compuglobal Documentation

Release 0.1.8

MitchellAW

Apr 25, 2022

CONTENTS

1 Contents: **3**
- 1.1 API Reference . . . 3
- 1.2 Async API Reference . . . 13

2 Installation **23**

3 Basic Usage **25**

4 Async Usage **27**

5 What's New **29**

6 Preview **31**

7 Credits **33**

Index **35**

Python wrapper for the following undocumented APIs:

Frinkiac, Morbotron, Master Of All Science, Capital Beat Us and Good God Lemon

Note: I'll be keeping the wrapper up to date as more APIs are released.

Allows for both random and searchable screencaps, images and gifs embedded with default or custom captions for the following shows:

The Simpsons, Futurama, Rick and Morty, West Wing and 30 Rock.

CHAPTER

ONE

CONTENTS:

1.1 API Reference

CompuGlobal allows for both random and searchable screencaps, images and gifs embedded with default or custom captions for the following shows:

The Simpsons, Futurama, Rick and Morty, West Wing and 30 Rock.

Note: This library relies upon undocumented external APIs.

1.1.1 Contents

CompuGlobalAPI

class compuglobal.**CompuGlobalAPI**(*url*, *title*)

Represents an API Wrapper used for accessing the cghmc API endpoints.

Parameters

- **url** (*str*) – The url of the API.
- **title** (*str*) – The title of the tv show/movie/skit that the url leads to.

Variables

- **random_url** (*str*) – Endpoint used for getting a random screencap.
- **caption_url** (*str*) – Endpoint for getting caption info using episode and timestamp `e = episode & t = timestamp`.
- **search_url** (*str*) – Endpoint for getting screencaps using a search query `q = search query`.
- **frames_url** (*str*) – Endpoint for getting all valid frames before & after an episode and timestamp `episode/timestamp/before/after`.
- **nearby_url** (*str*) – Endpoint for getting all valid frames nearby an episode and timestamp `e = episode & t = timestamp`.
- **episode_url** (*str*) – Endpoint for getting episode info and subtitles from start to end for episode `episode/start/end`.

static encode_caption(*caption*)

Loops through the caption and encodes it in base64 for use in the url.

Parameters **caption** (*str*) – The caption to encode in base64.

Returns The caption encoded in base64.

Return type str

format_caption(*caption*, *max_lines=4*, *max_chars=24*, *shorten=True*)

Loops through the caption and formats it using max_lines and max_chars and returns the formatted outcome.

Parameters

- **caption** (*str*) – The caption to format.
- **max_lines** (*int, optional*) – The maximum number of lines of captions allowed.
- **max_chars** (*int, optional*) – The maximum number of characters allowed per line.
- **shorten** (*bool, optional*) – Whether to shorten the caption at its latest sentence ending.

Returns The formatted caption.

Return type str

generate_gif(*gif_url*)

Performs a GET request using gif_url and returns the direct url for the gif once it has been generated.

Parameters **gif_url** (*str*) – The url of the gif to generate.

Returns The direct url for the generated gif

Return type str

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

get_frames(*episode*, *timestamp*, *before*, *after*)

Performs a GET request to the `api/frames/{episode}/{timestamp}/{before}/{after}` endpoint and gets a list of all valid frames before and after the timestamp of the episode.

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **timestamp** (*int*) – The timestamp of the screencap.
- **before** (*int*) – The number of milliseconds before the timestamp.
- **after** (*int*) – The number of milliseconds after the timestamp.

Returns A list of valid frames before and after the timestamp of the episode, containing the id, episode and timestamp for each frame.

Return type list

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for displaying the valid frames available for the gifmaker.

---

get_nearby_frames(*episode*, *timestamp*)

Performs a GET request to the `api/nearby?e={}&t={}` endpoint and gets a list of seven nearby using the episode key `e={}` and timestamp `t={}` the episode.

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **timestamp** (*int*) – The timestamp of the screencap.

Returns A list of valid frames before and after the timestamp of the episode, containing the id, episode and timestamp for each frame.

Return type list

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for displaying the seven frames in the "More from this scene:" frame selection screen with arrows.

`get_random_screencap()`

Performs a GET request to the `api/random` endpoint and gets a random TV Show screencap.

Returns A random screencap object.

