

Portfoliodokumentation

Contents

Module portfolio_app	1
Sub-modules	1
Module portfolio_app.routes	1
Functions	1
Function home	1
Function list	2
Function not_found_error	2
Function show_project	2
Function techniques	2
Module jen_api	3
Functions	3
Function add_project	3
Function get_project	3
Function get_project_count	3
Function get_search_fields	4
Function get_technique_stats	4
Function get_techniques	4
Function load	4
Function main	5
Function remove_project	5
Function search	5
Function write_db_to_json	6

Module portfolio_app

Sub-modules

- [portfolio_app.routes](#)

Module portfolio_app.routes

Functions

Function home

```
def home()
```

Homepage. Contains info on author.

Arguments:

- None

Returns:

- If search has been used, redirect to [list\(\)](#). Otherwise:
- Render home page.

Function list

```
def list()
```

List of projects, with advanced search.

Arguments:

- None

Returns:

- Render list page.

Function not_found_error

```
def not_found_error(
    error
)
```

404 Page. Shows if page no found.

Arguments:

- error (int) - Error code.

Returns:

- If search has been used, redirect to [list\(\)](#). Otherwise:
- Render 404 page. And:
- 404 (int), Error code 404.

Function show_project

```
def show_project(
    id
)
```

Projectpage. Shows info on project with id id.

Arguments:

- id (int) - Id of project to show.

Returns:

- If search has been used, redirect to [list\(\)](#). Otherwise:
- Render 404 page.

Function techniques

```
def techniques()
```

Techniques page. Shows projects based on techniques used

Arguments:

- None

Returns:

- If search has been used, redirect to [list\(\)](#). Otherwise:
- Render techniques page.

Module `jen_api`

Functions

Function `add_project`

```
def add_project(  
    db,  
    project  
)
```

Adds project to database

Parameters:

- `db` (list) - Database, as generated by [load\(\)](#).
- `project` (dictionary) - Project to be added.

Returns:

- None

Function `get_project`

```
def get_project(  
    db,  
    id_  
)
```

Returns project matching `id` in `db`.

Parameters:

- `db` (list) - Database, as generated by [load\(\)](#).
- `id_` - Projects `id`.

Returns:

- (dictionary) - Project containing `project_id`: `id_`.

Function `get_project_count`

```
def get_project_count(  
    db  
)
```

Returns number of projects in `db`.

Parameters:

- `db` (list) - Database, as generated by [load\(\)](#).

Returns:

- (int) - Length of database.

Function get_search_fields

```
def get_search_fields(  
    db  
)
```

Returns available fields to search in db.

Parameters:

- db (list) - Database, as generated by [load\(\)](#).

Returns:

- (list) - List of all keys of dictionaries in db.

Function get_technique_stats

```
def get_technique_stats(  
    db  
)
```

Returns a dictionary where all techniques used are the keys, and the contents are lists of dictionaries of projects IDs and names.

Parameters:

- db (list) - Database, as generated by [load\(\)](#).

Returns:

- (dictionary) - Dictionary containing lists for every technique, which in turn contains dictionaries of id and name for every project using that technique.

Function get_techniques

```
def get_techniques(  
    db  
)
```

Returns available techniques in db.

Parameters:

- db (list) - Database, as generated by [load\(\)](#).

Returns:

- (list) - List of all techniques used in projects in db.

Function load

```
def load(  
    path  
)
```

Loads json and returns list sorted on projects ID

Parameters:

- path (string) - Path to the json file.

Returns:

- (list) - List containing dictionaries of all projects in json file.

Function main

```
def main()
```

Only used for testing purposes.

Parameters:

- None

Returns:

- None

Function remove_project

```
def remove_project(  
    db,  
    id_  
)
```

Removes project from 'db' if id matches project.

Parameters:

- db (list) - Database, as generated by [load\(\)](#).
- id_ - Projects id.

Returns:

- None

Function search

```
def search(  
    db,  
    sort_by='start_date',  
    sort_order='desc',  
    techniques=None,  
    search=None,  
    search_fields=None  
)
```

Will return a list of projects in db matching all of the following search requirements:

1. Only match projects containing all techniques.
2. Only match projects containing string search() in atleast one field in search_fields. If search_fields is empty, all fields will be checked.

Parameters:

- db (list) - Database, as generated by [load\(\)](#).
- sort_by (string) - Field of projects to sort by.
- sort_order (string) - Order to sort in. 'asc' for ascending order, 'desc' for descending.
- techniques (list) - Techniques that must be in project to be returned. Field is ignored if empty or None.
- search() (string) - Free text search.
- search_fields (list) - Fields to be searched. If empty, no fields are searched. If None, all fields are searched.

Returns:

- (list) - A list containing dictionaries for all projects matching given search criteria.

Function write_db_to_json

```
def write_db_to_json(  
    db,  
    save_file  
)
```

Writes db to file with path save_file, overwriting existing data.

Parameters:

- db (list) - Database, as generated by [load\(\)](#).
- save_file - Path to file to write to.

Returns:

- None

Generated by pdoc 0.9.1 (<https://pdoc3.github.io>).