fives = 'RANKED_TEAM_5x5'
ranked_premade_fives = 'RANKED_PREMADE_5x5'
ranked_premade_threes = 'RANKED_PREMADE_3x3'
ranked_solo = 'RANKED_SOLO_5x5'
ranked_solo_queue = 'TEAM_BUILDER_RANKED_SOLO'
ranked_threes = 'RANKED_TEAM_3x3'
showdown_duo = 'FIRSTBLOOD_2x2'
showdown_solo = 'FIRSTBLOOD_1x1'
team_builder = 'GROUP_FINDER_5x5'
urf = 'URF_5x5'

```

```
class cassiopeia.data.Region
```

```
Bases: enum.Enum
```

```
An enumeration.
```

```
brazil = 'BR'
```

```
default_locale
```

```
europe_north_east = 'EUNE'
```

```
europe_west = 'EUW'
```

```
japan = 'JP'
```

```
korea = 'KR'
```

```
latin_america_north = 'LAN'
```

```
latin_america_south = 'LAS'
```

```
north_america = 'NA'
```

```
oceania = 'OCE'
```

```
pbe = 'PBE'
```

```
platform
```

```
russia = 'RU'
```

```
turkey = 'TR'
```

```
class cassiopeia.data.Resource
    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
    Bases: enum.Enum

    An enumeration.

    adc = 'DUO_CARRY'
    jungle = 'JUNGLE'
    middle = 'MIDDLE'
    support = 'DUO_SUPPORT'
    top = 'TOP'

class cassiopeia.data.RuneType
    Bases: enum.Enum

    An enumeration.

    glyph = 'blue'
    mark = 'red'
    quint = 'black'
    seal = 'yellow'

class cassiopeia.data.Season
    Bases: enum.Enum

    An enumeration.

    from_id(id: int)

    preseason_3 = 'PRESEASON3'
    preseason_4 = 'PRESEASON2014'
```

```

preseason_5 = 'PRESEASON2015'
preseason_6 = 'PRESEASON2016'
preseason_7 = 'PRESEASON2017'
season_3 = 'SEASON3'
season_4 = 'SEASON2014'
season_5 = 'SEASON2015'
season_6 = 'SEASON2016'
season_7 = 'SEASON2017'

class cassiopeia.data.Side
    Bases: enum.Enum

    An enumeration.

    blue = 100
    red = 200

class cassiopeia.data.Tier
    Bases: enum.Enum

    An enumeration.

    bronze = 'BRONZE'
    challenger = 'CHALLENGER'
    diamond = 'DIAMOND'
    gold = 'GOLD'
    master = 'MASTER'
    platinum = 'PLATINUM'
    silver = 'SILVER'
    unranked = 'UNRANKED'

class cassiopeia.patches.Patch(season: str, name: str, start: typing.Union[datetime.datetime, float],
                                end: typing.Union[datetime.datetime, float, NoneType])
    Bases: object

    end

    classmethod from_date (date: datetime.datetime) → cassiopeia.patches.Patch
    classmethod from_str (string: str) → cassiopeia.patches.Patch

    major
    majorminor
    minor
    name
    revision
    season
    start

```

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.

Install from Source

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

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

On Windows, follow the instructions [here](#). Note that if you need multiple paths on your PYTHONPATH, you can separate them 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 PYTHONPATH should end in `.../cassiopeia`.

Setting Your API Key and Other Settings

By default, Cass's settings are stored in a json file located `cassiopeia/configuration/default.json`. You can modify this file or create a custom settings file and pass it in as the first argument to your program.

In order to set your API key, Cass will look for an environment variable on your system called `RIOT_API_KEY`. You can change this by manually specifying your API key in your settings file, or changing the name of the environment variable. 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 your terminal/IDE.

In your settings file, you can also set a default region. This region will be used if you did not provide a region when using Cassiopeia objects.

We will add more customizable settings in the future, so feel free to check this section every so often.

Settings

That 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. A default settings file (`cassiopeia/configuration/default.json`) will be used if you do not run `cass.apply_settings`.

The most important settings are the Riot API key and the default region. Both of these can be set programmatically, and if not set they will be read from your settings file. See `cassiopeia/configuration/default.json` for an up to date example of a settings file.

Each setting is explained below, and should be added as separate entries to the settings file (in `json` format) or into a python dictionary.

Globals

The `"default_region"` 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 `null`. 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 `null`. `"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 `null`, 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.

Pipeline

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

Example (note that the `SimpleKVDiskStore` and `ChampionGG` are not included by default):

```
{
  "pipeline": {
    "Cache": {},

    "SimpleKVDiskStore": {
      "package": "cassiopeia-datastores.diskstore.diskstore"
    },

    "DDragon": {},

    "RiotAPI": {
      "api_key": "RIOT_API_KEY"
    },

    "ChampionGG": {
      "package": "cassiopeia-plugins.championgg.championgg",
      "api_key": "CHAMPIONGG_KEY"
    }
  }
}
```

```
}  
}
```

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`.

The `request_by_id` variable determines whether the Riot API will request static data and champion statuses by id when a single piece of data is accessed, or whether it will request all the champions/items/etc when one is asked for. The default is `True`, meaning that individual elements will be requested one at a time. Be aware that you may quickly hit your rate limit if you aren't careful.

Request Handling

The `"request_handling"` variable specifies how Riot API errors should be handled. There are three options, each of which have their own set of parameters: `"throw"` simply causes an 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:

```
"Riot API": {  
  "api_key": "RIOT_API_KEY",  
  "limiting_share": 1.0,  
  "request_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"
}
}
}

```

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.

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. For example, `cass.get_summoner(name="Kalturi")` and `Summoner(name="Kalturi")` both work.

Settings

There are a few settings in Cass that should be modified, and more that can. We provide default settings in the file `cassiopeia/configuration/default.ini`.

Most importantly, your API key should be set. You can set this by providing the key itself, or by providing the name of an environment variable that contains your API key. We encourage you to put your API key in an environment variable, but if you put your API key in the file just be careful not to push it to version control.

A default region can be set in the settings, and any object that requires a region but isn't provided when during initialization will use the default region.

(Not yet implemented) All Cass objects have expiration timeouts that can be modified. See the [Data Pipeline](#) section below.

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")` followed by `kalturi.level`. The latter makes a call to the data pipeline (discussed below) to pull the rest of the data for `kalturi` by using `kalturi.name`.

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

The implementation for ghost objects can be found in our Meraki Commons repository on GitHub.

Data Pipeline

The data pipeline is the series of caches, databases, and data sources (such as the Riot API) that both provide and store data. Some parts of the pipeline are just data sources (the Riot API), while many are both data sources and data sinks (caches and databases). Data sources provide data, while data sinks store data. The sources and sinks in the data pipeline are in a specific order, usually with faster data sources at the beginning and slower ones at the end.

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). The query is passed up the data pipeline through the data sources, and at each data source the data pipeline asks if that source has the object corresponding to the query. If the source does contain the object, it is returned. If the source does not contain 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. An 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.

(Not yet implemented) Expiration times for objects are allowed. If an object in a data sink expires, it will be removed from the data sink. This happens periodically when data is queried. Users can force all expired objects in data sinks to be removed using `settings.pipeline.expire()`.

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(account=27994129).match_history["Teemo"]
```

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

```
all_teemo_games = Summoner(account=27994129).match_history.find("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.

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

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.

(Not yet implemented) 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 the default expiration periods below.

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 no parameters (i.e. `{ }`).

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.

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.

The simple disk store takes no parameters except it's package location, which is `cassiopeia-datastores.diskstore`.

ChampionGG

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

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 relative path to 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

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-plugins.championgg.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

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-datastores.diskstore.diskstore",
        "path": "/absolute/path/to/store/data/"
    },
    ...
}
```

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 our [contributions page](#).

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, Mongo, and SQLAlchemy. To do so, new datasources should be added (along side the Riot API and the in-memory cache) to support each of these.
- Add `get_many_*` functions to the ddragon data source, similar to the riotapi data source.
- 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.
- The patch dates are approximate and are the same for all regions. Instead, the patches should be set per-region and should be as precise of a timestamp as we can get. We can figure this out by pulling matches near when patches change, and carefully checking the creation date of the match and it's patch / version. If someone can help us get precise patch values or even write us a script for doing that, it would be a huge help.
- Currently, the patches file needs to be updated with the correct start date every time a new patch is released. There must be some way to automate this.
- Allow `MatchHistory` objects to be indexable by patch number. Note that version numbers (e.g. "7.14.2") are more strict than patch numbers (e.g. "7.14") if that matters.

CHAPTER 6

Top Level APIs

CHAPTER 7

Index and Search

- `genindex`
- `modindex`
- `search`

C

`cassiopeia.data`, [48](#)

`cassiopeia.patches`, [53](#)

A

- ability_power (cassiopeia.core.staticdata.item.ItemStats attribute), 19
- account (cassiopeia.Summoner attribute), 44
- Account (class in cassiopeia.core.summoner), 44
- adc (cassiopeia.data.Role attribute), 52
- after_id (cassiopeia.core.match.Event attribute), 35
- all_random_summoners_rift (cassiopeia.data.Queue attribute), 50
- all_random_urf (cassiopeia.data.Queue attribute), 50
- ally_tips (cassiopeia.Champion attribute), 11
- altars_captured (cassiopeia.core.match.ParticipantStats attribute), 32
- altars_neutralized (cassiopeia.core.match.ParticipantStats attribute), 32
- alternative_images (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 14
- alternative_images (cassiopeia.SummonerSpell attribute), 46
- append() (cassiopeia.ChampionMasteries method), 16
- append() (cassiopeia.Champions method), 10
- append() (cassiopeia.core.league.LeagueEntries method), 23
- append() (cassiopeia.core.league.Leagues method), 21
- append() (cassiopeia.core.masterypage.MasteryPages method), 28
- append() (cassiopeia.core.match.MatchHistory method), 30
- append() (cassiopeia.core.runepage.RunePages method), 40
- append() (cassiopeia.FeaturedMatches method), 42
- append() (cassiopeia.Items method), 17
- append() (cassiopeia.Locales method), 24
- append() (cassiopeia.Maps method), 25
- append() (cassiopeia.Masteries method), 26
- append() (cassiopeia.ProfileIcons method), 36
- append() (cassiopeia.Runes method), 38
- append() (cassiopeia.SummonerSpells method), 45
- append() (cassiopeia.Versions method), 47
- aram (cassiopeia.data.GameMode attribute), 48
- aram (cassiopeia.data.Queue attribute), 50
- armor (cassiopeia.core.staticdata.champion.Stats attribute), 12
- armor (cassiopeia.core.staticdata.item.ItemStats attribute), 19
- armor_per_level (cassiopeia.core.staticdata.champion.Stats attribute), 12
- arsr (cassiopeia.data.GameMode attribute), 48
- ascended_type (cassiopeia.core.match.Event attribute), 35
- ascension (cassiopeia.data.GameMode attribute), 48
- ascension (cassiopeia.data.Queue attribute), 50
- assassinate (cassiopeia.data.GameMode attribute), 48
- assisting_participant_ids (cassiopeia.core.match.Event attribute), 35
- assists (cassiopeia.core.match.ParticipantStats attribute), 32
- attack (cassiopeia.core.staticdata.champion.Info attribute), 12
- attack_damage (cassiopeia.core.staticdata.champion.Stats attribute), 12
- attack_damage (cassiopeia.core.staticdata.item.ItemStats attribute), 19
- attack_damage_per_level (cassiopeia.core.staticdata.champion.Stats attribute), 12
- attack_range (cassiopeia.core.staticdata.champion.Stats attribute), 12
- attack_speed (cassiopeia.core.staticdata.champion.Stats attribute), 12
- attack_speed (cassiopeia.core.staticdata.item.ItemStats attribute), 19