Return type compuglobal.Screencap

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for getting a random screencap when clicking the "RANDOM" button.

`get_screencap(episode=None, timestamp=None, frame=None)`

Performs a GET request to the `api/caption?e={}&t={}` endpoint and gets a TV Show screencap using episode `e={}` and timestamp `t={} or a frame`

Parameters

- **episode** (*str, optional*) – The episode key of the screencap.
- **timestamp** (*int, optional*) – The timestamp of the screencap.
- **frame** (*compuglobal.Frame*) – The frame of the screencap.

Returns A *Screencap* objecct for the episode and timestamp.

Return type compuglobal.Screencap

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for getting the episode info and caption shown below each screencap.

`static json_to_caption(subtitles_json)`

Loops through the subtitles of the json response, concatenates all lines and returns all subtitles combined as a complete caption.

Parameters **subtitles_json** (*dict*) – The json response containing the subtitles of the screencap.

Returns **caption** – The subtitles combined as a complete caption.

Return type str

`search`(*search_text*)

Performs a GET request to the `api/search?q=` endpoint and gets a list of search results using the search text as the search query `q={}` for the request.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns **search_results** – A list of search results containing the id, episode and timestamp for each result.

Return type list

Raises

- *`APIPageStatusError`* – Raises an exception if the status code of the request is not 200.
- *`NoSearchResultsFound`* – Raises an exception if there are no search results found using search_text.

Note: Used for displaying all the search results and their screencaps.

`search_for_screencap`(*search_text*)

Performs a GET request to the `api/search?q=` endpoint using `search()` to get a list of search results using search_text and gets a screencap using the episode and timestamp of the first search result.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns A screencap object of the first search result found using search_text.

Return type compuglobal.Screencap

Raises

- *`APIPageStatusError`* – Raises an exception if the status code of the request is not 200.
- *`NoSearchResultsFound`* – Raises an exception if there are no search results found using search_text.

`static shorten_caption`(*caption*)

Loops through the caption and trims it at its latest sentence ending (., !, ? or ♪).

Parameters **caption** (*str*) – The caption to shorten/trim.

Returns **caption** – The shortened caption, ending at its latest sentence ending.

Return type str

`view_episode`(*episode*, *start*, *end*)

Performs a GET request to the `api/episode/{episode}/{start}/{end}` endpoint and returns the json response containing episode information.

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **start** (*int*) – The starting timestamp for the episode information.
- **end** (*int*) – The ending timestamp for the episode information.

Returns The json response containing the episode information and subtitles for the timestamps.

Return type dict

Note: Used for displaying the rest of an episode when using the "View Episode" button next to each screencap.

Screencap

`class compuglobal.Screencap(api, json: dict)`

Represents a screencap of a TVShow/Movie/Skit generated using an instance of CompuGlobalAPI.

Parameters

- **api** (*CompuGlobalAPI*) – The CompuGlobalAPI object that was used to generate the screencap.
- **json** (*dict*) – The json response from the API for the screencap.

Variables

- **frame** (*Frame*) – The representative frame of the screencap.
- **id** (*int*) – The screencap ID.
- **key** (*str*) – The episode key (S01E01) of the screencap's representative frame.
- **timestamp** (*int*) – The timestamp of the screencap's represenative frame.
- **episode** (*int*) – The episode number of the screencap.
- **season** (*int*) – The season number of the screencap.
- **title** (*str*) – The title of the episode.
- **director** (*str*) – The director(s) of the episode.
- **writer** (*str*) – The writer(s) of the episode.
- **air_date** (*str*) – The original air date of the episode.
- **wiki_url** (*str*) – Url to the wiki of the episode.
- **caption** (*str*) – The caption/subtitles during the screencap.
- **gif_url** (*str*) – The gif url format for the screencap embedded with a caption.
- **mp4_url** (*str*) – The mp4 url format for the screencap embedded with a caption.

`get_real_timestamp()`

Gets a readable timestamp for the frame in format "mm:ss"

Returns A readable timestamp for the frame in format *mm:ss*.

Return type str

`get_image_url()`

Returns the direct image url for the screencap without any caption.

Returns The image url for the screencap without any caption.

Return type str

`get_meme_url`(*caption=None*)

Encodes the caption with base64 and then returns the meme url for the screencap with an embedded caption.

