

NEO DolFin

Updated: 16/11/2023

Branch Management

Authorised Code Owners:

- Shane Dzartov (@SDzartov)
- Jeremy Conway (Jezza12345)
- Aidan John Stuart Wong (wongaid)

GitHub Management

Branch Structure

master

- **Source of Truth**
 - This branch is the release branch, pushes on this branch require 2 codeowner approvals to merge from develop, as it is the final check to contribute to the next release.

develop

- **Source of Truth for all development**
 - This branch is a copy of the most recent release branch, pushes on this branch require 1 codeowner approval to merge from feature, as it is the first check to contribute to the next release.

feature

- **This branch is where you must commit your work**
- Please follow the following naming convention for these branches:
`\feature_{COMPONENT}<INSERT_FILE_OR_FUNCTION_CHANGE>`
 - EXAMPLE: `\feature_FRONT_update_homepage`
- COMPONENT must be equal to == FRONT OR FLASK.
 - FRONT is any change made exclusively to a file in `Templates` or the `layout.py` file.
 - FLASK is any change made to `app.py` or `app.py` + ANY OTHER FILE/DIRECTORY or ANY OTHER FILE/DIRECTORY in the project folder.

{NEW} Rebase from Develop

Rebasing your branch from the develop branch is a way to integrate the latest changes from the develop branch into your own branch while maintaining a linear history.

- THIS IS RECOMMENDED IF THERE ARE COMMITS ON THE DEVELOP WHILE YOU ARE WORKING ON A FLASK branch.

Follow these steps to perform a rebase:

1. Update Your Local Repository: Before you start the rebase, ensure your local repository is up-to-date.

Run the following commands:

```
git checkout develop
git pull origin develop
```

2. Switch to Your Branch: Switch to the branch you want to rebase.

```
git checkout your-branch-name
```

3. Start the Rebase: Initiate the rebase onto the develop branch.

```
git rebase develop
```

4. Resolve Conflicts (if any): If there are any conflicts between your branch's changes and the develop branch's changes, Git will pause the rebase process. Resolve the conflicts in each affected file. After resolving, use the following commands to continue the rebase:

```
git add .           # Add resolved files
git rebase --continue # Continue the rebase
```

5. Push the Rebased Branch: Once the rebase is complete, force-push the rebased branch to the remote repository. Note that force-pushing rewrites the history, so use it with caution.

```
git push origin your-branch-name --force
```

Local Deployment

Requirements:

- IDE, like *Visual Studio Code*
- Python Version == **3.11**
- GitHub Desktop (RECOMMENDED)

Deploy Neo_Dolphin to **Localhost**

- Pull this repo and select this branch, if you are unconfident in your GIT bash skills, please download GitHub Desktop: <https://desktop.github.com/>
- Once you have the repo folder open in your IDE, do the following in the BELOW ORDER:

- Open a terminal window and move into the neo_dolphin directory via the terminal command: `cd <path/to>/GitHub/NEO_Dolphin/neo_dolphin`
 - **(NOTE: INSERT YOUR PATH AS REQUIRED, YOUR PATH MAY DIFFER)**
- Initiate a new `venv` env using the following terminal command: `python -m venv venv`
- Activate the `venv` env using the following terminal command: `venv\scripts\activate`
- Install the required libraries into the `venv` env using the following terminal command: `pip install -r requirements.txt`
- To run the flask application, use the following terminal command: `python app.py`
- Navigate to `127.0.0.8000` in your web browser.

Deploy Dolphin_Analytica to **Localhost**

- Pull this repo and select this branch, if you are unconfident in your GIT bash skills, please download GitHub Desktop: <https://desktop.github.com/>
- Once you have the repo folder open in your IDE, do the following in the BELOW ORDER:
 - Open a terminal window and move into the neo_dolphin directory via the terminal command: `cd <path/to>/GitHub/NEO_Dolphin/dolphin_analytica`
 - **(NOTE: INSERT YOUR PATH AS REQUIRED, YOUR PATH MAY DIFFER)**
 - Initiate a new `venv` env using the following terminal command: `python -m venv venv2`
 - Activate the `venv` env using the following terminal command: `venv\scripts\activate`
 - Install the required libraries into the `venv` env using the following terminal command: `pip install -r requirements.txt`
 - To run the flask application, use the following terminal command: `python app.py`
 - Navigate to `127.0.0.5000` in your web browser.

{NEW} Testing

We have testing scripts enabled for this application.

- Make sure to install the pytest-flask module, it is currently listed in the `requirements.txt`
- To run, follow the steps for local deployment, but instead of the `python app.py` command, please use the `pytest` command.

PLEASE RUN THE PYTEST AND ATTACH THE LOG TEXT/SCREENSHOT TO YOUR PULL REQUEST

Warnings are OKAY, Fails will result in the PULL request not being reviewed.

GCP Deployment

WORK IN PROGRESS