B

- bans (cassiopeia.core.match.Team attribute), 31
- bans (cassiopeia.core.spectator.Team attribute), 43
- baron_kills (cassiopeia.core.match.Team attribute), 31
- before_id (cassiopeia.core.match.Event attribute), 35

- `begin_index` (cassiopeia.core.match.MatchHistory attribute), 30
- `begin_time` (cassiopeia.core.match.MatchHistory attribute), 30
- `black_market` (cassiopeia.data.Queue attribute), 50
- `block` (cassiopeia.core.staticdata.item.ItemStats attribute), 19
- `blood_hunt` (cassiopeia.data.Queue attribute), 50
- `blood_well` (cassiopeia.data.Resource attribute), 52
- `blue` (cassiopeia.data.Side attribute), 53
- `blue_team` (cassiopeia.core.spectator.CurrentMatch attribute), 43
- `blue_team` (cassiopeia.Match attribute), 31
- `blurb` (cassiopeia.Champion attribute), 11
- `bot_beginner_fives` (cassiopeia.data.Queue attribute), 50
- `bot_dominion` (cassiopeia.data.Queue attribute), 50
- `bot_fives` (cassiopeia.data.Queue attribute), 50
- `bot_intermediate_fives` (cassiopeia.data.Queue attribute), 50
- `bot_intro_fives` (cassiopeia.data.Queue attribute), 50
- `bot_lane` (cassiopeia.data.Lane attribute), 49
- `bot_threes` (cassiopeia.data.Queue attribute), 50
- `bot_urf` (cassiopeia.data.Queue attribute), 50
- `brazil` (cassiopeia.data.Platform attribute), 49
- `brazil` (cassiopeia.data.Region attribute), 51
- `bronze` (cassiopeia.data.Tier attribute), 53
- `building_type` (cassiopeia.core.match.Event attribute), 35
- `builds_from` (cassiopeia.Item attribute), 18
- `builds_into` (cassiopeia.Item attribute), 18
- `butchers_bridge` (cassiopeia.data.Queue attribute), 50
- ## C
- `cassiopeia.data` (module), 48
- `cassiopeia.patches` (module), 53
- `cdn` (cassiopeia.Realms attribute), 38
- `challenger` (cassiopeia.data.Tier attribute), 53
- `ChallengerLeague` (class in cassiopeia), 22
- `champion` (cassiopeia.ChampionMastery attribute), 17
- `champion` (cassiopeia.core.match.Participant attribute), 32
- `champion` (cassiopeia.core.spectator.Participant attribute), 43
- `champion` (cassiopeia.Item attribute), 18
- `Champion` (class in cassiopeia), 11
- `champion_key` (cassiopeia.core.staticdata.champion.Skin attribute), 13
- `champion_level` (cassiopeia.core.match.ParticipantStats attribute), 33
- `champion_masteries` (cassiopeia.Summoner attribute), 44
- `ChampionMasteries` (class in cassiopeia), 16
- `ChampionMastery` (class in cassiopeia), 16
- `champions` (cassiopeia.core.match.MatchHistory attribute), 30
- `Champions` (class in cassiopeia), 10
- `ChampionSpell` (class in cassiopeia.core.staticdata.champion), 14
- `chest_granted` (cassiopeia.ChampionMastery attribute), 17
- `classic` (cassiopeia.data.GameMode attribute), 48
- `clear()` (cassiopeia.ChampionMasteries method), 16
- `clear()` (cassiopeia.Champions method), 10
- `clear()` (cassiopeia.core.league.LeagueEntries method), 23
- `clear()` (cassiopeia.core.league.Leagues method), 21
- `clear()` (cassiopeia.core.masterypage.MasteryPages method), 28
- `clear()` (cassiopeia.core.match.MatchHistory method), 30
- `clear()` (cassiopeia.core.runepage.RunePages method), 40
- `clear()` (cassiopeia.FeaturedMatches method), 42
- `clear()` (cassiopeia.Items method), 17
- `clear()` (cassiopeia.Locales method), 24
- `clear()` (cassiopeia.Maps method), 25
- `clear()` (cassiopeia.Masteries method), 26
- `clear()` (cassiopeia.ProfileIcons method), 36
- `clear()` (cassiopeia.Runes method), 38
- `clear()` (cassiopeia.SummonerSpells method), 45
- `clear()` (cassiopeia.Versions method), 47
- `client_refresh_interval` (cassiopeia.FeaturedMatches attribute), 42
- `coefficients` (cassiopeia.core.staticdata.champion.SpellVars attribute), 15
- `coefficients` (cassiopeia.core.staticdata.summonerspell.SpellVars attribute), 47
- `combat_player_score` (cassiopeia.core.match.ParticipantStats attribute), 33
- `consume_on_full` (cassiopeia.Item attribute), 18
- `consumed` (cassiopeia.Item attribute), 18
- `contains()` (cassiopeia.ChampionMasteries method), 16
- `contains()` (cassiopeia.Champions method), 10
- `contains()` (cassiopeia.core.league.LeagueEntries method), 23
- `contains()` (cassiopeia.core.league.Leagues method), 21
- `contains()` (cassiopeia.core.masterypage.MasteryPages method), 28
- `contains()` (cassiopeia.core.match.MatchHistory method), 30
- `contains()` (cassiopeia.core.runepage.RunePages method), 40
- `contains()` (cassiopeia.FeaturedMatches method), 42
- `contains()` (cassiopeia.Items method), 17
- `contains()` (cassiopeia.Locales method), 24
- `contains()` (cassiopeia.Maps method), 25
- `contains()` (cassiopeia.Masteries method), 26
- `contains()` (cassiopeia.ProfileIcons method), 36
- `contains()` (cassiopeia.Runes method), 38
- `contains()` (cassiopeia.SummonerSpells method), 45
- `contains()` (cassiopeia.Versions method), 47

- cooldowns (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 14
- cooldowns (cassiopeia.SummonerSpell attribute), 46
- coop_ai_enabled (cassiopeia.Champion attribute), 11
- copy() (cassiopeia.ChampionMasteries method), 16
- copy() (cassiopeia.Champions method), 10
- copy() (cassiopeia.core.league.LeagueEntries method), 23
- copy() (cassiopeia.core.league.Leagues method), 21
- copy() (cassiopeia.core.masterypage.MasteryPages method), 28
- copy() (cassiopeia.core.match.MatchHistory method), 30
- copy() (cassiopeia.core.runepage.RunePages method), 40
- copy() (cassiopeia.FeaturedMatches method), 42
- copy() (cassiopeia.Items method), 17
- copy() (cassiopeia.Locales method), 24
- copy() (cassiopeia.Maps method), 25
- copy() (cassiopeia.Masteries method), 26
- copy() (cassiopeia.ProfileIcons method), 36
- copy() (cassiopeia.Runes method), 38
- copy() (cassiopeia.SummonerSpells method), 45
- copy() (cassiopeia.Versions method), 47
- costs (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 14
- costs (cassiopeia.SummonerSpell attribute), 46
- count() (cassiopeia.ChampionMasteries method), 16
- count() (cassiopeia.Champions method), 10
- count() (cassiopeia.core.league.LeagueEntries method), 23
- count() (cassiopeia.core.league.Leagues method), 21
- count() (cassiopeia.core.masterypage.MasteryPages method), 28
- count() (cassiopeia.core.match.MatchHistory method), 30
- count() (cassiopeia.core.runepage.RunePages method), 40
- count() (cassiopeia.FeaturedMatches method), 42
- count() (cassiopeia.Items method), 17
- count() (cassiopeia.Locales method), 24
- count() (cassiopeia.Maps method), 25
- count() (cassiopeia.Masteries method), 26
- count() (cassiopeia.ProfileIcons method), 36
- count() (cassiopeia.Runes method), 38
- count() (cassiopeia.SummonerSpells method), 45
- count() (cassiopeia.Versions method), 47
- courage (cassiopeia.data.Resource attribute), 52
- creation (cassiopeia.core.spectator.CurrentMatch attribute), 43
- creation (cassiopeia.Match attribute), 31
- creator_id (cassiopeia.core.match.Event attribute), 35
- creeps_per_min_deltas (cassiopeia.core.match.ParticipantTimeline attribute), 34
- crimson_rush (cassiopeia.data.Resource attribute), 52
- critical_strike_chance (cassiopeia.core.staticdata.champion.Stats attribute), 12
- critical_strike_chance (cassiopeia.core.staticdata.item.ItemStats attribute), 19
- critical_strike_chance_per_level (cassiopeia.core.staticdata.champion.Stats attribute), 12
- critical_strike_damage (cassiopeia.core.staticdata.item.ItemStats attribute), 19
- cs_diff_per_min_deltas (cassiopeia.core.match.ParticipantTimeline attribute), 34
- css_version (cassiopeia.Realms attribute), 38
- cunning (cassiopeia.data.MasteryTree attribute), 49
- current (cassiopeia.core.masterypage.MasteryPage attribute), 29
- current (cassiopeia.core.runepage.RunePage attribute), 41
- current_gold (cassiopeia.core.match.ParticipantFrame attribute), 35
- current_match (cassiopeia.cassiopeia.Summoner attribute), 42
- current_match (cassiopeia.Summoner attribute), 44
- CurrentMatch (class in cassiopeia.core.spectator), 42
- custom (cassiopeia.data.GameType attribute), 49
- custom (cassiopeia.data.Queue attribute), 50
- custom_enabled (cassiopeia.Champion attribute), 11
- ## D
- damage_dealt_to_objectives (cassiopeia.core.match.ParticipantStats attribute), 33
- damage_dealt_to_turrets (cassiopeia.core.match.ParticipantStats attribute), 33
- damage_self_mitigated (cassiopeia.core.match.ParticipantStats attribute), 33
- damage_taken_diff_per_min_deltas (cassiopeia.core.match.ParticipantTimeline attribute), 34
- damage_taken_per_min_deltas (cassiopeia.core.match.ParticipantTimeline attribute), 34
- dark_star (cassiopeia.data.GameMode attribute), 48
- darkstar (cassiopeia.data.Queue attribute), 50
- deaths (cassiopeia.core.match.ParticipantStats attribute), 33
- default_locale (cassiopeia.data.Platform attribute), 49
- default_locale (cassiopeia.data.Region attribute), 51
- defense (cassiopeia.core.staticdata.champion.Info attribute), 12