Parameters **caption** (*str, optional*) – The caption to embed in the image, if it is None, it will use the screencaps original caption.

Returns The meme url for the screencap with an embedded caption.

Return type str

`get_gif_url`(*caption=None*, *before=3000*, *after=4000*)

Gets the timestamps of the frames before and after the timestamp of the screencap using the frames endpoint for the screencap's API and returns the url for the gif with an embedded caption.

Parameters

- **caption** (*str, optional*) – The caption to embed in the gif, if it is None, it will use the screencaps original caption.
- **before** (*int, optional*) – The number of milliseconds before the screencap's timestamp to begin the gif, defaults to 3 seconds (3000ms).
- **after** (*int, optional*) – The number of milliseconds after the screencap's timestamp to begin the gif, defaults to 4 seconds (4000ms).

Returns The gif url for the screencap with an embedded caption.

Return type str

Note: Defaults gif duration to ~7 seconds (7000ms).

`get_mp4_url`(*caption=None*, *before=3000*, *after=4000*)

Gets the timestamps of the frames before and after the timestamp of the screencap using the frames endpoint for the screencap's API and returns the url for the mp4 with an embedded caption.

Parameters

- **caption** (*str, optional*) – The caption to embed in the mp4, if it is None, it will use the screencaps original caption.
- **before** (*int, optional*) – The number of milliseconds before the screencap's timestamp to begin the mp4, defaults to 3 seconds (3000ms).
- **after** (*int, optional*) – The number of milliseconds after the screencap's timestamp to begin the mp4, defaults to 4 seconds (4000ms).

Returns The mp4 url for the screencap with an embedded caption.

Return type str

Note: Defaults mp4 duration to ~7 seconds (7000ms).

Frame

class compuglobal.**Frame**(*api*, *frame_json*)

Represents a single frame of a TVShow/Movie/Skit generated using an instance of CompuGlobalAPI.

Parameters

- **api** (*CompuGlobalAPI*) – The CompuGlobalAPI object that was used to generate the screencap.
- **frame_json** (*dict*) – The json response from the API for the screencap.

Variables

- **json** (*dict*) – The json response used to create the frame.
- **id** (*int*) – The ID of the frame.
- **key** (*str*) – The episode key (S01E01) of the frame.
- **timestamp** (*int*) – The timestamp of the frame.
- **image_url** (*str*) – The direct url for the frame image.

get_meme_url(*caption=None*)

Encodes the caption with base64 and then returns the meme url for the frame with an embedded caption.

Parameters **caption** (*str*) – The caption to embed in the image.

Returns The meme url for the frame with an embedded caption.

Return type str

get_real_timestamp()

Gets a readable timestamp for the frame in format "mm:ss"

Returns A readable timestamp for the frame in format *mm:ss*.

Return type str

1.1.2 API Endpoints

class compuglobal.**CompuGlobalAPI**(*url*, *title*)

Represents an API Wrapper used for accessing the cghmc API endpoints.

Parameters

- **url** (*str*) – The url of the API.
- **title** (*str*) – The title of the tv show/movie/skit that the url leads to.

Variables

- **random_url** (*str*) – Endpoint used for getting a random screencap.
- **caption_url** (*str*) – Endpoint for getting caption info using episode and timestamp `e = episode & t = timestamp`.
- **search_url** (*str*) – Endpoint for getting screencaps using a search query `q = search query`.
- **frames_url** (*str*) – Endpoint for getting all valid frames before & after an episode and timestamp `episode/timestamp/before/after`.

- **nearby_url** (*str*) – Endpoint for getting all valid frames nearby an episode and timestamp `e = episode & t = timestamp`.
- **episode_url** (*str*) – Endpoint for getting episode info and subtitles from start to end for episode `episode/start/end`.

get_screencap(*episode=None*, *timestamp=None*, *frame=None*)

Performs a GET request to the `api/caption?e={}&t={}` endpoint and gets a TV Show screencap using episode `e={}` and timestamp `t={} or a frame`

Parameters

- **episode** (*str, optional*) – The episode key of the screencap.
- **timestamp** (*int, optional*) – The timestamp of the screencap.
- **frame** (*compuglobal.Frame*) – The frame of the screencap.

Returns A *Screencap* objecct for the episode and timestamp.

