

Search...

## CGoldstraw's Covid Dashboard documentation

### CGOLDSTRAW'S COVID DASHBOARD DOCUMENTATION

main.py  
covid\_data\_handler.py  
covid\_news\_handling.py  
config\_handler.py  
update\_handler.py

### INDICES AND TABLES

#### Github Repository

#### main.py

Renders recent news articles and COVID data to a dashboard template.

Uses flask to host a dashboard displaying recent COVID data and news articles, with the COVID data location and news search terms decided by the user in the config file. Calls events to be handled such as scheduling of updates, removal of updates, and removal of news articles.

```
covid_dashboard_cgoldstraw.main.index() → str
```

Handles events and renders recent news and COVID data.

Checks URL parameters for events (creating and deleting updates, and blocking news articles), runs the schedulers to check for any scheduled updates, formats pending updates for the website, refreshes COVID data and news articles, and renders the template with the current data.

#### Returns

The html template rendered with the most recent data.

#### covid\_data\_handler.py

Handles requesting and parsing recent COVID data.

Requests and processes recent COVID data from the uk\_covid19 API. Creation and deletion of scheduled updates is also implemented.

```
covid_dashboard_cgoldstraw.covid_data_handler.covid_API_request(location: str = 'Exeter', location_type: str = 'ltla') → dict[str]
```

Requests current COVID data from the Cov19API for a given area.

Uses the Cov19API to request the most recent COVID data for a given area. Returns data as a list of comma separated strings.

#### Parameters

- **location** – The requested COVID data location.
- **location\_type** – The type of area requested (“nation” or “ltla”).

#### Returns

A dictionary containing a csv file containing COVID information for an area, indexed by the area's name.

```
covid_dashboard_cgoldstraw.covid_data_handler.create_covid_update(update_time: str, update_name: str, repeats: str) → None
```

Create and schedule a COVID update for a given time in HH:MM.

Creates a new update with a given name and repeating information, calculates seconds difference between the current time the update time, then schedules an update with that delay.

#### Parameters

- **update\_time** – Time of the update in HH:MM.
- **update\_name** – The identifier of the update.
- **repeats** – Either None or “repeat”.

```
covid_dashboard_cgoldstraw.covid_data_handler.first_non_blank_cell(csv_lines: list[str], column: int) → int
```

Find the first cell with data in a column of a csv file.

Search a column of a csv file for the first non-empty value (excluding the first header row) and return its index. Returns -1 if there is no cell with data.

#### Parameters

- **csv\_lines** – Rows of a csv file in a list.
- **column** – The index of the column to be searched.

#### Returns

The index of the first row with data for the given column, excluding the first header row, or -1 if no data found.

#### Example

```
>>> csv_lines = ["Cats,Dogs,Fish", "1,,3", "2,1,3"]
>>> first_non_blank_cell(csv_lines, 1)
2
```

```
covid_dashboard_cgoldstraw.covid_data_handler.parse_csv_data(csv_filename: str) → list[str]
```

Reads a csv file and returns it as a list of strings.

Reads a csv file in the current directory and returns a list of rows, where each row is a string.

#### Parameters

- **csv\_filename** – The name of the csv file.

#### Returns

A list of strings, each representing a row of the csv file.

```
covid_dashboard_cgoldstraw.covid_data_handler.process_covid_csv_data(covid_csv_data:  
list[str]) → tuple[int, int, int]
```

Summarises a csv file containing COVID data.

Takes a csv file containing COVID data, in the format returned by the uk\_covid19 API, and returns the cases in the last 7 days, the current hospital cases, and the cumulative deaths.

#### Parameters

- **covid\_csv\_data** – A csv file containing COVID data, with
- **deaths** (*cumulative*) –
- **cases** (*hospital*) –
- **columns.** (*and daily cases*) –

#### Returns

A tuple containing the cases in the last 7 days, the current hospital cases, and the total deaths.

```
covid_dashboard_cgoldstraw.covid_data_handler.read_csv_value(csv_row: str, column:  
int) → int
```

Returns an integer value at a given column in a csv row.

#### Parameters

- **csv\_row** – The comma separated string to read.
- **column** – The index of the column to be read.

#### Returns

The integer value in the specified location if a value exists. Returns 0 as a default value if the cell is empty.

Example

```
>>> row = "0,10,20,30,40,50"  
>>> read_csv_value(row, 3)  
30
```

```
covid_dashboard_cgoldstraw.covid_data_handler.run_covid_update(update_name: str)  
→ None
```

Enacts a scheduled update and either repeats or deletes it.

Updates COVID data for the area specified in config.json, and schedules a new update if the update repeats, otherwise the update is removed.

#### Parameters

**update\_name** – The identifier of the update.

```
covid_dashboard_cgoldstraw.covid_data_handler.schedule_covid_updates(update_interval:  
int, update_name: str) → None
```

Schedules a COVID update in a specified number of seconds.

Calls a COVID update in a specified number of seconds, and creates a new update if the identifier is not found.

#### Parameters

- **update\_interval** – The number of seconds until the update.
- **update\_name** – The identifier of the update.

## covid\_news\_handling.py

Handles requesting, filtering and formatting recent news articles.

Uses newsapi.org to request top headlines, and filters them based on search terms defined in the config file. Creation and deletion of scheduled updates is also implemented.

```
covid_dashboard_cgoldstraw.covid_news_handling.block_article(title: str) → None
```

Removes and blocks a news article.

