API Documentation

API Documentation

October 12, 2017

Contents

Contents **1**

1 Module myFlaskProject **2**
1.1 Functions 2
1.2 Variables 4

1 Module myFlaskProject

1.1 Functions

website_index()

A route to the website index page named "/" with Flask. The index website displays a maximum of five of the newest projects and the latest projects based on project id are collected from the database in a new list. The database is sorted in a reversed order, since the newest project will have the highest project id. The information gathered are the modified database as well as the information about the author of the portfolio. This returns a Jinja2 rendered template with the gathered information with a Flask built in function.

project_list()

A Route for the project list page named "/list page" with Flask. Saves information about the current search possibilities, current search bar string and other search parameters selected by the user. This returns a Jinja2 rendered template with the gathered information with a Flask built in function.

project_techniques()

A Route for the technique page named "/techniques" with Flask. Information about the selected techniques on the website are gathered in a list, the information about the projects in the database that used the selected techniques are gathered in a list. There is also a list containing all the technique used in all projects, this is mainly used to display all the technique buttons on the website. This returns a Jinja2 rendered template with the gathered information with a Flask built in function.

show_project(*id*=`None`)

A route for a specific project page named "/project/<int:id>" with Flask. This collects all the data for a specific project and collects the name of the project images from the mapstructure. This returns a Jinja2 rendered template with the gathered information with a Flask built in function.

get_data()

Simplified method to load our json database file.

get_project(*id*)

Get a single project by it's id from the default json database.

get_all_techniques()

Returns all techniques as a list of strings from the default json database.

get_all_search_fields()

Returns all search fields that can be selected on the search page.

get_all_sortby_fields()

Returns all fields that can be sorted by on projectlist page

get_selected_techniques_for_search()

Returns all techniques selected on the projectlist page.

get_selected_fields_for_search()

Returns all search fields selected on the projectlist page.

get_selected_sortby_field_for_search()

Returns the selected sort by field on the projectlist page.

get_selected_sortorder_field_for_search()

Returns the selected order for sorting the results on the projectlist page. If no order has been chosen, the default is descending.

get_projects_by_selected_techniques(*selected_techniques*)

Returns all projects from the default json database that used all techiques in the passed list of techniques. If the passed list is None all projects will be returned.

get_author_data()

Returns all data about the author/user of the portfolio.

set_default_values(*db*)

Looks for a valid image type in big_image and small_image in a database, and a valid image type is as long as the type name contains 3 to 4 letters. If there aren't a valid picture for a project a default image is set for the project. Otherwise a picture named after the project ID number is set. This only modifies a copy of the database and returns it.

get_request_args(**arg_names*)

Specific parameters from the GET or POST request can be extracted. The wanted parameter(s) are passed into the function, and the parameters with their values are returned in a dictionary. Accepts any number of string values - one for each key.

form_value_add_item(*form_value*, *item*)

Appends a string to a comma separated GET value string. X,Y -> X,Y,<item>.

form_value_remove_item(*form_value*, *item*)

Removes a string from a comma separated GET value string. X,<item>,Y -> X,Y.

assign_random_wing_pair_to_technique_list(*technique_list*)

Iterates a list of technique dictionaries and adds a set of matching characters to each dictionary with the keys 'left_wing' and 'right_wing'.

randomize_list(*lst*)

Iterates a list and swaps each element with another element at a random position in the list. The passed list is transformed!

1.2 Variables

Name	Description
app	**Value:** `Flask(__name__)`
techniques_decorative_wing_pairs	**Value:** `[('<', '>'), ('(', ')'), ('[', ']'), ('{', '}'), ('/', '/...`

Index

myFlaskProject *(module)*, 2–4
myFlaskProject.assign_random_wing_pair_to_technique_list *(function)*, 3
myFlaskProject.form_value_add_item *(function)*, 3
myFlaskProject.form_value_remove_item *(function)*, 3
myFlaskProject.get_all_search_fields *(function)*, 2
myFlaskProject.get_all_sortby_fields *(function)*, 2
myFlaskProject.get_all_techniques *(function)*, 2
myFlaskProject.get_author_data *(function)*, 3
myFlaskProject.get_data *(function)*, 2
myFlaskProject.get_project *(function)*, 2
myFlaskProject.get_projects_by_selected_techniques *(function)*, 3
myFlaskProject.get_request_args *(function)*, 3
myFlaskProject.get_selected_fields_for_search *(function)*, 3
myFlaskProject.get_selected_sortby_field_for_search *(function)*, 3
myFlaskProject.get_selected_sortorder_field_for_search *(function)*, 3
myFlaskProject.get_selected_techniques_for_search *(function)*, 3
myFlaskProject.project_list *(function)*, 2
myFlaskProject.project_techniques *(function)*, 2
myFlaskProject.randomize_list *(function)*, 4
myFlaskProject.set_default_values *(function)*, 3
myFlaskProject.show_project *(function)*, 2
myFlaskProject.website_index *(function)*, 2