Return type compuglobal.Screencap

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for getting the episode info and caption shown below each screencap.

get_random_screencap()

Performs a GET request to the `api/random` endpoint and gets a random TV Show screencap.

Returns A random screencap object.

Return type compuglobal.Screencap

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for getting a random screencap when clicking the "RANDOM" button.

search(*search_text*)

Performs a GET request to the `api/search?q=` endpoint and gets a list of search results using the search text as the search query `q={}` for the request.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns **search_results** – A list of search results containing the id, episode and timestamp for each result.

Return type list

Raises

- *APIPageStatusError* – Raises an exception if the status code of the request is not 200.
- *NoSearchResultsFound* – Raises an exception if there are no search results found using search_text.

Note: Used for displaying all the search results and their screencaps.

`search_for_screencap`(*search_text*)

Performs a GET request to the `api/search?q=` endpoint using `search()` to get a list of search results using search_text and gets a screencap using the episode and timestamp of the first search result.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns A screencap object of the first search result found using search_text.

Return type compuglobal.Screencap

Raises

- *`APIPageStatusError`* – Raises an exception if the status code of the request is not 200.
- *`NoSearchResultsFound`* – Raises an exception if there are no search results found using search_text.

`get_frames`(*episode*, *timestamp*, *before*, *after*)

Performs a GET request to the `api/frames/{episode}/{timestamp}/{before}/{after}` endpoint and gets a list of all valid frames before and after the timestamp of the episode.

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **timestamp** (*int*) – The timestamp of the screencap.
- **before** (*int*) – The number of milliseconds before the timestamp.
- **after** (*int*) – The number of milliseconds after the timestamp.

Returns A list of valid frames before and after the timestamp of the episode, containing the id, episode and timestamp for each frame.

Return type list

Raises *`APIPageStatusError`* – Raises an exception if the status code of the request is not 200.

Note: Used for displaying the valid frames available for the gifmaker.

`get_nearby_frames`(*episode*, *timestamp*)

Performs a GET request to the `api/nearby?e={}&t={}` endpoint and gets a list of seven nearby using the episode key `e={}` and timestamp `t={}` the episode.

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **timestamp** (*int*) – The timestamp of the screencap.

Returns A list of valid frames before and after the timestamp of the episode, containing the id, episode and timestamp for each frame.

Return type list

Raises *`APIPageStatusError`* – Raises an exception if the status code of the request is not 200.

Note: Used for displaying the seven frames in the "More from this scene:" frame selection screen with arrows.

---

view_episode(*episode*, *start*, *end*)

Performs a GET request to the `api/episode/{episode}/{start}/{end}` endpoint and returns the json response containing episode information.

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **start** (*int*) – The starting timestamp for the episode information.
- **end** (*int*) – The ending timestamp for the episode information.

Returns The json response containing the episode information and subtitles for the timestamps.

Return type dict

Note: Used for displaying the rest of an episode when using the "View Episode" button next to each screencap.

---

generate_gif(*gif_url*)

Performs a GET request using gif_url and returns the direct url for the gif once it has been generated.

Parameters **gif_url** (*str*) – The url of the gif to generate.

Returns The direct url for the generated gif

Return type str

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

1.1.3 Errors

class compuglobal.**APIPageStatusError**(*page_status*, *url*)

Raised when the status code for the API is not 200.

Parameters

- **page_status** (*int*) – The page status number for the error.
- **url** (*str*) – The url page that encountered the error.

class compuglobal.**NoSearchResultsFound**

Raised when no search results are returned during a search query to the search endpoint of the API.

1.1.4 Supported APIs

Frinkiac

class compuglobal.**Frinkiac**

An API Wrapper for accessing Frinkiac API endpoints (The Simpsons).

Morbotron

class compuglobal.**Morbotron**

An API Wrapper for accessing Morbotron API endpoints (Futurama).

Master Of All Science

class compuglobal.**MasterOfAllScience**

An API Wrapper for accessing MasterOfAllScience API endpoints (Rick and Morty).

Good God Lemon

class compuglobal.**GoodGodLemon**

An API Wrapper for accessing GoodGodLemon API endpoints (30 Rock).

Capital Beat Us

class compuglobal.**CapitalBeatUs**

An API Wrapper for accessing CapitalBeatUs API endpoints (West Wing).