Adds an article title to the blocked articles to hide the article and removes the article from the stored articles.

#### Parameters

- **title** – The title of the article to block.

```
covid_dashboard_cgoldstraw.covid_news_handling.create_news_update(update_time: str, update_name: str, repeats: str) → None
```

Create and schedule a news update for a given time in HH:MM.

Creates a new update with a given name and repeating information, calculates seconds difference between the current time the update time, then schedules an update with that delay.

#### Parameters

- **update\_time** – Time of the update in HH:MM.
- **update\_name** – The identifier of the update.
- **repeats** – Either None or “repeat”.

```
covid_dashboard_cgoldstraw.covid_news_handling.filter_articles(articles: list[dict], search_terms: str) → list[dict]
```

Returns all articles containing a given search term in them.

Takes a list of articles and filters them by their title, only returning articles that contain at least one of the search terms in their title.

#### Parameters

- **articles** – A list of news articles.
- **search\_terms** – A space separated string of search terms.

#### Returns

A list containing only the articles with titles that contained a search term.

```
covid_dashboard_cgoldstraw.covid_news_handling.format_article(article: dict[str] → dict[str])
```

Shortens an article's contents and removes unnecessary data.

Format an article for the data structure used on the website by removing unnecessary data, shortens the contents to 20 words and adds a link to the article in the article's contents.

#### Parameters

**article** – The news article to format.

#### Returns

The new shortened article.

```
covid_dashboard_cgoldstraw.covid_news_handling.news_API_request(covid_terms: str = 'Covid COVID-19 coronavirus' → list[dict])
```

Requests recent filtered top headlines from newsapi.org.

Sends an API request to newsapi.org, filter the articles by the given terms and return the filtered articles.

#### Parameters

**covid\_terms** – Space separated terms to search news articles for.

#### Returns

A list of news articles, with titles containing at least one search term.

```
covid_dashboard_cgoldstraw.covid_news_handling.schedule_news_updates(update_interval: int, update_name: str → None)
```

Schedules a news update in a specified number of seconds.

Calls a news update in a specified number of seconds, and creates a new update if the identifier is not found.

#### Parameters

- **update\_interval** – The number of seconds until the update.
- **update\_name** – The identifier of the update.

```
covid_dashboard_cgoldstraw.covid_news_handling.update_news(update_name: Optional[str] = None → None)
```

Updates a list of relevant articles with the news API.

Gathers and formats recent top headlines, adds new unblocked headlines to the stored articles. If the update was scheduled, the update is either removed or repeated.

#### Parameters

**update\_name** (*Optional*) – The identifier of the update.

config\_handler.py

Handles reading and validating fields in the config file.

```
covid_dashboard_cgoldstraw.config_handler.read_config(*fields: str) → list[str]
```

Reads the config file for given fields.

Open config.json in the current directory and return the data in the fields specified in the arguments.

#### Parameters

**fields** – Variable number of fields to return from config.json.

#### Returns

A list containing the data inside the fields specified by the fields argument, in the order that the fields were given in. Returns None for any fields with the wrong type.

```
covid_dashboard_cgoldstraw.config_handler.validate_field(field: str, value: Any) → Any
```

Type checks a given config field.

Compare a field value's type with the expected type for that field. Return the value if the type matches, otherwise return the default value for that field.

#### Parameters

- **field** – The name of the field.
- **value** – The value stored in that field.

#### Returns

Either the value of the field or the default value of the field, depending on whether the value is of the expected type.

## update\_handler.py

Handles formatting, creating, and deleting updates.

Combines the separate COVID and news update lists, and formats them for the website. Also adds and removes updates from these lists after user events from the dashboard.

```
covid_dashboard_cgoldstraw.update_handler.create_update(sched_time: str) → None
```

Creates new COVID and news updates.

Uses the parameters in the URL to create updates for COVID data and news articles at a given time, with an optional repeating feature.

#### Parameters

**sched\_time** – The time of the update in HH:MM format.

```
covid_dashboard_cgoldstraw.update_handler.format_update(update: dict[str], covid: bool, news: bool) → str
```

Generates a string describing an update for the website.

#### Parameters

- **update** – A dictionary containing update information.
- **covid** – If True, the update affects COVID data.
- **news** – If True, the update affects news articles.

#### Returns

A string containing the time of the update, the data updated, and whether the update repeats or not.

#### Examples

```
>>> update = {"time": "01:23", "repeats": "repeat"}
>>> format_update(update, True, True)
"01:23, Updates Covid & News, Repeats"
```

```
>>> update = {"time": "12:34", "repeats": None}
>>> format_update(update, False, True)
"12:34, Updates News, Doesn't Repeat"
```

```
covid_dashboard_cgoldstraw.update_handler.format_updates() → None
```

Combine the COVID and news updates into a list.

Convert the separate update dictionaries into a formatted list of updates with titles and contents for the website to render.

```
covid_dashboard_cgoldstraw.update_handler.remove_update(name: str) → None
```

Removes COVID and News updates.

Removes an update, specified by its name, by cancelling the next scheduled event and removing the update from the dictionaries.

#### Parameters

**name** – The identifier of the update.

## Indices and tables

- [Index](#)
- [Module Index](#)
- [Search Page](#)