REST API

LANDSTOCK-TKPROJECT

◦ Developed with the software and tools below.

Table of Contents

Overview

- **Framework:** The project is built on the Fastify framework, known for its high performance and low overhead.
- **Database:** We use the powerful ClickHouse database to efficiently store and manage real estate data.
- **Domain:** Visit our domain at https://b.thienkhoi.com/mbls to access the API's endpoints and explore the wealth of information available.

API Documentation

Explore the API's endpoints and discover how to use it by checking out our detailed Swagger documentation.

API Routes

▶ Health Check

Description: This route handles health checks for the API.

- **Get the health of the API:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls`
  - **Method:** `GET`
  - **Description:** Returns the health of the API.

▶ BDS

Description: This route handles interactions related to real estate properties.

- **Get all the real estates (condition support):**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/bds`
  - **Method:** `GET`
  - **Description:** Returns a list of all available real estate properties.
  - **Authentication:** JWT is required to access this endpoint.
- **Get a real estate by sMa:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/bds/sMa`
  - **Method:** `GET`
  - **Query Parameter:** `sMa`
  - **Description:** Returns a real estate property that matches the provided sMa.
- **Create a new real estate:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/bds`
  - **Method:** `POST`
  - **Description:** Creates a new real estate property.
- **Delete a real estate by sMa:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/bds/sMa`
  - **Method:** `DELETE`
  - **Description:** Deletes a real estate property that matches the provided sMa.

More route details and documentation can be found in the provided link to the source code.

▶ Func

Description: This route handles interactions related to the functions of the API.

- **Get the results of the real estates after applying aggregate functions (condition support):**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/func/bds`
  - **Method:** `GET`
  - **Description:** Returns the results of the real estates after applying aggregate functions.
  - **Authentication:** JWT is required to access this endpoint.

More route details and documentation can be found in the provided link to the source code.

▶ List

Description: This route contains sub-routes that list all properties of a certain type.

- **Get all the cities:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/list/tinh`
  - **Method:** `GET`
  - **Description:** Returns a list of all available cities.
  - **Authentication:** JWT is required to access this endpoint.
- **Get all the districts:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/list/quan`
  - **Method:** `GET`
  - **Description:** Returns a list of all available districts.
  - **Authentication:** JWT is required to access this endpoint.
- **Get all the wards:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/list/phuongxa`
  - **Method:** `GET`
  - **Description:** Returns a list of all available wards.
  - **Authentication:** JWT is required to access this endpoint.
- **Get all the directions:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/list/huongnha`
  - **Method:** `GET`
  - **Description:** Returns a list of all available directions.
  - **Authentication:** JWT is required to access this endpoint.
- **Get all the sections:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/list/loaihang`
  - **Method:** `GET`
  - **Description:** Returns a list of all available sections.
  - **Authentication:** JWT is required to access this endpoint.

More route details and documentation can be found in the provided link to the source code.

▶ Customers

Description: This route handles interactions related to customer properties.

- **Get all the customers:**
  - **Endpoint:** `https://b.thienkhoi.com/mbls/kh`
  - **Method:** `GET`
  - **Description:** Returns a list of all available customers.
  - **Authentication:** JWT is required to access this endpoint.
- **Get a customer by sMa:**

    - **Endpoint:** `https://b.thienkhoi.com/mbls/kh/sMa`
    - **Method:** `GET`
    - **Query Parameter:** `sMa`
    - **Description:** Returns a customer that matches the provided sMa.
    - **Authentication:** JWT is required to access this endpoint.

- **Create a new customer:**

    - **Endpoint:** `https://b.thienkhoi.com/mbls/kh`
    - **Method:** `POST`
    - **Description:** Creates a new customer.
    - **Authentication:** JWT is required to access this endpoint.

- **Delete a customer by sMa:**

    - **Endpoint:** `https://b.thienkhoi.com/mbls/kh/sMa`
    - **Method:** `DELETE`
    - **Description:** Deletes a customer that matches the provided sMa.

Note: GET routes query parameters contain the following properties:

- `limit`: The maximum number of results to return.
- `offset`: The number of results to skip before returning the first result.
- `_sort`: The column to sort the results by.

More route details and documentation can be found in the provided link to the source code.

⚠️ **Notice:**

- All GET routes and POST requests to `https://b.thienkhoi.com/mbls/kh` in this API require JWT authentication.

- For POST routes, ensure the request body is formatted as an array of JSON objects.