FrinkiHams

class compuglobal.**FrinkiHams**

An API Wrapper for accessing FriniHams API endpoints (The Simpsons - Steamed Hams Skit).

1.2 Async API Reference

CompuGlobal allows for both random and searchable screencaps, images and gifs embedded with default or custom captions for the following shows:

The Simpsons, Futurama, Rick and Morty, West Wing and 30 Rock.

Note: This library relies upon undocumented external APIs.

1.2.1 Contents

CompuGlobalAPI

class compuglobal.aio.**CompuGlobalAPI**(*url*, *title*, *timeout*)

Represents an API Wrapper used for accessing the cghmc API endpoints.

Parameters

- **url** (*str*) – The url of the API.
- **title** (*str*) – The title of the tv show/movie/skit that the url leads to.
- **timeout** (*float*) – The timeout for websocket read.

Variables

- **random_url** (*str*) – Endpoint used for getting a random screencap.
- **caption_url** (*str*) – Endpoint for getting caption info using episode and timestamp `e = episode & t = timestamp`.
- **search_url** (*str*) – Endpoint for getting screencaps using a search query `q = search query`.
- **frames_url** (*str*) – Endpoint for getting all valid frames before & after an episode and timestamp `episode/timestamp/before/after`.
- **nearby_url** (*str*) – Endpoint for getting all valid frames nearby an episode and timestamp `e = episode & t = timestamp`.
- **episode_url** (*str*) – Endpoint for getting episode info and subtitles from start to end for episode `episode/start/end`.

static encode_caption(*caption*)

Loops through the caption and formats it using max_lines and max_chars and finally encodes it in base64 for use in the url.

Parameters **caption** (*str*) – The caption to format and encode.

Returns The caption encoded in base64.

Return type str

format_caption(*caption*, *max_lines=4*, *max_chars=24*, *shorten=True*)

Loops through the caption and formats it using max_lines and max_chars and returns the formatted outcome.

Parameters

- **caption** (*str*) – The caption to format.
- **max_lines** (*int, optional*) – The maximum number of lines of captions allowed.
- **max_chars** (*int, optional*) – The maximum number of characters allowed per line.
- **shorten** (*bool, optional*) – Whether to shorten the caption at its latest sentence ending.

Returns The formatted caption.

Return type str

coroutine generate_gif(*gif_url*)

Performs a GET request using gif_url and returns the direct url for the gif once it has been generated.

Parameters **gif_url** (*str*) – The url of the gif to generate.

Returns The direct url for the generated gif

Return type str

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

coroutine get_frames(*episode*, *timestamp*, *before*, *after*)

Performs a GET request to the `api/frames/{episode}/{timestamp}/{before}/{after}` endpoint and gets a list of all valid frames before and after the timestamp of the episode.

Parameters

- **episode** (*str*) – The episode key of the screencap.

- **timestamp** (*int*) – The timestamp of the screencap.
- **before** (*int*) – The number of milliseconds before the timestamp.
- **after** (*int*) – The number of milliseconds after the timestamp.

Returns A list of valid frames before and after the timestamp of the episode, containing the id, episode and timestamp for each frame.

Return type list

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for displaying the valid frames available for the gifmaker.

`coroutine get_random_screencap()`

Performs a GET request to the `api/random` endpoint and gets a random TV Show screencap.

Returns A random screencap object.

Return type compuglobal.Screencap

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for getting a random screencap when clicking the "RANDOM" button.

`coroutine get_screencap(episode=None, timestamp=None, frame=None)`

Performs a GET request to the `api/caption?e={}&t={}` endpoint and gets a TV Show screencap using episode `e={}` and timestamp `t={}`

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **timestamp** (*int*) – The timestamp of the screencap.
- **frame** (*compuglobal.Frame*) – The frame of the screencap.

Returns A *Screencap* objecct for the episode and timestamp.

Return type compuglobal.Screencap

Raises

- *APIPageStatusError* – Raises an exception if the status code of the request is not 200.
- `TypeError` – Raises an exception if the constructor does not receive episode and timestamp, or compuglobal.Frame

Note: Used for getting the episode info and caption shown below each screencap.

`static json_to_caption(subtitles_json)`

Loops through the subtitles of the json response, concatenates all lines and returns all subtitles combined as a complete caption.

