Habit App Backend

Django, SQLite

Table of Contents

Framework Selection

This project is developed using Django Web Framework for following purposes

- Django is a Python framework that makes it easier to create web sites using Python.
- Django was created with the goal of creating a framework that would allow developers to build web applications in less time.

SQLite is used as it is transactional, self-contained and needs minimal configuration.

Stages of work

- Design Database Structure
- Design entity models and develop class relationships.
- Document the business logic.

- Develop endpoints using Django framework.
- Store records in SQLite database using SQLite3 library.

- Create test fixtures and perform Unit Testing using the same.
- Find for any bugs and debug if any.

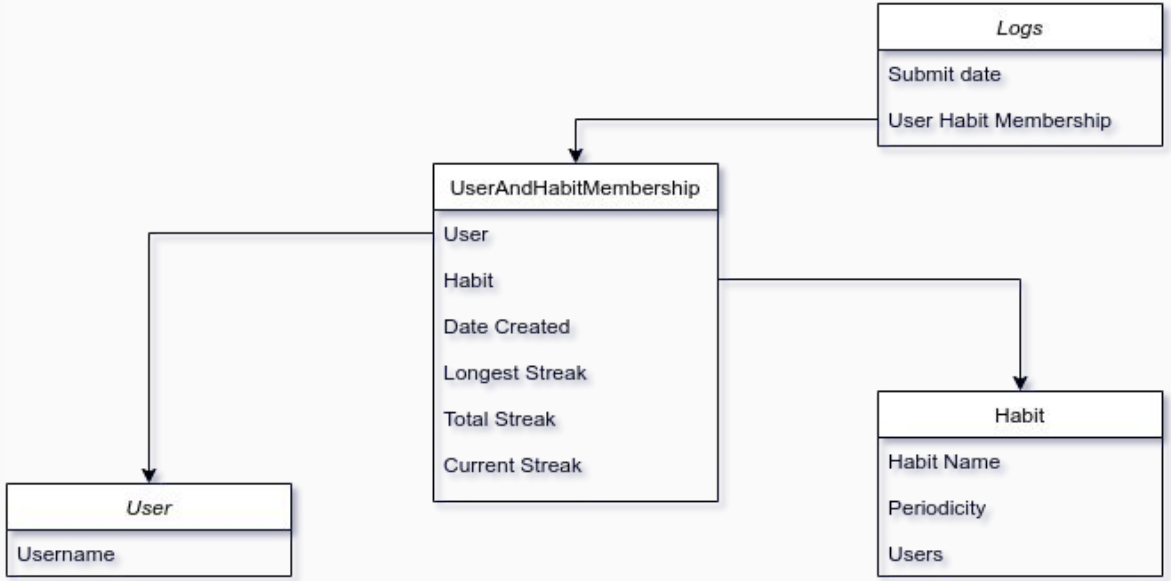
Business Logic

- A user registers to the project using an unique username.
- User can register to any pre-existing habit or they can create their own new habit. Habit contains information like habit name, periodicity and users linked to the habit.
- Users can track their habits streak as well as check the habit rating.
- All records are stored as a log as soon as any task in completed.

Class Diagram

The different entities in our project are as follows:

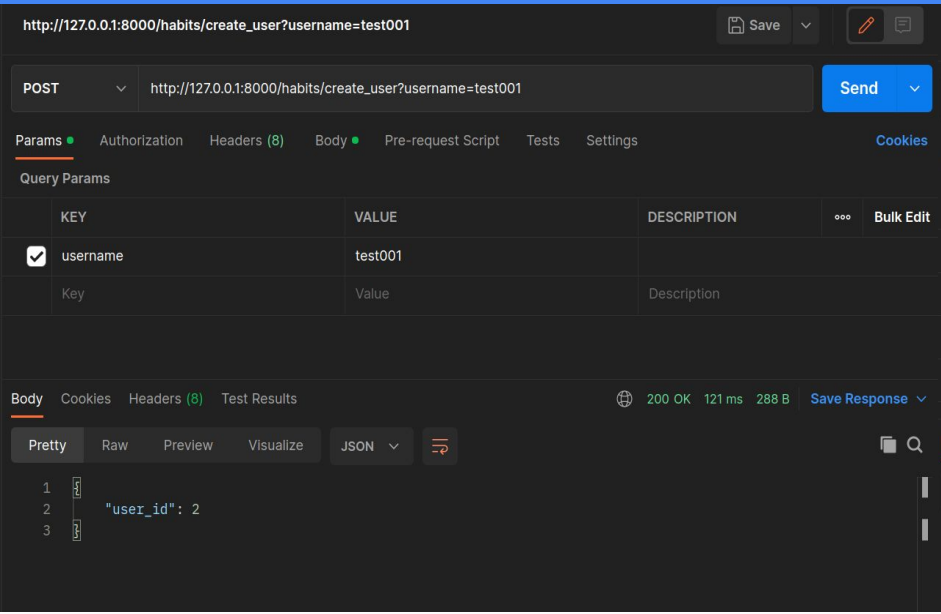
- User
- Habit
- UserAndHabitMembership
- Logs