definitely_not_dominion (cassiopeia.data.Queue attribute), 50

delete() (cassiopeia.ChampionMasteries method), 16

delete() (cassiopeia.Champions method), 10

delete() (cassiopeia.core.league.LeagueEntries method), 23

delete() (cassiopeia.core.league.Leagues method), 21

delete() (cassiopeia.core.masterypage.MasteryPages method), 28

delete() (cassiopeia.core.match.MatchHistory method), 30

delete() (cassiopeia.core.runepage.RunePages method), 40

delete() (cassiopeia.FeaturedMatches method), 42

delete() (cassiopeia.Items method), 17

delete() (cassiopeia.Locales method), 24

delete() (cassiopeia.Maps method), 25

delete() (cassiopeia.Masteries method), 26

delete() (cassiopeia.ProfileIcons method), 36

delete() (cassiopeia.Runes method), 38

delete() (cassiopeia.SummonerSpells method), 45

delete() (cassiopeia.Versions method), 47

description (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 14

description (cassiopeia.core.staticdata.champion.Passive attribute), 13

description (cassiopeia.Item attribute), 18

description (cassiopeia.Mastery attribute), 27

description (cassiopeia.Rune attribute), 39

description (cassiopeia.SummonerSpell attribute), 46

diamond (cassiopeia.data.Tier attribute), 53

difficulty (cassiopeia.core.staticdata.champion.Info attribute), 12

division (cassiopeia.core.league.LeagueEntry attribute), 24

Division (class in cassiopeia.data), 48

dodge (cassiopeia.core.staticdata.item.ItemStats attribute), 19

dominion (cassiopeia.data.GameMode attribute), 48

dominion_blind (cassiopeia.data.Queue attribute), 50

dominion_draft (cassiopeia.data.Queue attribute), 50

dominion_score (cassiopeia.core.match.ParticipantFrame attribute), 35

dominion_victory_score (cassiopeia.core.match.Team attribute), 31

doom_bots (cassiopeia.data.GameMode attribute), 48

doom_bots_1 (cassiopeia.data.Queue attribute), 50

doom_bots_2 (cassiopeia.data.Queue attribute), 50

doom_bots_5 (cassiopeia.data.Queue attribute), 50

double_kills (cassiopeia.core.match.ParticipantStats attribute), 33

dragon_kills (cassiopeia.core.match.Team attribute), 31

duration (cassiopeia.core.spectator.CurrentMatch attribute), 43

duration (cassiopeia.Match attribute), 31

dynamic (cassiopeia.core.staticdata.champion.SpellVars attribute), 15

dynamic (cassiopeia.core.staticdata.summonerspell.SpellVars attribute), 47

dynamic_queue (cassiopeia.data.Queue attribute), 50

E

effect (cassiopeia.Item attribute), 18

effects (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 14

effects (cassiopeia.SummonerSpell attribute), 46

effects_by_level (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 14

enabled (cassiopeia.Champion attribute), 11

end (cassiopeia.patches.Patch attribute), 53

end_index (cassiopeia.core.match.MatchHistory attribute), 30

end_time (cassiopeia.core.match.MatchHistory attribute), 30

enemy_tips (cassiopeia.Champion attribute), 11

energy (cassiopeia.core.staticdata.item.ItemStats attribute), 19

energy (cassiopeia.data.Resource attribute), 52

energy_regen (cassiopeia.core.staticdata.item.ItemStats attribute), 19

entries (cassiopeia.ChallengerLeague attribute), 22

entries (cassiopeia.core.league.League attribute), 22

entries (cassiopeia.MasterLeague attribute), 22

enumerate() (cassiopeia.ChampionMasteries method), 16

enumerate() (cassiopeia.Champions method), 10

enumerate() (cassiopeia.core.league.LeagueEntries method), 23

enumerate() (cassiopeia.core.league.Leagues method), 21

enumerate() (cassiopeia.core.masterypage.MasteryPages method), 28

enumerate() (cassiopeia.core.match.MatchHistory method), 30

enumerate() (cassiopeia.core.runepage.RunePages method), 40

enumerate() (cassiopeia.FeaturedMatches method), 42

enumerate() (cassiopeia.Items method), 17

enumerate() (cassiopeia.Locales method), 24

enumerate() (cassiopeia.Maps method), 25

enumerate() (cassiopeia.Masteries method), 27

enumerate() (cassiopeia.ProfileIcons method), 37

enumerate() (cassiopeia.Runes method), 38

enumerate() (cassiopeia.SummonerSpells method), 45

enumerate() (cassiopeia.Versions method), 47

europe_north_east (cassiopeia.data.Platform attribute), 49

europe_north_east (cassiopeia.data.Region attribute), 51

europe_west (cassiopeia.data.Platform attribute), 49

europe_west (cassiopeia.data.Region attribute), 51

Event (class in cassiopeia.core.match), 35

- event_type (cassiopeia.core.match.Event attribute), 36
 - events (cassiopeia.core.match.Frame attribute), 35
 - exists (cassiopeia.core.spectator.CurrentMatch attribute), 43
 - exists (cassiopeia.Summoner attribute), 44
 - extend() (cassiopeia.ChampionMasteries method), 16
 - extend() (cassiopeia.Champions method), 10
 - extend() (cassiopeia.core.league.LeagueEntries method), 23
 - extend() (cassiopeia.core.league.Leagues method), 21
 - extend() (cassiopeia.core.masterypage.MasteryPages method), 28
 - extend() (cassiopeia.core.match.MatchHistory method), 30
 - extend() (cassiopeia.core.runepage.RunePages method), 40
 - extend() (cassiopeia.FeaturedMatches method), 42
 - extend() (cassiopeia.Items method), 17
 - extend() (cassiopeia.Locales method), 24
 - extend() (cassiopeia.Maps method), 25
 - extend() (cassiopeia.Masteries method), 27
 - extend() (cassiopeia.ProfileIcons method), 37
 - extend() (cassiopeia.Runes method), 39
 - extend() (cassiopeia.SummonerSpells method), 45
 - extend() (cassiopeia.Versions method), 47
- ## F
- FeaturedMatches (class in cassiopeia), 42
 - ferocity (cassiopeia.data.MasteryTree attribute), 49
 - ferocity (cassiopeia.data.Resource attribute), 52
 - filter() (cassiopeia.ChampionMasteries method), 16
 - filter() (cassiopeia.Champions method), 10
 - filter() (cassiopeia.core.league.LeagueEntries method), 23
 - filter() (cassiopeia.core.league.Leagues method), 21
 - filter() (cassiopeia.core.masterypage.MasteryPages method), 28
 - filter() (cassiopeia.core.match.MatchHistory method), 30
 - filter() (cassiopeia.core.runepage.RunePages method), 40
 - filter() (cassiopeia.FeaturedMatches method), 42
 - filter() (cassiopeia.Items method), 18
 - filter() (cassiopeia.Locales method), 24
 - filter() (cassiopeia.Maps method), 25
 - filter() (cassiopeia.Masteries method), 27
 - filter() (cassiopeia.ProfileIcons method), 37
 - filter() (cassiopeia.Runes method), 39
 - filter() (cassiopeia.SummonerSpells method), 45
 - filter() (cassiopeia.Versions method), 47
 - find() (cassiopeia.ChampionMasteries method), 16
 - find() (cassiopeia.Champions method), 10
 - find() (cassiopeia.core.league.LeagueEntries method), 23
 - find() (cassiopeia.core.league.Leagues method), 21
 - find() (cassiopeia.core.masterypage.MasteryPages method), 28
 - find() (cassiopeia.core.match.MatchHistory method), 30
 - find() (cassiopeia.core.runepage.RunePages method), 40
 - find() (cassiopeia.FeaturedMatches method), 42
 - find() (cassiopeia.Items method), 18
 - find() (cassiopeia.Locales method), 24
 - find() (cassiopeia.Maps method), 25
 - find() (cassiopeia.Masteries method), 27
 - find() (cassiopeia.ProfileIcons method), 37
 - find() (cassiopeia.Runes method), 39
 - find() (cassiopeia.SummonerSpells method), 45
 - find() (cassiopeia.Versions method), 47
 - first_baron (cassiopeia.core.match.Team attribute), 31
 - first_blood (cassiopeia.core.match.Team attribute), 32
 - first_blood_assist (cassiopeia.core.match.ParticipantStats attribute), 33
 - first_blood_kill (cassiopeia.core.match.ParticipantStats attribute), 33
 - first_dragon (cassiopeia.core.match.Team attribute), 32
 - first_inhibitor (cassiopeia.core.match.Team attribute), 32
 - first_inhibitor_assist (cassiopeia.core.match.ParticipantStats attribute), 33
 - first_inhibitor_kill (cassiopeia.core.match.ParticipantStats attribute), 33
 - first_rift_herald (cassiopeia.core.match.Team attribute), 32
 - first_tower (cassiopeia.core.match.Team attribute), 32
 - first_tower_assist (cassiopeia.core.match.ParticipantStats attribute), 33
 - first_tower_kill (cassiopeia.core.match.ParticipantStats attribute), 33
 - five (cassiopeia.data.Division attribute), 48
 - fives (cassiopeia.core.league.LeagueEntries attribute), 23
 - fives (cassiopeia.core.league.Leagues attribute), 21
 - flex (cassiopeia.core.league.LeagueEntries attribute), 23
 - flex (cassiopeia.core.league.Leagues attribute), 21
 - flex (cassiopeia.data.Queue attribute), 50
 - flex_threes (cassiopeia.data.Queue attribute), 50
 - flow (cassiopeia.data.Resource attribute), 52
 - four (cassiopeia.data.Division attribute), 48
 - Frame (class in cassiopeia.core.match), 35
 - frame_interval (cassiopeia.core.match.Timeline attribute), 35
 - frames (cassiopeia.core.match.Timeline attribute), 35
 - free_to_play (cassiopeia.Champion attribute), 11
 - fresh_blood (cassiopeia.core.league.LeagueEntry attribute), 24
 - from_data() (cassiopeia.ChampionMasteries method), 16
 - from_data() (cassiopeia.Champions method), 10
 - from_data() (cassiopeia.core.league.LeagueEntries method), 23
 - from_data() (cassiopeia.core.league.Leagues method), 21