Parameters **subtitles_json** (*dict*) – The json response containing the subtitles of the screencap.

Returns **caption** – The subtitles combined as a complete caption.

Return type str

coroutine search(*search_text*)

Performs a GET request to the `api/search?q=` endpoint and gets a list of search results using the search text as the search query `q={}` for the request.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns **search_results** – A list of search results containing the id, episode and timestamp for each result.

Return type list

Raises

- *APIPageStatusError* – Raises an exception if the status code of the request is not 200.
- *NoSearchResultsFound* – Raises an exception if there are no search results found using search_text.

Note: Used for displaying all the search results and their screencaps.

coroutine search_for_screencap(*search_text*)

Performs a GET request to the `api/search?q=` endpoint using `search()` to get a list of search results using search_text and gets a screencap using the episode and timestamp of the first search result.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns A screencap object of the first search result found using search_text.

Return type compuglobal.Screencap

Raises

- *APIPageStatusError* – Raises an exception if the status code of the request is not 200.
- *NoSearchResultsFound* – Raises an exception if there are no search results found using search_text.

static shorten_caption(*caption*)

Loops through the caption and trims it at its latest sentence ending (., !, ? or ♪).

Parameters **caption** (*str*) – The caption to shorten/trim.

Returns **caption** – The shortened caption, ending at its latest sentence ending.

Return type str

Screencap

class compuglobal.aio.**AIOScreencap**(*api*, *json: dict*)

coroutine get_real_timestamp()

Gets a readable timestamp for the frame in format "mm:ss"

Returns A readable timestamp for the frame in format *mm:ss*.

Return type str

coroutine get_image_url()

Returns the direct image url for the screencap without any caption.

Returns The image url for the screencap without any caption.

Return type str

`coroutine get_meme_url(caption=None)`

Encodes the caption with base64 and then returns the meme url for the screencap with an embedded caption.

Parameters **caption** (*str, optional*) – The caption to embed in the image, if it is None, it will use the screencaps original caption.

Returns The meme url for the screencap with an embedded caption.

Return type str

`coroutine get_gif_url(caption=None, before=3000, after=4000)`

Gets the timestamps of the frames before and after the timestamp of the screencap using the frames endpoint for the screencap's API and returns the url for the gif with an embedded caption.

Parameters

- **caption** (*str, optional*) – The caption to embed in the gif, if it is None, it will use the screencaps original caption.
- **before** (*int, optional*) – The number of milliseconds before the screencap's timestamp to begin the gif, defaults to 3 seconds (3000ms).
- **after** (*int, optional*) – The number of milliseconds after the screencap's timestamp to begin the gif, defaults to 4 seconds (4000ms).

Returns The gif url for the screencap with an embedded caption.

Return type str

Note: Defaults gif duration to ~7 seconds (7000ms).

`coroutine get_mp4_url(caption=None, before=3000, after=4000)`

Gets the timestamps of the frames before and after the timestamp of the screencap using the frames endpoint for the screencap's API and returns the url for the mp4 with an embedded caption.

Parameters

- **caption** (*str, optional*) – The caption to embed in the mp4, if it is None, it will use the screencaps original caption.
- **before** (*int, optional*) – The number of milliseconds before the screencap's timestamp to begin the mp4, defaults to 3 seconds (3000ms).
- **after** (*int, optional*) – The number of milliseconds after the screencap's timestamp to begin the mp4, defaults to 4 seconds (4000ms).

Returns The mp4 url for the screencap with an embedded caption.

Return type str

Note: Defaults mp4 duration to ~7 seconds (7000ms).

1.2.2 API Endpoints

class compuglobal.aio.**CompuGlobalAPI**(*url*, *title*, *timeout*)

Represents an API Wrapper used for accessing the cghmc API endpoints.

Parameters

- **url** (*str*) – The url of the API.
- **title** (*str*) – The title of the tv show/movie/skit that the url leads to.
- **timeout** (*float*) – The timeout for websocket read.

