BiteBody Developer’s Guide

BITEBODY

Web Application Dev Guide

Created By: Bryan Rojas, Hector Mendoza
David Ibarra, Malik Coleman.
(Team Late)

Bitebody-flask-api

Team Late's back-end API for bitebody.xyz created with Flask (Python micro-framework).

Created by **Bryan Rojas**, **Hector Mendoza**, **Malik Coleman**, **David Ibarra**

Prerequisites

Specifications & Tools

- 64-bit environment (Trying to make it system agnostic)
- Git

Back-end

- Python 3.7.x
- Pip

Downloading Application

1. Ensure git is installed on computer. You can verify by running the following command on your terminal.

```
git --version
```

2. Clone the GitHub repository.

```
git clone https://github.com/Bite-Body/bitebody-flask-api.git
```

Prerequisites

1. Verify you have Python 3.7.x, open up terminal and run the following command.

```
python -V
```

Running Application locally

1. Change directory (cd) into the project folder.

```
cd bitebody-flask-api
```

2. Install Python dependencies.

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

3. Running the API server.

```
python manage.py
```

Deployment Notes

1. Log into heroku CLI

```
heroku login
```

2. Login into Heroku containers

```
heroku container:login
```

3. Heroku push container

```
heroku container:push web --app gentle-inlet-25364
```

4. Heroku release container

```
heroku container:release web --app gentle-inlet-25364
```

5. Check logs to verify

```
heroku logs --tail
```

Testing API endpoints through Individual Python Files

Before going through with the steps outlined below, please have a local instance of the application running. Documentation for running locally can be found above in the "Running Application Locally" section of the README. :)

Each of the tables stored in the database can have the API's CRUD operations tested through the command line.

A) To test the USER table follow the following instructions:

1. Open the Command Prompt on your machine.
2. Change directory to backend

```
cd bitebody-flask-api
```

3. change directory to test folder (where test files are housed)

```
cd tests
```

4. run user_Test file

```
python user_Test.py
```

B) To test the COLLABORATOR table follow the following instructions:

1. Open the Command Prompt on your machine.
2. Change directory to backend

```
cd bitebody-flask-api
```

3. change directory to test folder (where test files are housed)

```
cd tests
```

4. run user_Test file

```
python collab_Test.py
```

C) To test the YOUTUBE VIDEOS table follow the following instructions:

1. Open the Command Prompt on your machine.
2. Change directory to backend

```
cd bitebody-flask-api
```

3. change directory to test folder (where test files are housed)

```
cd tests
```

4. run youtube_video_test file

```
python youtube_video_Test.py
```

D) To test the MEAL table follow the following instructions:

1. Open the Command Prompt on your machine.
2. Change directory to backend

```
cd bitebody-flask-api
```

3. change directory to test folder (where test files are housed)

```
cd tests
```

4. run meal_test file

```
python meal_Test.py
```

E) To test the WORKOUT table follow the following instructions:

1. Open the Command Prompt on your machine.
2. Change directory to backend

```
cd bitebody-flask-api
```

3. change directory to test folder (where test files are housed)

```
cd tests
```

4. run meal_test file

```
python workout_Test.py
```

Testing API endpoints through A SINGLE Python File

1. Open the Command Prompt on your machine.
2. Change directory to backend

```
cd bitebody-flask-api
```

3. run Test_Tables file

```
python Test_Tables.py
```

4. Once you start the file running, make sure to press any key whenever text stops appearing on screen. This is meant fo you to proceed whenever you're done reading each table's set of tests.

Bitebody - Front-end

CECS 491A - Front-end for BiteBody, created with React.js

Created by **Bryan Rojas**, **Hector Mendoza**, **David Ibarra**, **Malik Coleman**

Prerequisites

- Git
- Node.js

Downloading Application

1. Ensure git is installed on computer. You can verify by running the following command on your terminal.

```
git --version
```

2. Clone the GitHub repository.

```
git clone https://github.com/Bite-Body/bitebody-front-end.git
```

Verify Prerequisites

1. Verify Node.js is installed.

```
npm -version
```

Running Application

1. Change directory (cd) into the project folder.

```
cd bitebody-front-end
```

2. Install Node.js dependencies.

```
npm install
```

3. Running front-end client.

```
npm run start
```

Contributing

1. Ensure the repository is cloned and you can change directory (cd) where it is located.

```
git clone https://github.com/Project-RHR/Front-end.git
cd Front-end
```

1. Pull the latest version from GitHub

```
git pull
```

2. Create a branch for your changes. The name of the branch should be relevant to the work/changes being done.

```
git checkout -b "Bug-Fix"
```

3. Add changes to your branch via git add command. "." character can be replace with a specific file if you don't want to add all changed files.

```
git add .
```

4. Commit changes with a relevant message.

```
git commit -m "Bug-Fix: Fixed issue with login."
```

5. Push changes to GitHub

```
git push origin HEAD
```

6. Create Pull Request

7. Send pull request to CECS 491A Discord Chat

8. Wait for code review and someone else to approve your changes.

9. Merge changes to master branch.

Bitebody Meal Planner Microservice

Team Late's back-end meal planner microservice for bitebody.xyz created via python utilizing AWS Lambda.

Deployment

1. Zip this entire repository.
2. Upload zip file to AWS Lambda - mealplanner

Endpoint

```
https://2o8jsg6z03.execute-api.us-west-1.amazonaws.com/default/mealplanner
```

Sample JSON

```
 {
"request": {
    "diet-type": "Vegan",
    "calories": "2500"
    }
}
```

Environment Variables

- API_USER_KEY
- API_USER_ID
- API_USER_KEY_BACKUP
- API_USER_ID_BACKUP

Bitebody - Admin Panel Front-end

CECS 491A - Admin Panel front-end for BiteBody, created with React.js

Created by **Bryan Rojas**, **Hector Mendoza**, **David Ibarra**, **Malik Coleman**

Prerequisites

- Git
- Node.js

Downloading Application

1. Ensure git is installed on computer. You can verify by running the following command on your terminal.

```
git --version
```

2. Clone the GitHub repository.

```
git clone https://github.com/Bite-Body/bitebody-admin-panel.git
```

Verify Prerequisites

1. Verify Node.js is installed.

```
npm -version
```

Running Application

1. Change directory (cd) into the project folder.

```
cd bitebody-admin-panel
```

2. Install Node.js dependencies.

```
npm install
```

3. Running front-end client.

```
npm run start
```

Contributing

1. Ensure the repository is cloned and you can change directory (cd) where it is located.

```
git clone https://github.com/Bite-Body/bitebody-admin-panel.git
```

1. Pull the latest version from GitHub

```
git pull
```

2. Create a branch for your changes. The name of the branch should be relevant to the work/changes being done.

```
git checkout -b "Bug-Fix"
```

3. Add changes to your branch via git add command. "." character can be replace with a specific file if you don't want to add all changed files.

```
git add .
```

4. Commit changes with a relevant message.

```
git commit -m "Bug-Fix: Fixed issue with login."
```

5. Push changes to GitHub

```
git push origin HEAD
```

6. Create Pull Request

7. Send pull request to CECS 491A Discord Chat

8. Wait for code review and someone else to approve your changes.

9. Merge changes to master branch.