Spending Tracker

\*Добавить описание\*

Данный проект состоит из следующих модулей:

* Клиентская часть приложения

Стэк: vue.js \*продолжить список\*

Включает в себя такие страницы:

* + основная страница (-)
  + страница просмотра полной статистики (-)
  + Page Not Found (-)
  + форма добавления пункта траты (-)
  + форма авторизации (-)
  + форма регистрации (-)
* Серверная часть – модуль работы с информацией о тратах:

Стэк: Golang, viper (для конфигурации), test

API включает в себя такие запросы:

* Добавить пункт траты: \*дополнить\*

Тип запроса: POST

\*Дополнить\*

* \*Продолжить список\*
* Серверная часть – модуль авторизации:

Стэк: Golang, viper (для конфигурации), test

API включает в себя такие запросы:

* Зарегистрировать пользователя:

Тип запроса: POST

Входная информация (JSON):

Тело запроса: значения с формы регистрации, объединенные в объект JSON (имя пользователя и пароль)

Возвращаемые значения (JSON):

Статус 200 – Пользователь успешно зарегистрирован

Статус 400 – Пользователь с таким именем уже существует

Примечание: Данные на сервер гарантированно приходят валидными (длина имени пользователя и пароля), т.к.предварительная проверка происходит на стороне клиента.

* Получить информацию о пользователе:

Тип запроса: GET

Входная информация:

Заголовок запроса (header): JWT

Возвращаемые значения (JSON):

Статус 200 – Возвращается объект, который включает статус, информацию о пользователе (имя пользователя и роль. **Пароль возвращать ЗАПРЕЩЕНО!!!!**)

Статус 401 – Пользователь не авторизован (JWT не валиден или отсутствуем). Возвращается объект, который включает статус и сообщение.

\*При появлении новой информации документация будет дополняться\*

Use cases:

\*Расписать\*