Variables

- **random_url** (*str*) – Endpoint used for getting a random screencap.
- **caption_url** (*str*) – Endpoint for getting caption info using episode and timestamp `e = episode & t = timestamp`.
- **search_url** (*str*) – Endpoint for getting screencaps using a search query `q = search query`.
- **frames_url** (*str*) – Endpoint for getting all valid frames before & after an episode and timestamp `episode/timestamp/before/after`.
- **nearby_url** (*str*) – Endpoint for getting all valid frames nearby an episode and timestamp `e = episode & t = timestamp`.
- **episode_url** (*str*) – Endpoint for getting episode info and subtitles from start to end for episode `episode/start/end`.

coroutine get_screencap(*episode=None*, *timestamp=None*, *frame=None*)

Performs a GET request to the `api/caption?e={}&t={}` endpoint and gets a TV Show screencap using episode `e={}` and timestamp `t={}`

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **timestamp** (*int*) – The timestamp of the screencap.
- **frame** (*compuglobal.Frame*) – The frame of the screencap.

Returns A *Screencap* objecct for the episode and timestamp.

Return type compuglobal.Screencap

Raises

- ***APIPageStatusError*** – Raises an exception if the status code of the request is not 200.
- **TypeError** – Raises an exception if the constructor does not receive episode and timestamp, or compuglobal.Frame

---

Note: Used for getting the episode info and caption shown below each screencap.

---

coroutine get_random_screencap()

Performs a GET request to the `api/random` endpoint and gets a random TV Show screencap.

Returns A random screencap object.

Return type compuglobal.Screencap

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for getting a random screencap when clicking the "RANDOM" button.

coroutine search(*search_text*)

Performs a GET request to the `api/search?q=` endpoint and gets a list of search results using the search text as the search query `q={}` for the request.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns **search_results** – A list of search results containing the id, episode and timestamp for each result.

Return type list

Raises

- *APIPageStatusError* – Raises an exception if the status code of the request is not 200.
- *NoSearchResultsFound* – Raises an exception if there are no search results found using search_text.

Note: Used for displaying all the search results and their screencaps.

coroutine search_for_screencap(*search_text*)

Performs a GET request to the `api/search?q=` endpoint using `search()` to get a list of search results using search_text and gets a screencap using the episode and timestamp of the first search result.

Parameters **search_text** (*str*) – The text/quote to search for.

Returns A screencap object of the first search result found using search_text.

Return type compuglobal.Screencap

Raises

- *APIPageStatusError* – Raises an exception if the status code of the request is not 200.
- *NoSearchResultsFound* – Raises an exception if there are no search results found using search_text.

coroutine get_frames(*episode*, *timestamp*, *before*, *after*)

Performs a GET request to the `api/frames/{episode}/{timestamp}/{before}/{after}` endpoint and gets a list of all valid frames before and after the timestamp of the episode.

Parameters

- **episode** (*str*) – The episode key of the screencap.
- **timestamp** (*int*) – The timestamp of the screencap.
- **before** (*int*) – The number of milliseconds before the timestamp.
- **after** (*int*) – The number of milliseconds after the timestamp.

Returns A list of valid frames before and after the timestamp of the episode, containing the id, episode and timestamp for each frame.

Return type list

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

Note: Used for displaying the valid frames available for the gifmaker.

coroutine generate_gif(*gif_url*)

Performs a GET request using gif_url and returns the direct url for the gif once it has been generated.

Parameters **gif_url** (*str*) – The url of the gif to generate.

Returns The direct url for the generated gif

Return type str

Raises *APIPageStatusError* – Raises an exception if the status code of the request is not 200.

1.2.3 Errors

class compuglobal.aio.**APIPageStatusError**(*page_status*, *url*)

Raised when the status code for the API is not 200.

Parameters

- **page_status** (*int*) – The page status number for the error.
- **url** (*str*) – The url page that encountered the error.

class compuglobal.aio.**NoSearchResultsFound**

Raised when no search results are returned during a search query to the search endpoint of the API.

1.2.4 Supported APIs

Frinkiac

class compuglobal.aio.**Frinkiac**(*timeout=15*)

An API Wrapper for accessing Frinkiac API endpoints (The Simpsons).

Morbotron

class compuglobal.aio.**Morbotron**(*timeout=15*)

An API Wrapper for accessing Morbotron API endpoints (Futurama).

Master Of All Science

class compuglobal.aio.**MasterOfAllScience**(*timeout=15*)

An API Wrapper for accessing MasterOfAllScience API endpoints (Rick and Morty).

