

Data Migration - Final Round

Task 1

Automatically update `index.rst` file when there is a new rst file added.

Requirements

Write a java/python command line application to get the index.rst content and adding new entry for new requirement if the entry is not existed.

Input: The `index.rst` file in GitHub, and the path to the new rst file

Output: The `index.rst` is got updated and pushed to GitHub

Evaluation

- [✓] Read index.rst
- [✓] Update index.rst
- [✓] Push to Github
- [✓] Clean code

Installation

Install the required packages by using `pip`

```
$ pip install -r requirements.txt
```

Getting Started

```
$ python main.py -f docs/src/ECU_Requirement.rst
```

Technical Overview

- We use the GitHub REST API to upload files.
 - **First**, use a GET method to pull the content of the `index.rst` from GitHub.
 - **Second**, map the new rst module_type to correspondent caption, we assume that every module_type is currently map to `System Requirements`
 - **Third**, find the `toctree` section corresponding to the caption, then add the path to the new rst file at the end of the section
 - **Finally**, push the new content of the `index.rst` to GitHub

- Of course, you must provide your personal **access token** and this token must be authorized to access the resources.
- For more detail, please refer to [GitHub REST API documentation](#)

Generative Access Token

- You can following this [guidelines](#) to generate your own **access_token**
- Note that make sure that your **access token** have scope and expiration dates specified enough time and have valid credentials to **read** and **write** the repo

Usage

The program have **1 arguments** for the path to the new rst file since path to the **index.rst** is fixed. You can check the documentation by using the command

```
$ python main.py -h
```

```
usage: main.py [-h] [-f FILE_PATH]
```

optional arguments:

```
-h, --help            show this help message and exit  
-f FILE_PATH, --file_path FILE_PATH  
                        Path to the new rst file
```

Example command:

```
$ python main.py -f docs/src/ECU_Requirement.rst
```

- **Step 1:** Modify the **/Task1/github_config.yaml**

```
AUTHENTICATION:  
  TOKEN: <GitHub-Access-Token>  
  USERNAME: <username>  
MESSAGE: <commit-message>  
REPOSITORY:  
  BRANCH: <name-of-branch>  
  NAME: <name-of-repo>  
  OWNER: <repo-owner>
```

Note that the repository must have the branch before pushing.

Example configuration for a [repository](#):

```
AUTHENTICATION:  
  TOKEN: <GitHub-Access-Token>  
  USERNAME: <username>  
MESSAGE: test-github-access-token  
REPOSITORY:  
  BRANCH: main  
  NAME: Bosch_CodeRace_TheThree  
  OWNER: nlthanhcse
```

- **Step 2:** run the `main.py` to start uploading

```
$ python main.py -f docs/src/ECU_Requirement.rst
```