`from_data()` (cassiopeia.core.masterypage.MasteryPages method), 28

`from_data()` (cassiopeia.core.match.MatchHistory method), 30

`from_data()` (cassiopeia.core.match.Participant class method), 32

`from_data()` (cassiopeia.core.match.ParticipantStats class method), 33

`from_data()` (cassiopeia.core.match.Team class method), 32

`from_data()` (cassiopeia.core.runepage.RunePages method), 40

`from_data()` (cassiopeia.core.spectator.CurrentMatch class method), 43

`from_data()` (cassiopeia.core.spectator.Participant class method), 43

`from_data()` (cassiopeia.core.spectator.Team class method), 43

`from_data()` (cassiopeia.core.staticdata.champion.ItemSet class method), 14

`from_data()` (cassiopeia.core.staticdata.champion.RecommendedItem class method), 14

`from_data()` (cassiopeia.FeaturedMatches method), 42

`from_data()` (cassiopeia.Items method), 18

`from_data()` (cassiopeia.Locales method), 24

`from_data()` (cassiopeia.Maps method), 25

`from_data()` (cassiopeia.Masteries method), 27

`from_data()` (cassiopeia.ProfileIcons method), 37

`from_data()` (cassiopeia.Runes method), 39

`from_data()` (cassiopeia.SummonerSpells method), 45

`from_data()` (cassiopeia.Versions method), 47

`from_date()` (cassiopeia.patches.Patch class method), 53

`from_id()` (cassiopeia.data.Queue method), 50

`from_id()` (cassiopeia.data.Season method), 52

`from_match_reference()` (cassiopeia.Match class method), 31

`from_str()` (cassiopeia.patches.Patch class method), 53

`fury` (cassiopeia.data.Resource attribute), 52

G

`GameMode` (class in cassiopeia.data), 48

`GameType` (class in cassiopeia.data), 49

`get_challenger_league()` (cassiopeia.cassiopeia method), 21

`get_champion()` (cassiopeia.cassiopeia method), 10

`get_champion_masteries()` (cassiopeia.cassiopeia method), 15

`get_champion_mastery()` (cassiopeia.cassiopeia method), 15

`get_champions()` (cassiopeia.cassiopeia method), 10

`get_current_match()` (cassiopeia.cassiopeia method), 42

`get_featured_matches()` (cassiopeia.cassiopeia method), 42

`get_items()` (cassiopeia.cassiopeia method), 17

`get_language_strings()` (cassiopeia.cassiopeia method), 20

`get_leagues()` (cassiopeia.cassiopeia method), 21

`get_locales()` (cassiopeia.cassiopeia method), 24

`get_maps()` (cassiopeia.cassiopeia method), 25

`get_master_league()` (cassiopeia.cassiopeia method), 21

`get_masteries()` (cassiopeia.cassiopeia method), 26

`get_match()` (cassiopeia.cassiopeia method), 29

`get_match_history()` (cassiopeia.cassiopeia method), 29

`get_profile_icons()` (cassiopeia.cassiopeia method), 36

`get_realms()` (cassiopeia.cassiopeia method), 38

`get_runes()` (cassiopeia.cassiopeia method), 38

`get_status()` (cassiopeia.cassiopeia method), 41

`get_summoner()` (cassiopeia.cassiopeia method), 44

`get_summoner_spells()` (cassiopeia.cassiopeia method), 45

`get_version()` (cassiopeia.cassiopeia method), 47

`get_versions()` (cassiopeia.cassiopeia method), 47

`glyph` (cassiopeia.data.RuneType attribute), 52

`gold` (cassiopeia.data.Tier attribute), 53

`gold_earned` (cassiopeia.Item attribute), 18

`gold_earned` (cassiopeia.core.match.ParticipantStats attribute), 33

`gold_per_min_deltas` (cassiopeia.core.match.ParticipantTimeline attribute), 34

`gold_spent` (cassiopeia.core.match.ParticipantStats attribute), 33

`group` (cassiopeia.Item attribute), 18

H

`health` (cassiopeia.core.staticdata.champion.Stats attribute), 12

`health` (cassiopeia.core.staticdata.item.ItemStats attribute), 19

`health_per_level` (cassiopeia.core.staticdata.champion.Stats attribute), 12

`health_regen` (cassiopeia.core.staticdata.champion.Stats attribute), 12

`health_regen` (cassiopeia.core.staticdata.item.ItemStats attribute), 20

`health_regen_per_level` (cassiopeia.core.staticdata.champion.Stats attribute), 13

`heat` (cassiopeia.data.Resource attribute), 52

`hexakill_summoners_rift` (cassiopeia.data.Queue attribute), 50

`hexakill_twisted_treeline` (cassiopeia.data.Queue attribute), 50

`hide` (cassiopeia.Item attribute), 18

`hostname` (cassiopeia.ShardStatus attribute), 41

`hot_streak` (cassiopeia.core.league.LeagueEntry attribute), 24

I

- `id` (cassiopeia.Champion attribute), 11
- `id` (cassiopeia.core.masterypage.MasteryPage attribute), 29
- `id` (cassiopeia.core.match.Participant attribute), 32
- `id` (cassiopeia.core.match.ParticipantStats attribute), 33
- `id` (cassiopeia.core.match.ParticipantTimeline attribute), 34
- `id` (cassiopeia.core.match.Timeline attribute), 35
- `id` (cassiopeia.core.runepage.RunePage attribute), 41
- `id` (cassiopeia.core.spectator.CurrentMatch attribute), 43
- `id` (cassiopeia.core.staticdata.champion.Skin attribute), 13
- `id` (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), 37
- `id` (cassiopeia.core.summoner.Account attribute), 44
- `id` (cassiopeia.Item attribute), 19
- `id` (cassiopeia.Map attribute), 26
- `id` (cassiopeia.Mastery attribute), 27
- `id` (cassiopeia.Match attribute), 31
- `id` (cassiopeia.Rune attribute), 39
- `id` (cassiopeia.Summoner attribute), 44
- `id` (cassiopeia.SummonerSpell attribute), 46
- `image` (cassiopeia.Champion attribute), 11
- `image` (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), 37
- `image` (cassiopeia.Item attribute), 19
- `image` (cassiopeia.Map attribute), 26
- `image` (cassiopeia.Mastery attribute), 27
- `image` (cassiopeia.Rune attribute), 39
- `image` (cassiopeia.SummonerSpell attribute), 46
- `image_info` (cassiopeia.core.staticdata.champion.Champion attribute), 14
- `image_info` (cassiopeia.core.staticdata.champion.Passive attribute), 13
- `in_store` (cassiopeia.Item attribute), 19
- `included_data` (cassiopeia.Champion attribute), 11
- `included_data` (cassiopeia.Champions attribute), 10
- `included_data` (cassiopeia.Item attribute), 19
- `included_data` (cassiopeia.Items attribute), 18
- `included_data` (cassiopeia.Masteries attribute), 27
- `included_data` (cassiopeia.Mastery attribute), 27
- `included_data` (cassiopeia.Rune attribute), 39
- `included_data` (cassiopeia.Runes attribute), 39
- `included_data` (cassiopeia.SummonerSpell attribute), 46
- `included_data` (cassiopeia.SummonerSpells attribute), 45
- `index()` (cassiopeia.ChampionMasteries method), 16
- `index()` (cassiopeia.Champions method), 10
- `index()` (cassiopeia.core.league.LeagueEntries method), 23
- `index()` (cassiopeia.core.league.Leagues method), 21
- `index()` (cassiopeia.core.masterypage.MasteryPages method), 28
- `index()` (cassiopeia.core.match.MatchHistory method), 30
- `index()` (cassiopeia.core.runepage.RunePages method), 40
- `index()` (cassiopeia.FeatruedMatches method), 42
- `index()` (cassiopeia.Items method), 18
- `index()` (cassiopeia.Locales method), 24
- `index()` (cassiopeia.Maps method), 25
- `index()` (cassiopeia.Masteries method), 27
- `index()` (cassiopeia.ProfileIcons method), 37
- `index()` (cassiopeia.Runes method), 39
- `index()` (cassiopeia.SummonerSpells method), 45
- `index()` (cassiopeia.Versions method), 47
- `info` (cassiopeia.Champion attribute), 11
- `Info` (class in cassiopeia.core.staticdata.champion), 12
- `inhibitor_kills` (cassiopeia.core.match.ParticipantStats attribute), 33
- `inhibitor_kills` (cassiopeia.core.match.Team attribute), 32
- `insert()` (cassiopeia.ChampionMasteries method), 16
- `insert()` (cassiopeia.Champions method), 10
- `insert()` (cassiopeia.core.league.LeagueEntries method), 23
- `insert()` (cassiopeia.core.league.Leagues method), 21
- `insert()` (cassiopeia.core.masterypage.MasteryPages method), 28
- `insert()` (cassiopeia.core.match.MatchHistory method), 30
- `insert()` (cassiopeia.core.runepage.RunePages method), 40
- `insert()` (cassiopeia.FeatruedMatches method), 42
- `insert()` (cassiopeia.Items method), 18
- `insert()` (cassiopeia.Locales method), 24
- `insert()` (cassiopeia.Maps method), 25
- `insert()` (cassiopeia.Masteries method), 27
- `insert()` (cassiopeia.ProfileIcons method), 37
- `insert()` (cassiopeia.Runes method), 39
- `insert()` (cassiopeia.SummonerSpells method), 45
- `insert()` (cassiopeia.Versions method), 47
- `is_bot` (cassiopeia.core.spectator.Participant attribute), 43
- `Item` (class in cassiopeia), 18
- `item_id` (cassiopeia.core.match.Event attribute), 36
- `item_sets` (cassiopeia.core.staticdata.champion.RecommendedItems attribute), 14
- `items` (cassiopeia.core.match.ParticipantStats attribute), 33
- `items` (cassiopeia.core.staticdata.champion.ItemSet attribute), 14
- `Items` (class in cassiopeia), 17
- `ItemSet` (class in cassiopeia.core.staticdata.champion), 14
- `ItemStats` (class in cassiopeia.core.staticdata.item), 19