Good God Lemon

class compuglobal.aio.**GoodGodLemon**(*timeout=15*)

An API Wrapper for accessing GoodGodLemon API endpoints (30 Rock).

Capital Beat Us

class compuglobal.aio.**CapitalBeatUs**(*timeout=15*)

An API Wrapper for accessing CapitalBeatUs API endpoints (West Wing).

FrinkiHams

class compuglobal.aio.**FrinkiHams**(*timeout=15*)

An API Wrapper for accessing FriniHams API endpoints (The Simpsons - Steamed Hams Skit).

CHAPTER

TWO

INSTALLATION

To install the library, you can just run the following command:

```
python3 -m pip install -U compuglobal
```

CHAPTER

THREE

BASIC USAGE

```
import compuglobal

simpsons = compuglobal.Frinkiac()

# Random
screencap = simpsons.get_random_screencap()

# Search
screencap = simpsons.search_for_screencap('nothing at all')

# Images/Gifs
image = screencap.get_meme_url()
gif = screencap.get_gif_url()
```

For documented examples, check here.

CHAPTER

FOUR

ASYNC USAGE

```
import asyncio

import compuglobal


async def main():
    simpsons = compuglobal.Frinkiac()

    # Random
    screencap = await simpsons.get_random_screencap()

    # Search
    screencap = await simpsons.search_for_screencap('nothing at all')

    # Images/Gifs
    image = screencap.get_meme_url()
    gif = await screencap.get_gif_url()

if __name__ == '__main__':
    loop = asyncio.get_event_loop()
    loop.run_until_complete(main())
```

CHAPTER

FIVE

WHAT'S NEW

0.2.7 - Breaking changes

These changes are intended to provide more flexibility in the formatting of captions:

- Added *format_caption()* method to CompuGlobalAPI objects. This replaces the formatting logic previously within the *encode_caption()* method. The *encode_caption()* method now only takes a caption parameter.
- Captions will no longer be formatted and shortened before generation of memes, gifs, and mp4 urls if a caption is given. Behaviour remains the same if no caption is given or if *format_caption()* is used on the caption beforehand.

0.2.1 - Breaking Changes

- Added Frame object: *search()*, *get_frames()* and *get_nearby_frames()* now all return a list of Frame objects instead of the json response.

  Before:

```
search_results = simpsons.search('stupid sexy flanders')
top_result = search_results[0]
screencap = simpsons.get_screencap(top_result['Episode'],
                                   top_result['Timestamp'])
```

  After:

```
search_results = simpsons.search('stupid sexy flanders')
top_result = search_results[0]
screencap = simpsons.get_screencap(top_result.key,
                                   top_result.timestamp)
```

CHAPTER

SIX

PREVIEW

CHAPTER

SEVEN

CREDITS

Creators and contributors of Frinkiac, Morbotron, Master of All Science, Good God Lemon and Capital Beat Us:

- Paul Kehrer
- Sean Schulte
- Allie Young
- Max, Jon Bernhardt, Nick Beatty, Vimp, juz, Iconfactory and Ged Maheux

BLUEamnesiac for the Internet King Popup Banner

Named CompuGlobal as shorthand for CompuGlobalHyperMegaCap, *as the family of websites are named.*

INDEX

A

`APIPageStatusError` (*class in compuglobal*), 12
`APIPageStatusError` (*class in compuglobal.aio*), 20

G

`get_gif_url()` (*compuglobal.aio.AIOScreencap method*), 17
`get_gif_url()` (*compuglobal.Screencap method*), 8
`get_image_url()` (*compuglobal.aio.AIOScreencap method*), 16
`get_image_url()` (*compuglobal.Screencap method*), 7
`get_meme_url()` (*compuglobal.aio.AIOScreencap method*), 17
`get_meme_url()` (*compuglobal.Screencap method*), 7
`get_mp4_url()` (*compuglobal.aio.AIOScreencap method*), 17
`get_mp4_url()` (*compuglobal.Screencap method*), 8
`get_real_timestamp()` (*compuglobal.aio.AIOScreencap method*), 16
`get_real_timestamp()` (*compuglobal.Screencap method*), 7

N

`NoSearchResultsFound` (*class in compuglobal*), 12
`NoSearchResultsFound` (*class in compuglobal.aio*), 20