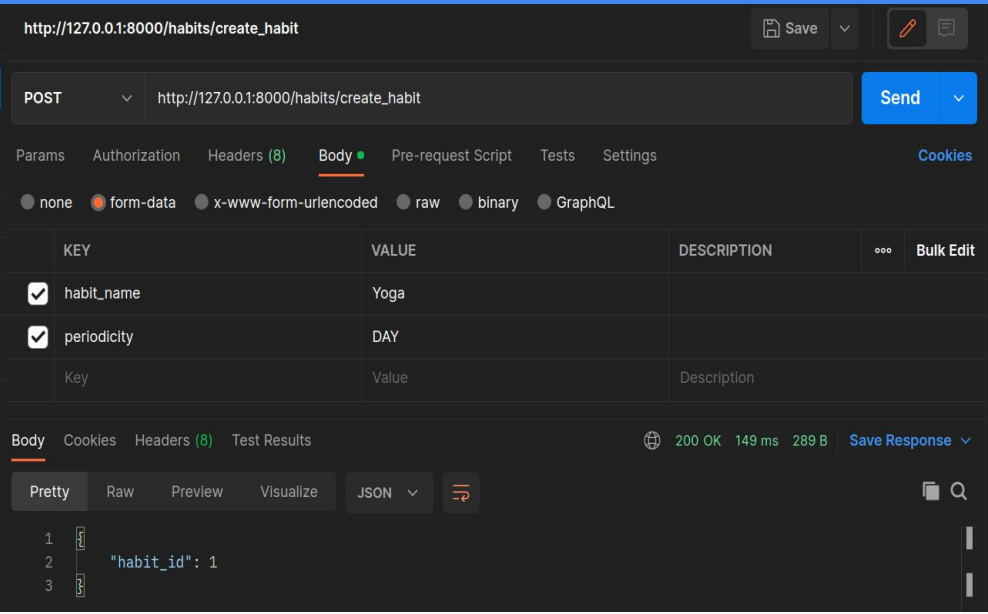


Testing

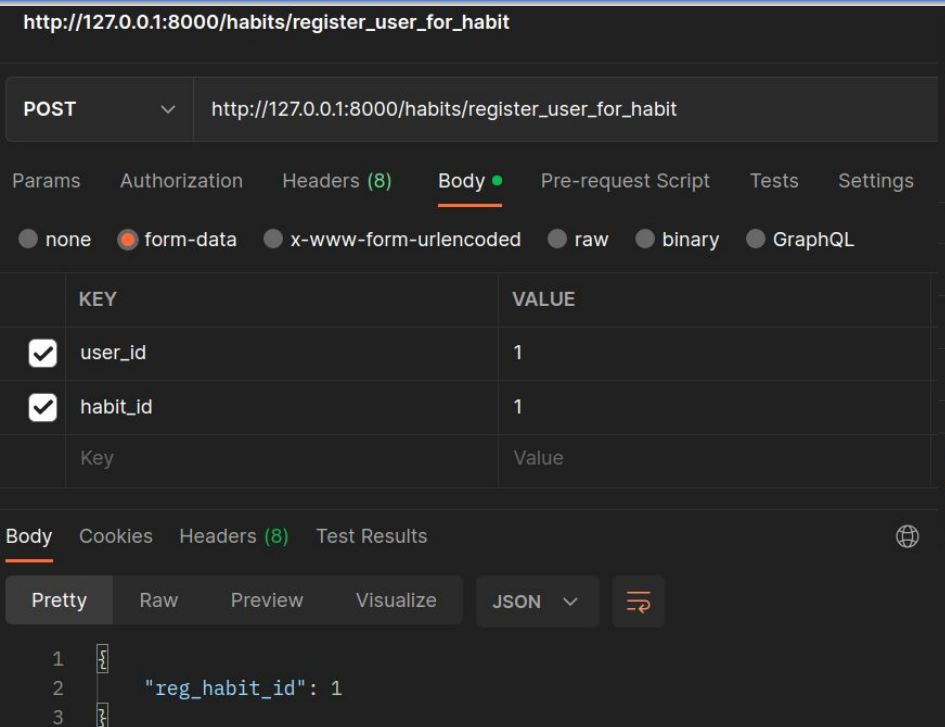
```
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Habit Creation Done
User Creation Done
Add Habit to User Done
Checking off by user done
Checking off by user done
Checking off by user done
get_user Done
get_habit_score Done
Fixture Testing Completed
.
----------------------------------------------------------------------
Ran 1 test in 0.100s

OK
Destroying test database for alias 'default'...
```

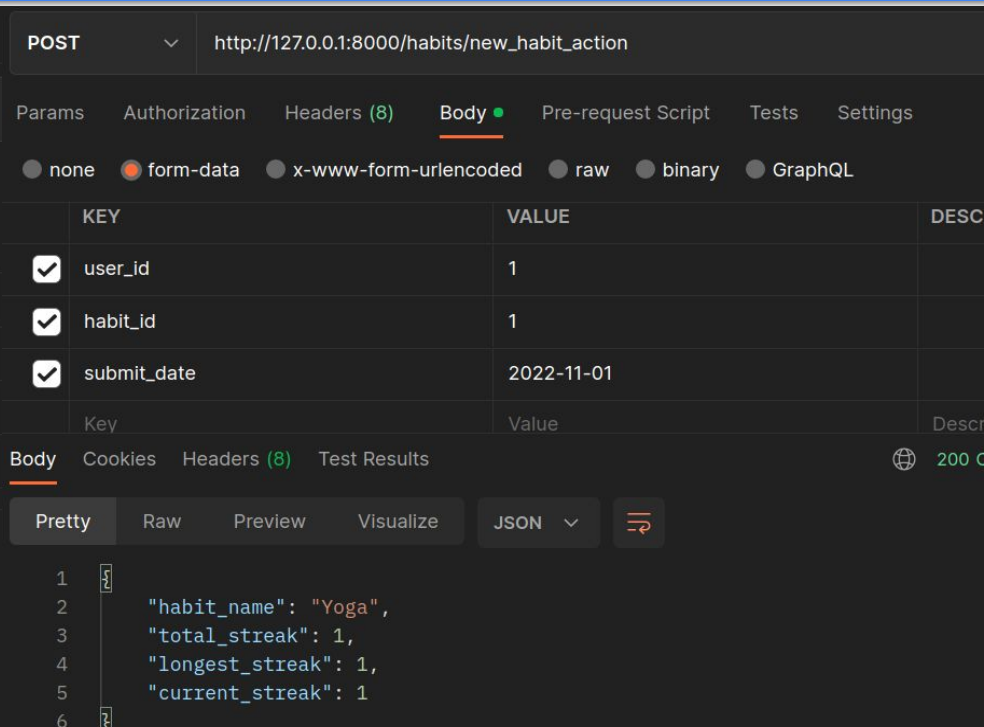
Postman Outputs





Postman Outputs





Conclusion

- The project is developed using Django and SQLite libraries.
- Future scope includes visualisation feature for analytics module, making the login process more secure and so on.
- Django applications like these can be deployed on the internet very easily.

Thanks!

- Ilya Vasiliev