API Models

- BDS models can be found here
- Image models can be found here
- City models can be found here
- District models can be found here
- Ward models can be found here
- Direction models can be found here
- Section models can be found here
- Customer models can be found here

Database Migration

Follow the steps below to manage database operations seamlessly:

Migration Script Files

Find all the necessary migration script files in the scripts directory.

Generate Table Creation Queries

To generate SQL queries for table creation in the database, utilize the generate_ctb_queries_file.sh script. This script outputs .sql files containing the necessary queries.

```
./scripts/generate_ctb_queries_file.py
```

Run Table Creation Queries

Execute the generated queries to create tables in the database by using the run_ctb_queries_file.sh script.

```
./scripts/run_ctb_queries_file.py
```

Insert CSV Data

For populating the database with CSV data, leverage the insert_csv_data.sh script. This script streamlines the process of inserting data into the created tables.

```
./scripts/insert_csv_data.py
```

Drop All Tables

To remove all tables from the database, utilize the drop_tables.sh script. This script aids in the clean-up and removal of existing tables.

```
./scripts/drop_tables.py
```

Feel free to explore and adapt these scripts according to your specific requirements for the API's database management.

Plugins

The following npm packages and Fastify plugins are used in this project to enhance the functionality and security of the API:

- **@clickhouse/client**: A ClickHouse client for connecting and querying a ClickHouse database.
- **@fastify/autoload**: A Fastify plugin for auto-loading routes, schemas, and plugins.

- **@fastify/cors**: A Fastify plugin for handling Cross-Origin Resource Sharing (CORS) to allow or restrict cross-origin requests.
- **@fastify/helmet**: A Fastify plugin for securing your application by setting various HTTP headers.
- **@fastify/rate-limit**: A Fastify plugin for implementing rate limiting to control the number of requests from clients.
- **@fastify/sensible**: A Fastify plugin that provides a set of common utility functions and decorators.
- **@fastify/swagger**: A Fastify plugin that generates OpenAPI documentation for your API.
- **@fastify/swagger-ui**: A Fastify plugin that serves the Swagger UI for viewing and interacting with your API's documentation.
- **@fastify/jwt**: A Fastify plugin for implementing JSON Web Tokens (JWT) for authentication and authorization.
- **fastify**: The core Fastify framework, used for building web applications.
- **fastify-cli**: A Fastify command-line tool for generating project templates and scaffolding.
- **fastify-plugin**: A utility for creating Fastify plugins with ease.
- **moment**: A popular library for parsing, validating, manipulating, and formatting dates and times in JavaScript.

Make sure to install these dependencies using npm or yarn before running your Fastify application. You can find more details on how to use these plugins in the project's documentation or by referring to their respective npm package pages.

Getting Started

Dependencies

Please ensure you have the following dependencies installed on your system:

- Node.js (v14.17.0 or higher)
- npm (v6.14.13 or higher)
- fastify (v3.20.1 or higher)
- ClickHouse (v21.3.10.1 or higher)

Installation

1. Clone the landstock-tkproject repository:

```
git clone https://github.com/minhtran241/landstock-tkproject
```

2. Change to the project directory:

```
cd landstock-tkproject
```

3. Install the dependencies:

```
npm install
```

Environment Variables

- Create a `.env` file in the root directory of the project.
- Add the listed environment variables in the .env.example file to the `.env` file.
- Replace the values of the environment variables with your own values.

Running landstock-tkproject

```
npm run start
```

Tests

```
npm test
```

Contributing

Contributions are always welcome! Please follow these steps:

1. Fork the project repository. This creates a copy of the project on your account that you can modify without affecting the original project.
2. Clone the forked repository to your local machine using a Git client like Git or GitHub Desktop.
3. Create a new branch with a descriptive name (e.g., `new-feature-branch` or `bugfix-issue-123`).

```
git checkout -b new-feature-branch
```

4. Make changes to the project's codebase.
5. Commit your changes to your local branch with a clear commit message that explains the changes you've made.

```
git commit -m 'Implemented new feature.'
```

6. Push your changes to your forked repository on GitHub using the following command

```
git push origin new-feature-branch
```

7. Create a new pull request to the original project repository. In the pull request, describe the changes you've made and why they're necessary. The project maintainers will review your changes and provide feedback or merge them into the main branch.

License

Acknowledgments

- Fastify
- ClickHouse
- Swagger
- Node.js
- npm
- Moment.js

↑ Return