J

- `japan` (cassiopeia.data.Platform attribute), 49
- `japan` (cassiopeia.data.Region attribute), 51
- `jungle` (cassiopeia.data.Role attribute), 52
- `jungle_minions_killed` (cassiopeia.core.match.ParticipantFrame attribute),

35

K

- key (cassiopeia.Champion attribute), 11
 - key (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 15
 - key (cassiopeia.core.staticdata.champion.SpellVars attribute), 15
 - key (cassiopeia.core.staticdata.summonerspell.SpellVars attribute), 47
 - key (cassiopeia.SummonerSpell attribute), 46
 - keywords (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 15
 - keywords (cassiopeia.Item attribute), 19
 - killer_id (cassiopeia.core.match.Event attribute), 36
 - killing_sprees (cassiopeia.core.match.ParticipantStats attribute), 33
 - kills (cassiopeia.core.match.ParticipantStats attribute), 33
 - korea (cassiopeia.data.Platform attribute), 49
 - korea (cassiopeia.data.Region attribute), 51
- ## L
- lane (cassiopeia.core.match.ParticipantTimeline attribute), 35
 - Lane (class in cassiopeia.data), 49
 - lane_type (cassiopeia.core.match.Event attribute), 36
 - language (cassiopeia.Realms attribute), 38
 - LanguageStrings (class in cassiopeia), 20
 - largest_critical_strike (cassiopeia.core.match.ParticipantStats attribute), 33
 - largest_killing_sprees (cassiopeia.core.match.ParticipantStats attribute), 33
 - largest_multi_kill (cassiopeia.core.match.ParticipantStats attribute), 33
 - last_played (cassiopeia.ChampionMastery attribute), 17
 - latest_data_dragon (cassiopeia.Realms attribute), 38
 - latest_versions (cassiopeia.Realms attribute), 38
 - latin_america_north (cassiopeia.data.Platform attribute), 49
 - latin_america_north (cassiopeia.data.Region attribute), 51
 - latin_america_south (cassiopeia.data.Platform attribute), 49
 - latin_america_south (cassiopeia.data.Region attribute), 51
 - League (class in cassiopeia.core.league), 22
 - league_points (cassiopeia.core.league.LeagueEntry attribute), 24
 - league_positions (cassiopeia.Summoner attribute), 44
 - LeagueEntries (class in cassiopeia.core.league), 23
 - LeagueEntry (class in cassiopeia.core.league), 23
 - leagues (cassiopeia.cassiopeia.Summoner attribute), 21
 - leagues (cassiopeia.Summoner attribute), 44
 - Leagues (class in cassiopeia.core.league), 21
 - legacy_mode (cassiopeia.Realms attribute), 38
 - level (cassiopeia.ChampionMastery attribute), 17
 - level (cassiopeia.core.match.ParticipantFrame attribute), 35
 - level (cassiopeia.Summoner attribute), 44
 - level_up_type (cassiopeia.core.match.Event attribute), 36
 - life_steal (cassiopeia.core.staticdata.item.ItemStats attribute), 20
 - link (cassiopeia.core.staticdata.champion.SpellVars attribute), 15
 - link (cassiopeia.core.staticdata.summonerspell.SpellVars attribute), 47
 - loading_image (cassiopeia.core.staticdata.champion.Skin attribute), 13
 - loading_image_url (cassiopeia.core.staticdata.champion.Skin attribute), 13
 - locale (cassiopeia.Champion attribute), 11
 - locale (cassiopeia.Champions attribute), 10
 - locale (cassiopeia.core.masterypage.MasteryPage attribute), 29
 - locale (cassiopeia.core.runepage.RunePage attribute), 41
 - locale (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), 37
 - locale (cassiopeia.Item attribute), 19
 - locale (cassiopeia.Items attribute), 18
 - locale (cassiopeia.LanguageStrings attribute), 20
 - locale (cassiopeia.Map attribute), 26
 - locale (cassiopeia.Maps attribute), 25
 - locale (cassiopeia.Masteries attribute), 27
 - locale (cassiopeia.Mastery attribute), 27
 - locale (cassiopeia.ProfileIcons attribute), 37
 - locale (cassiopeia.Realms attribute), 38
 - locale (cassiopeia.Rune attribute), 39
 - locale (cassiopeia.Runes attribute), 39
 - locale (cassiopeia.SummonerSpell attribute), 46
 - locale (cassiopeia.SummonerSpells attribute), 45
 - locales (cassiopeia.ShardStatus attribute), 41
 - Locales (class in cassiopeia), 24
 - longest_time_spent_living (cassiopeia.core.match.ParticipantStats attribute), 33
 - lore (cassiopeia.Champion attribute), 11
 - losses (cassiopeia.core.league.LeagueEntry attribute), 24
 - losses (cassiopeia.core.league.Miniseries attribute), 22
- ## M
- magic (cassiopeia.core.staticdata.champion.Info attribute), 12
 - magic_damage_dealt (cassiopeia.core.match.ParticipantStats attribute), 33

- [magic_damage_dealt_to_champions](#) (cassiopeia.core.match.ParticipantStats attribute), [33](#)
[magic_resist](#) (cassiopeia.core.staticdata.champion.Stats attribute), [13](#)
[magic_resist](#) (cassiopeia.core.staticdata.item.ItemStats attribute), [20](#)
[magic_resist_per_level](#) (cassiopeia.core.staticdata.champion.Stats attribute), [13](#)
[magical_damage_taken](#) (cassiopeia.core.match.ParticipantStats attribute), [33](#)
[major](#) (cassiopeia.patches.Patch attribute), [53](#)
[majorminor](#) (cassiopeia.patches.Patch attribute), [53](#)
[mana](#) (cassiopeia.core.staticdata.champion.Stats attribute), [13](#)
[mana](#) (cassiopeia.core.staticdata.item.ItemStats attribute), [20](#)
[mana](#) (cassiopeia.data.Resource attribute), [52](#)
[mana_per_level](#) (cassiopeia.core.staticdata.champion.Stats attribute), [13](#)
[mana_regen](#) (cassiopeia.core.staticdata.champion.Stats attribute), [13](#)
[mana_regen](#) (cassiopeia.core.staticdata.item.ItemStats attribute), [20](#)
[mana_regen_per_level](#) (cassiopeia.core.staticdata.champion.Stats attribute), [13](#)
[map](#) (cassiopeia.core.spectator.CurrentMatch attribute), [43](#)
[map](#) (cassiopeia.core.staticdata.champion.RecommendedItems attribute), [14](#)
[map](#) (cassiopeia.Match attribute), [31](#)
[Map](#) (class in cassiopeia), [26](#)
[maps](#) (cassiopeia.Item attribute), [19](#)
[Maps](#) (class in cassiopeia), [25](#)
[mark](#) (cassiopeia.data.RuneType attribute), [52](#)
[master](#) (cassiopeia.data.Tier attribute), [53](#)
[masteries](#) (cassiopeia.core.masterypage.MasteryPage attribute), [29](#)
[masteries](#) (cassiopeia.core.match.Participant attribute), [32](#)
[masteries](#) (cassiopeia.core.spectator.Participant attribute), [43](#)
[Masteries](#) (class in cassiopeia), [26](#)
[MasterLeague](#) (class in cassiopeia), [22](#)
[Mastery](#) (class in cassiopeia), [27](#)
[mastery_pages](#) (cassiopeia.cassiopeia.Summoner attribute), [26](#)
[mastery_pages](#) (cassiopeia.Summoner attribute), [44](#)
[MasteryPage](#) (class in cassiopeia.core.masterypage), [29](#)
[MasteryPages](#) (class in cassiopeia.core.masterypage), [28](#)
[MasteryTree](#) (class in cassiopeia.data), [49](#)
[Match](#) (class in cassiopeia), [31](#)
[match_history](#) (cassiopeia.cassiopeia.Summoner attribute), [29](#)
[match_history](#) (cassiopeia.Summoner attribute), [44](#)
[match_history_uri](#) (cassiopeia.Summoner attribute), [44](#)
[matched](#) (cassiopeia.data.GameType attribute), [49](#)
[MatchHistory](#) (class in cassiopeia.core.match), [29](#)
[max_profile_icon_id](#) (cassiopeia.Realms attribute), [38](#)
[max_rank](#) (cassiopeia.core.staticdata.champion.ChampionSpell attribute), [15](#)
[max_rank](#) (cassiopeia.SummonerSpell attribute), [46](#)
[max_stacks](#) (cassiopeia.Item attribute), [19](#)
[mid_lane](#) (cassiopeia.data.Lane attribute), [49](#)
[middle](#) (cassiopeia.data.Role attribute), [52](#)
[minions_killed](#) (cassiopeia.core.match.ParticipantFrame attribute), [35](#)
[MiniSeries](#) (class in cassiopeia.core.league), [22](#)
[minor](#) (cassiopeia.patches.Patch attribute), [53](#)
[mode](#) (cassiopeia.core.spectator.CurrentMatch attribute), [43](#)
[mode](#) (cassiopeia.core.staticdata.champion.RecommendedItems attribute), [14](#)
[mode](#) (cassiopeia.Match attribute), [31](#)
[modes](#) (cassiopeia.SummonerSpell attribute), [46](#)
[monster_sub_type](#) (cassiopeia.core.match.Event attribute), [36](#)
[monster_type](#) (cassiopeia.core.match.Event attribute), [36](#)
[movespeed](#) (cassiopeia.core.staticdata.champion.Stats attribute), [13](#)
[movespeed](#) (cassiopeia.core.staticdata.item.ItemStats attribute), [20](#)

N

[name](#) (cassiopeia.ChallengerLeague attribute), [22](#)
[name](#) (cassiopeia.Champion attribute), [11](#)
[name](#) (cassiopeia.core.league.League attribute), [22](#)
[name](#) (cassiopeia.core.league.LeagueEntry attribute), [24](#)
[name](#) (cassiopeia.core.masterypage.MasteryPage attribute), [29](#)
[name](#) (cassiopeia.core.runepage.RunePage attribute), [41](#)
[name](#) (cassiopeia.core.staticdata.champion.ChampionSpell attribute), [15](#)
[name](#) (cassiopeia.core.staticdata.champion.Passive attribute), [13](#)
[name](#) (cassiopeia.core.staticdata.champion.Skin attribute), [13](#)
[name](#) (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), [37](#)
[name](#) (cassiopeia.Item attribute), [19](#)
[name](#) (cassiopeia.Map attribute), [26](#)
[name](#) (cassiopeia.MasterLeague attribute), [22](#)
[name](#) (cassiopeia.Mastery attribute), [27](#)
[name](#) (cassiopeia.patches.Patch attribute), [53](#)
[name](#) (cassiopeia.Rune attribute), [39](#)
[name](#) (cassiopeia.ShardStatus attribute), [41](#)

name (cassiopeia.Summoner attribute), 44
name (cassiopeia.SummonerSpell attribute), 46
nemesis_draft (cassiopeia.data.Queue attribute), 50
neutral_minions_killed (cassiopeia.core.match.ParticipantStats attribute), 33
neutral_minions_killed_enemy_jungle (cassiopeia.core.match.ParticipantStats attribute), 33
neutral_minions_killed_team_jungle (cassiopeia.core.match.ParticipantStats attribute), 33
nexus_siege (cassiopeia.data.GameMode attribute), 48
nexus_siege (cassiopeia.data.Queue attribute), 50
no_cost (cassiopeia.data.Resource attribute), 52
node_capture (cassiopeia.core.match.ParticipantStats attribute), 33
node_capture_assist (cassiopeia.core.match.ParticipantStats attribute), 33
node_neutralize (cassiopeia.core.match.ParticipantStats attribute), 33
node_neutralize_assist (cassiopeia.core.match.ParticipantStats attribute), 33
none (cassiopeia.data.Resource attribute), 52
normal_blind_fives (cassiopeia.data.Queue attribute), 51
normal_blind_threes (cassiopeia.data.Queue attribute), 51
normal_draft_fives (cassiopeia.data.Queue attribute), 51
north_america (cassiopeia.data.Platform attribute), 49
north_america (cassiopeia.data.Region attribute), 51
not_played (cassiopeia.core.league.MiniSeries attribute), 22
number (cassiopeia.core.staticdata.champion.Skin attribute), 13

O

objective_player_score (cassiopeia.core.match.ParticipantStats attribute), 34
observer_key (cassiopeia.core.spectator.CurrentMatch attribute), 43
oceania (cassiopeia.data.Platform attribute), 49
oceania (cassiopeia.data.Region attribute), 51
one (cassiopeia.data.Division attribute), 48
one_for_all (cassiopeia.data.GameMode attribute), 48
one_for_all (cassiopeia.data.Queue attribute), 51
one_for_all_mirror (cassiopeia.data.Queue attribute), 51

P

Participant (class in cassiopeia.core.match), 32
Participant (class in cassiopeia.core.spectator), 43
participant_frames (cassiopeia.core.match.Frame attribute), 35

participant_id (cassiopeia.core.match.Event attribute), 36
participant_id (cassiopeia.core.match.ParticipantFrame attribute), 35
ParticipantFrame (class in cassiopeia.core.match), 35
participants (cassiopeia.core.match.Team attribute), 32
participants (cassiopeia.core.spectator.Team attribute), 43
participants (cassiopeia.Match attribute), 31
ParticipantStats (class in cassiopeia.core.match), 32
ParticipantTimeline (class in cassiopeia.core.match), 34
passive (cassiopeia.Champion attribute), 11
Passive (class in cassiopeia.core.staticdata.champion), 13
Patch (class in cassiopeia.patches), 53
pbe (cassiopeia.data.Platform attribute), 49
pbe (cassiopeia.data.Region attribute), 51
penta_kills (cassiopeia.core.match.ParticipantStats attribute), 34
percent_ability_power (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_armor (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_attack_damage (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_attack_speed (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_attack_speed_per_level (cassiopeia.core.staticdata.champion.Stats attribute), 13
percent_block (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_critical_strike_damage (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_health (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_health_regen (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_magic_resist (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_mana_regen (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_movespeed (cassiopeia.core.staticdata.item.ItemStats attribute), 20
percent_xp_bonus (cassiopeia.core.staticdata.item.ItemStats attribute), 20
physical_damage_dealt (cassiopeia.core.match.ParticipantStats attribute),

- 34
- physical_damage_dealt_to_champions (cassiopeia.core.match.ParticipantStats attribute), 34
- physical_damage_taken (cassiopeia.core.match.ParticipantStats attribute), 34
- plaintext (cassiopeia.Item attribute), 19
- platform (cassiopeia.Champion attribute), 11
- platform (cassiopeia.ChampionMasteries attribute), 16
- platform (cassiopeia.ChampionMastery attribute), 17
- platform (cassiopeia.Champions attribute), 10
- platform (cassiopeia.core.league.League attribute), 22
- platform (cassiopeia.core.league.LeagueEntries attribute), 23
- platform (cassiopeia.core.league.LeagueEntry attribute), 24
- platform (cassiopeia.core.league.Leagues attribute), 21
- platform (cassiopeia.core.masterypage.MasteryPage attribute), 29
- platform (cassiopeia.core.masterypage.MasteryPages attribute), 28
- platform (cassiopeia.core.match.MatchHistory attribute), 30
- platform (cassiopeia.core.match.Timeline attribute), 35
- platform (cassiopeia.core.runepage.RunePage attribute), 41
- platform (cassiopeia.core.runepage.RunePages attribute), 40
- platform (cassiopeia.core.spectator.CurrentMatch attribute), 43
- platform (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), 37
- platform (cassiopeia.data.Region attribute), 51
- platform (cassiopeia.FeaturedMatches attribute), 42
- platform (cassiopeia.Item attribute), 19
- platform (cassiopeia.Items attribute), 18
- platform (cassiopeia.LanguageStrings attribute), 20
- platform (cassiopeia.Locales attribute), 25
- platform (cassiopeia.Map attribute), 26
- platform (cassiopeia.Maps attribute), 25
- platform (cassiopeia.Masteries attribute), 27
- platform (cassiopeia.Mastery attribute), 28
- platform (cassiopeia.Match attribute), 31
- platform (cassiopeia.ProfileIcons attribute), 37
- platform (cassiopeia.Realms attribute), 38
- platform (cassiopeia.Rune attribute), 40
- platform (cassiopeia.Runes attribute), 39
- platform (cassiopeia.ShardStatus attribute), 41
- platform (cassiopeia.Summoner attribute), 44
- platform (cassiopeia.SummonerSpell attribute), 46
- platform (cassiopeia.SummonerSpells attribute), 45
- platform (cassiopeia.Versions attribute), 48
- Platform (class in cassiopeia.data), 49
- platinum (cassiopeia.data.Tier attribute), 53
- point_captured (cassiopeia.core.match.Event attribute), 36
- points (cassiopeia.ChampionMastery attribute), 17
- points (cassiopeia.Mastery attribute), 28
- points_since_last_level (cassiopeia.ChampionMastery attribute), 17
- points_until_next_level (cassiopeia.ChampionMastery attribute), 17
- pop() (cassiopeia.ChampionMasteries method), 16
- pop() (cassiopeia.Champions method), 10
- pop() (cassiopeia.core.league.LeagueEntries method), 23
- pop() (cassiopeia.core.league.Leagues method), 21
- pop() (cassiopeia.core.masterypage.MasteryPages method), 28
- pop() (cassiopeia.core.match.MatchHistory method), 30
- pop() (cassiopeia.core.runepage.RunePages method), 40
- pop() (cassiopeia.FeaturedMatches method), 42
- pop() (cassiopeia.Items method), 18
- pop() (cassiopeia.Locales method), 25
- pop() (cassiopeia.Maps method), 25
- pop() (cassiopeia.Masteries method), 27
- pop() (cassiopeia.ProfileIcons method), 37
- pop() (cassiopeia.Runes method), 39
- pop() (cassiopeia.SummonerSpells method), 45
- pop() (cassiopeia.Versions method), 48
- poro_king (cassiopeia.data.GameMode attribute), 49
- poro_king (cassiopeia.data.Queue attribute), 51
- position (cassiopeia.core.match.Event attribute), 36
- position (cassiopeia.core.match.ParticipantFrame attribute), 35
- Position (class in cassiopeia.core.match), 36
- prerequisite (cassiopeia.Mastery attribute), 28
- preseason_3 (cassiopeia.data.Season attribute), 52
- preseason_4 (cassiopeia.data.Season attribute), 52
- preseason_5 (cassiopeia.data.Season attribute), 52
- preseason_6 (cassiopeia.data.Season attribute), 53
- preseason_7 (cassiopeia.data.Season attribute), 53
- priority (cassiopeia.core.staticdata.champion.RecommendedItems attribute), 14
- profile_icon (cassiopeia.cassiopeia.Summoner attribute), 36
- profile_icon (cassiopeia.Summoner attribute), 44
- ProfileIcon (class in cassiopeia.core.staticdata.profileicon), 37
- ProfileIcons (class in cassiopeia), 36
- progress (cassiopeia.core.league.MiniSeries attribute), 22
- promos (cassiopeia.core.league.LeagueEntry attribute), 24
- property() (cassiopeia.ChampionMasteries method), 16
- property() (cassiopeia.Champions method), 10
- property() (cassiopeia.core.league.LeagueEntries method), 23
- property() (cassiopeia.core.league.Leagues method), 21

`property()` (`cassiopeia.core.masterypage.MasteryPages` method), 28

`property()` (`cassiopeia.core.match.MatchHistory` method), 30

`property()` (`cassiopeia.core.runepage.RunePages` method), 40

`property()` (`cassiopeia.FeaturedMatches` method), 42

`property()` (`cassiopeia.Items` method), 18

`property()` (`cassiopeia.Locales` method), 25

`property()` (`cassiopeia.Maps` method), 25

`property()` (`cassiopeia.Masteries` method), 27

`property()` (`cassiopeia.ProfileIcons` method), 37

`property()` (`cassiopeia.Runes` method), 39

`property()` (`cassiopeia.SummonerSpells` method), 45

`property()` (`cassiopeia.Versions` method), 48

Q

`quadra_kills` (`cassiopeia.core.match.ParticipantStats` attribute), 34

`queue` (`cassiopeia.ChallengerLeague` attribute), 22

`queue` (`cassiopeia.core.league.League` attribute), 22

`queue` (`cassiopeia.core.league.LeagueEntry` attribute), 24

`queue` (`cassiopeia.core.spectator.CurrentMatch` attribute), 43

`queue` (`cassiopeia.MasterLeague` attribute), 22

`queue` (`cassiopeia.Match` attribute), 31

`Queue` (class in `cassiopeia.data`), 50

`queues` (`cassiopeia.core.match.MatchHistory` attribute), 30

`quint` (`cassiopeia.data.RuneType` attribute), 52R

R

`rage` (`cassiopeia.data.Resource` attribute), 52

`range` (`cassiopeia.core.staticdata.champion.ChampionSpell` attribute), 15

`range` (`cassiopeia.SummonerSpell` attribute), 46

`rank_last_season` (`cassiopeia.core.match.Participant` attribute), 32

`ranked_dynamic_queue` (`cassiopeia.data.Queue` attribute), 51

`ranked_enabled` (`cassiopeia.Champion` attribute), 11

`ranked_fives` (`cassiopeia.data.Queue` attribute), 51

`ranked_premade_fives` (`cassiopeia.data.Queue` attribute), 51

`ranked_premade_threes` (`cassiopeia.data.Queue` attribute), 51

`ranked_solo` (`cassiopeia.data.Queue` attribute), 51

`ranked_solo_queue` (`cassiopeia.data.Queue` attribute), 51

`ranked_threes` (`cassiopeia.data.Queue` attribute), 51

`ranks_with` (`cassiopeia.core.staticdata.champion.SpellVars` attribute), 15

`ranks_with` (`cassiopeia.core.staticdata.summonerspell.SpellVars` attribute), 47

`Realms` (class in `cassiopeia`), 38

`rec_math` (`cassiopeia.core.staticdata.champion.ItemSet` attribute), 14

`recommended_itemsets` (`cassiopeia.Champion` attribute), 12

`RecommendedItems` (class in `cassiopeia.core.staticdata.champion`), 13

`red` (`cassiopeia.data.Side` attribute), 53

`red_team` (`cassiopeia.core.spectator.CurrentMatch` attribute), 43

`red_team` (`cassiopeia.Match` attribute), 31

`region` (`cassiopeia.ChallengerLeague` attribute), 22

`region` (`cassiopeia.Champion` attribute), 12

`region` (`cassiopeia.ChampionMasteries` attribute), 16

`region` (`cassiopeia.ChampionMastery` attribute), 17

`region` (`cassiopeia.Champions` attribute), 10

`region` (`cassiopeia.core.league.League` attribute), 22

`region` (`cassiopeia.core.league.LeagueEntries` attribute), 23

`region` (`cassiopeia.core.league.LeagueEntry` attribute), 24

`region` (`cassiopeia.core.league.Leagues` attribute), 21

`region` (`cassiopeia.core.masterypage.MasteryPage` attribute), 29

`region` (`cassiopeia.core.masterypage.MasteryPages` attribute), 29

`region` (`cassiopeia.core.match.MatchHistory` attribute), 30

`region` (`cassiopeia.core.match.Timeline` attribute), 35

`region` (`cassiopeia.core.runepage.RunePage` attribute), 41

`region` (`cassiopeia.core.runepage.RunePages` attribute), 40

`region` (`cassiopeia.core.spectator.CurrentMatch` attribute), 43

`region` (`cassiopeia.core.staticdata.profileicon.ProfileIcon` attribute), 37

`region` (`cassiopeia.data.Platform` attribute), 50

`region` (`cassiopeia.FeaturedMatches` attribute), 42

`region` (`cassiopeia.Item` attribute), 19

`region` (`cassiopeia.Items` attribute), 18

`region` (`cassiopeia.LanguageStrings` attribute), 20

`region` (`cassiopeia.Locales` attribute), 25

`region` (`cassiopeia.Map` attribute), 26

`region` (`cassiopeia.Maps` attribute), 26

`region` (`cassiopeia.Masteries` attribute), 27

`region` (`cassiopeia.MasterLeague` attribute), 22

`region` (`cassiopeia.Mastery` attribute), 28

`region` (`cassiopeia.Match` attribute), 31

`region` (`cassiopeia.ProfileIcons` attribute), 37

`region` (`cassiopeia.Realms` attribute), 38

`region` (`cassiopeia.Rune` attribute), 40

`region` (`cassiopeia.Runes` attribute), 39

`region` (`cassiopeia.ShardStatus` attribute), 41

`region` (`cassiopeia.Summoner` attribute), 44

`region` (`cassiopeia.SummonerSpell` attribute), 46

`region` (`cassiopeia.SummonerSpells` attribute), 45

`region` (`cassiopeia.Versions` attribute), 48

- Region (class in cassiopeia.data), 51
 - region_tag (cassiopeia.ShardStatus attribute), 41
 - remove() (cassiopeia.ChampionMasteries method), 16
 - remove() (cassiopeia.Champions method), 10
 - remove() (cassiopeia.core.league.LeagueEntries method), 23
 - remove() (cassiopeia.core.league.Leagues method), 21
 - remove() (cassiopeia.core.masterypage.MasteryPages method), 29
 - remove() (cassiopeia.core.match.MatchHistory method), 30
 - remove() (cassiopeia.core.runepage.RunePages method), 41
 - remove() (cassiopeia.FeaturedMatches method), 42
 - remove() (cassiopeia.Items method), 18
 - remove() (cassiopeia.Locales method), 25
 - remove() (cassiopeia.Maps method), 26
 - remove() (cassiopeia.Masteries method), 27
 - remove() (cassiopeia.ProfileIcons method), 37
 - remove() (cassiopeia.Runes method), 39
 - remove() (cassiopeia.SummonerSpells method), 45
 - remove() (cassiopeia.Versions method), 48
 - resolve (cassiopeia.data.MasteryTree attribute), 49
 - resource (cassiopeia.Champion attribute), 12
 - resource (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 15
 - resource (cassiopeia.SummonerSpell attribute), 46
 - Resource (class in cassiopeia.data), 51
 - reverse() (cassiopeia.ChampionMasteries method), 16
 - reverse() (cassiopeia.Champions method), 10
 - reverse() (cassiopeia.core.league.LeagueEntries method), 23
 - reverse() (cassiopeia.core.league.Leagues method), 21
 - reverse() (cassiopeia.core.masterypage.MasteryPages method), 29
 - reverse() (cassiopeia.core.match.MatchHistory method), 31
 - reverse() (cassiopeia.core.runepage.RunePages method), 41
 - reverse() (cassiopeia.FeaturedMatches method), 42
 - reverse() (cassiopeia.Items method), 18
 - reverse() (cassiopeia.Locales method), 25
 - reverse() (cassiopeia.Maps method), 26
 - reverse() (cassiopeia.Masteries method), 27
 - reverse() (cassiopeia.ProfileIcons method), 37
 - reverse() (cassiopeia.Runes method), 39
 - reverse() (cassiopeia.SummonerSpells method), 45
 - reverse() (cassiopeia.Versions method), 48
 - revision (cassiopeia.patches.Patch attribute), 53
 - revision_date (cassiopeia.Summoner attribute), 44
 - rift_herald_kills (cassiopeia.core.match.Team attribute), 32
 - role (cassiopeia.core.match.ParticipantTimeline attribute), 35
 - Role (class in cassiopeia.data), 52
 - Rune (class in cassiopeia), 39
 - rune_pages (cassiopeia.cassiopeia.Summoner attribute), 38
 - rune_pages (cassiopeia.Summoner attribute), 44
 - RunePage (class in cassiopeia.core.runepage), 41
 - RunePages (class in cassiopeia.core.runepage), 40
 - runes (cassiopeia.core.match.Participant attribute), 32
 - runes (cassiopeia.core.runepage.RunePage attribute), 41
 - runes (cassiopeia.core.spectator.Participant attribute), 43
 - Runes (class in cassiopeia), 38
 - RuneType (class in cassiopeia.data), 52
 - russia (cassiopeia.data.Platform attribute), 50
 - russia (cassiopeia.data.Region attribute), 51
- ## S
- sanitized_description (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 15
 - sanitized_description (cassiopeia.core.staticdata.champion.Passive attribute), 13
 - sanitized_description (cassiopeia.Item attribute), 19
 - sanitized_description (cassiopeia.Mastery attribute), 28
 - sanitized_description (cassiopeia.Rune attribute), 40
 - sanitized_description (cassiopeia.SummonerSpell attribute), 46
 - sanitized_tooltip (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 15
 - sanitized_tooltip (cassiopeia.SummonerSpell attribute), 46
 - seal (cassiopeia.data.RuneType attribute), 52
 - search() (cassiopeia.ChampionMasteries method), 16
 - search() (cassiopeia.Champions method), 10
 - search() (cassiopeia.core.league.LeagueEntries method), 23
 - search() (cassiopeia.core.league.Leagues method), 22
 - search() (cassiopeia.core.masterypage.MasteryPages method), 29
 - search() (cassiopeia.core.match.MatchHistory method), 31
 - search() (cassiopeia.core.runepage.RunePages method), 41
 - search() (cassiopeia.FeaturedMatches method), 42
 - search() (cassiopeia.Items method), 18
 - search() (cassiopeia.Locales method), 25
 - search() (cassiopeia.Maps method), 26
 - search() (cassiopeia.Masteries method), 27
 - search() (cassiopeia.ProfileIcons method), 37
 - search() (cassiopeia.Runes method), 39
 - search() (cassiopeia.SummonerSpells method), 45
 - search() (cassiopeia.Versions method), 48
 - season (cassiopeia.Match attribute), 31
 - season (cassiopeia.patches.Patch attribute), 53

Season (class in cassiopeia.data), 52
season_3 (cassiopeia.data.Season attribute), 53
season_4 (cassiopeia.data.Season attribute), 53
season_5 (cassiopeia.data.Season attribute), 53
season_6 (cassiopeia.data.Season attribute), 53
season_7 (cassiopeia.data.Season attribute), 53
seasons (cassiopeia.core.match.MatchHistory attribute), 31
services (cassiopeia.ShardStatus attribute), 41
ShardStatus (class in cassiopeia), 41
shield (cassiopeia.data.Resource attribute), 52
showdown (cassiopeia.data.GameMode attribute), 49
showdown_duo (cassiopeia.data.Queue attribute), 51
showdown_solo (cassiopeia.data.Queue attribute), 51
side (cassiopeia.core.match.Participant attribute), 32
side (cassiopeia.core.match.Team attribute), 32
Side (class in cassiopeia.data), 53
sight_wards_bought_in_game (cassiopeia.core.match.ParticipantStats attribute), 34
silver (cassiopeia.data.Tier attribute), 53
skill_slot (cassiopeia.core.match.Event attribute), 36
Skin (class in cassiopeia.core.staticdata.champion), 13
skins (cassiopeia.Champion attribute), 12
slug (cassiopeia.ShardStatus attribute), 41
sort() (cassiopeia.ChampionMasteries method), 16
sort() (cassiopeia.Champions method), 10
sort() (cassiopeia.core.league.LeagueEntries method), 23
sort() (cassiopeia.core.league.Leagues method), 22
sort() (cassiopeia.core.masterypage.MasteryPages method), 29
sort() (cassiopeia.core.match.MatchHistory method), 31
sort() (cassiopeia.core.runepage.RunePages method), 41
sort() (cassiopeia.FeaturedMatches method), 42
sort() (cassiopeia.Items method), 18
sort() (cassiopeia.Locales method), 25
sort() (cassiopeia.Maps method), 26
sort() (cassiopeia.Masteries method), 27
sort() (cassiopeia.ProfileIcons method), 37
sort() (cassiopeia.Runes method), 39
sort() (cassiopeia.SummonerSpells method), 45
sort() (cassiopeia.Versions method), 48
special_recipe (cassiopeia.Item attribute), 19
spell_vamp (cassiopeia.core.staticdata.item.ItemStats attribute), 20
spells (cassiopeia.Champion attribute), 12
SpellVars (class in cassiopeia.core.staticdata.champion), 15
SpellVars (class in cassiopeia.core.staticdata.summonerspell), 47
splash (cassiopeia.core.staticdata.champion.Skin attribute), 13
splash_url (cassiopeia.core.staticdata.champion.Skin attribute), 13
sprite (cassiopeia.Champion attribute), 12
sprite (cassiopeia.Item attribute), 19
sprite (cassiopeia.Map attribute), 26
sprite (cassiopeia.Mastery attribute), 28
sprite (cassiopeia.Rune attribute), 40
sprite (cassiopeia.SummonerSpell attribute), 46
star_guardian (cassiopeia.data.GameMode attribute), 49
start (cassiopeia.patches.Patch attribute), 53
stats (cassiopeia.Champion attribute), 12
stats (cassiopeia.core.match.Participant attribute), 32
stats (cassiopeia.Item attribute), 19
stats (cassiopeia.Rune attribute), 40
Stats (class in cassiopeia.core.staticdata.champion), 12
store (cassiopeia.Realms attribute), 38
strings (cassiopeia.LanguageStrings attribute), 20
summoner (cassiopeia.ChampionMasteries attribute), 16
summoner (cassiopeia.ChampionMastery attribute), 17
summoner (cassiopeia.core.league.LeagueEntry attribute), 24
summoner (cassiopeia.core.masterypage.MasteryPage attribute), 29
summoner (cassiopeia.core.match.MatchHistory attribute), 31
summoner (cassiopeia.core.match.Participant attribute), 32
summoner (cassiopeia.core.runepage.RunePage attribute), 41
summoner (cassiopeia.core.spectator.Participant attribute), 43
Summoner (class in cassiopeia), 44
summoner_spell_d (cassiopeia.core.match.Participant attribute), 32
summoner_spell_d (cassiopeia.core.spectator.Participant attribute), 43
summoner_spell_f (cassiopeia.core.match.Participant attribute), 32
summoner_spell_f (cassiopeia.core.spectator.Participant attribute), 43
SummonerSpell (class in cassiopeia), 45
SummonerSpells (class in cassiopeia), 45
support (cassiopeia.data.Role attribute), 52

T

tags (cassiopeia.Champion attribute), 12
tags (cassiopeia.Item attribute), 19
tags (cassiopeia.Rune attribute), 40
team (cassiopeia.core.match.Participant attribute), 32
team (cassiopeia.core.spectator.Participant attribute), 44
Team (class in cassiopeia.core.match), 31
Team (class in cassiopeia.core.spectator), 43
team_builder (cassiopeia.data.Queue attribute), 51
team_id (cassiopeia.core.match.Event attribute), 36
team_objective (cassiopeia.core.match.ParticipantStats attribute), 34

- team_score (cassiopeia.core.match.ParticipantFrame attribute), 35
- teams (cassiopeia.core.spectator.CurrentMatch attribute), 43
- teams (cassiopeia.Match attribute), 31
- three (cassiopeia.data.Division attribute), 48
- threes (cassiopeia.core.league.LeagueEntries attribute), 23
- threes (cassiopeia.core.league.Leagues attribute), 22
- tier (cassiopeia.ChallengerLeague attribute), 22
- tier (cassiopeia.core.league.League attribute), 22
- tier (cassiopeia.core.league.LeagueEntry attribute), 24
- tier (cassiopeia.Item attribute), 19
- tier (cassiopeia.MasterLeague attribute), 22
- tier (cassiopeia.Rune attribute), 40
- Tier (class in cassiopeia.data), 53
- time_CCing_others (cassiopeia.core.match.ParticipantStats attribute), 34
- timeline (cassiopeia.core.match.Participant attribute), 32
- timeline (cassiopeia.Match attribute), 31
- Timeline (class in cassiopeia.core.match), 35
- timestamp (cassiopeia.core.match.Event attribute), 36
- timestamp (cassiopeia.core.match.Frame attribute), 35
- title (cassiopeia.Champion attribute), 12
- title (cassiopeia.core.staticdata.champion.RecommendedItems attribute), 14
- tooltip (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 15
- tooltip (cassiopeia.SummonerSpell attribute), 46
- top (cassiopeia.data.Role attribute), 52
- top_lane (cassiopeia.data.Lane attribute), 49
- total_damage_dealt (cassiopeia.core.match.ParticipantStats attribute), 34
- total_damage_dealt_to_champions (cassiopeia.core.match.ParticipantStats attribute), 34
- total_damage_taken (cassiopeia.core.match.ParticipantStats attribute), 34
- total_gold (cassiopeia.core.match.ParticipantFrame attribute), 35
- total_heal (cassiopeia.core.match.ParticipantStats attribute), 34
- total_minions_killed (cassiopeia.core.match.ParticipantStats attribute), 34
- total_player_score (cassiopeia.core.match.ParticipantStats attribute), 34
- total_score_rank (cassiopeia.core.match.ParticipantStats attribute), 34
- total_time_crowd_control_dealt (cassiopeia.core.match.ParticipantStats attribute), 34
- total_units_healed (cassiopeia.core.match.ParticipantStats attribute), 34
- tower_kills (cassiopeia.core.match.Team attribute), 32
- tower_type (cassiopeia.core.match.Event attribute), 36
- tree (cassiopeia.Mastery attribute), 28
- triple_kills (cassiopeia.core.match.ParticipantStats attribute), 34
- true_damage_dealt (cassiopeia.core.match.ParticipantStats attribute), 34
- true_damage_dealt_to_champions (cassiopeia.core.match.ParticipantStats attribute), 34
- true_damage_taken (cassiopeia.core.match.ParticipantStats attribute), 34
- turkey (cassiopeia.data.Platform attribute), 50
- turkey (cassiopeia.data.Region attribute), 51
- turret_kills (cassiopeia.core.match.ParticipantStats attribute), 34
- tutorial (cassiopeia.data.GameMode attribute), 49
- tutorial (cassiopeia.data.GameType attribute), 49
- two (cassiopeia.data.Division attribute), 48
- type (cassiopeia.core.match.Event attribute), 36
- type (cassiopeia.core.spectator.CurrentMatch attribute), 43
- type (cassiopeia.core.staticdata.champion.ItemSet attribute), 14
- type (cassiopeia.core.staticdata.champion.RecommendedItems attribute), 14
- type (cassiopeia.LanguageStrings attribute), 20
- type (cassiopeia.Match attribute), 31
- type (cassiopeia.Rune attribute), 40
- ## U
- unpurchasable_items (cassiopeia.Map attribute), 26
- unranked (cassiopeia.data.Tier attribute), 53
- unreal_kills (cassiopeia.core.match.ParticipantStats attribute), 34
- urf (cassiopeia.data.GameMode attribute), 49
- urf (cassiopeia.data.Queue attribute), 51
- url (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), 37
- ## V
- variables (cassiopeia.core.staticdata.champion.ChampionSpell attribute), 15
- variables (cassiopeia.SummonerSpell attribute), 47
- version (cassiopeia.Champion attribute), 12
- version (cassiopeia.Champions attribute), 11
- version (cassiopeia.core.match.Participant attribute), 32

version (cassiopeia.core.staticdata.profileicon.ProfileIcon attribute), [37](#)
version (cassiopeia.Item attribute), [19](#)
version (cassiopeia.Items attribute), [18](#)
version (cassiopeia.LanguageStrings attribute), [20](#)
version (cassiopeia.Map attribute), [26](#)
version (cassiopeia.Maps attribute), [26](#)
version (cassiopeia.Masteries attribute), [27](#)
version (cassiopeia.Mastery attribute), [28](#)
version (cassiopeia.Match attribute), [31](#)
version (cassiopeia.ProfileIcons attribute), [37](#)
version (cassiopeia.Realms attribute), [38](#)
version (cassiopeia.Rune attribute), [40](#)
version (cassiopeia.Runes attribute), [39](#)
version (cassiopeia.SummonerSpell attribute), [47](#)
version (cassiopeia.SummonerSpells attribute), [45](#)
Versions (class in cassiopeia), [47](#)
veteran (cassiopeia.core.league.LeagueEntry attribute), [24](#)
victim_id (cassiopeia.core.match.Event attribute), [36](#)
vilemaw_kills (cassiopeia.core.match.Team attribute), [32](#)
vision_score (cassiopeia.core.match.ParticipantStats attribute), [34](#)
vision_wards_bought_in_game (cassiopeia.core.match.ParticipantStats attribute), [34](#)

W

ward_type (cassiopeia.core.match.Event attribute), [36](#)
wards_killed (cassiopeia.core.match.ParticipantStats attribute), [34](#)
wards_placed (cassiopeia.core.match.ParticipantStats attribute), [34](#)
win (cassiopeia.core.match.ParticipantStats attribute), [34](#)
win (cassiopeia.core.match.Team attribute), [32](#)
wins (cassiopeia.core.league.LeagueEntry attribute), [24](#)
wins (cassiopeia.core.league.MiniSeries attribute), [22](#)
wins_required (cassiopeia.core.league.MiniSeries attribute), [22](#)

X

x (cassiopeia.core.match.Position attribute), [36](#)
xp (cassiopeia.core.match.ParticipantFrame attribute), [35](#)
xp_bonus (cassiopeia.core.staticdata.item.ItemStats attribute), [20](#)
xp_diff_per_min_deltas (cassiopeia.core.match.ParticipantTimeline attribute), [35](#)
xp_per_min_deltas (cassiopeia.core.match.ParticipantTimeline attribute), [35](#)

Y

y (cassiopeia.core.match.Position attribute), [36](#)