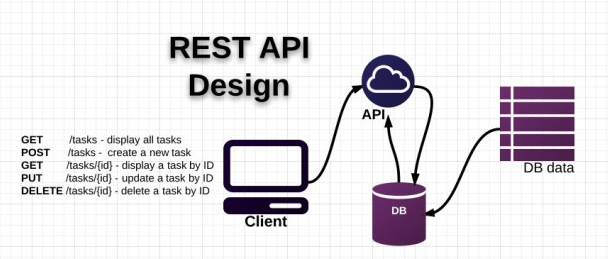
RESTful API

REST (REpresentational State Transfer) is an architectural style, and an approach to communications that is often used in the development of Web services. The use of REST is often preferred over the more heavyweight SOAP (Simple Object Access Protocol) style because REST does not leverage as much bandwidth, which makes it a better fit for use over the Internet. In simple three lines REST API is a:

1. A way to expose the internal system to the outside world.
2. Programmatic way of interfacing third party system.
3. Communication between different domains and technologies.



**Figure 1 REST API Design.**

The HTTP request methods are typically designed to affect a given resource in standard ways:

|  |  |
| --- | --- |
| HTTP Method | Action |
| GET | Obtain information about a resource |
| GET | Obtain information about a resource by id |
| POST | Create a new resource |
| PUT | Update a resource |
| DELETE | Delete a resource |

Here I have used the following things to build this API:

1. Python
2. Flask web framework
3. Flask-RESTful extension
4. SQLite3
5. SQLAlchemy

There are four files consists of models.py, main.py, setup.py, and config.py.

1. config.py:

This is the configuration file for the project. It consists the base directory path and SQLAlchemy database URL path.

1. setup.py:

This file is used to insert product and admin information into the database. Currently, there are two products and one admin information in the file. Before running the main.py file we need to run the setup.py file to create the database file. When we run the setup.py file this will create app.db file in the base directory. This file will take database model information from the models.py file. If you want to change the database model information, you need to follow the three simple steps:

* Change the models information as per your requirements.
* Delete the app.db file.
* Run the setup.py file to create the new database file.

1. models.py

This file contains the database model information. In this file, I have created three table for this information. One is for userprofile to store admin information, another is product which is used to store product information and lastly feedback which is used to store users feedback data into the database. I have used SQLAlchemy for this purpose.

1. main.py

This is the main file for this project. I have created basic HTTP authentication for admin login and two POST method (feedback and get\_rating). Feedback is used for normal users who can give feedback and get\_rating is used for the admin to get how many ratings is given for a selected time frame.

I have used HttpRequester (An Mozila Firefox Add-ons for easily making HTTP requests (GET/PUT/POST/DELETE) and viewing the responses, and keeping a history of transactions) to post feedback and get rating using admin login information.

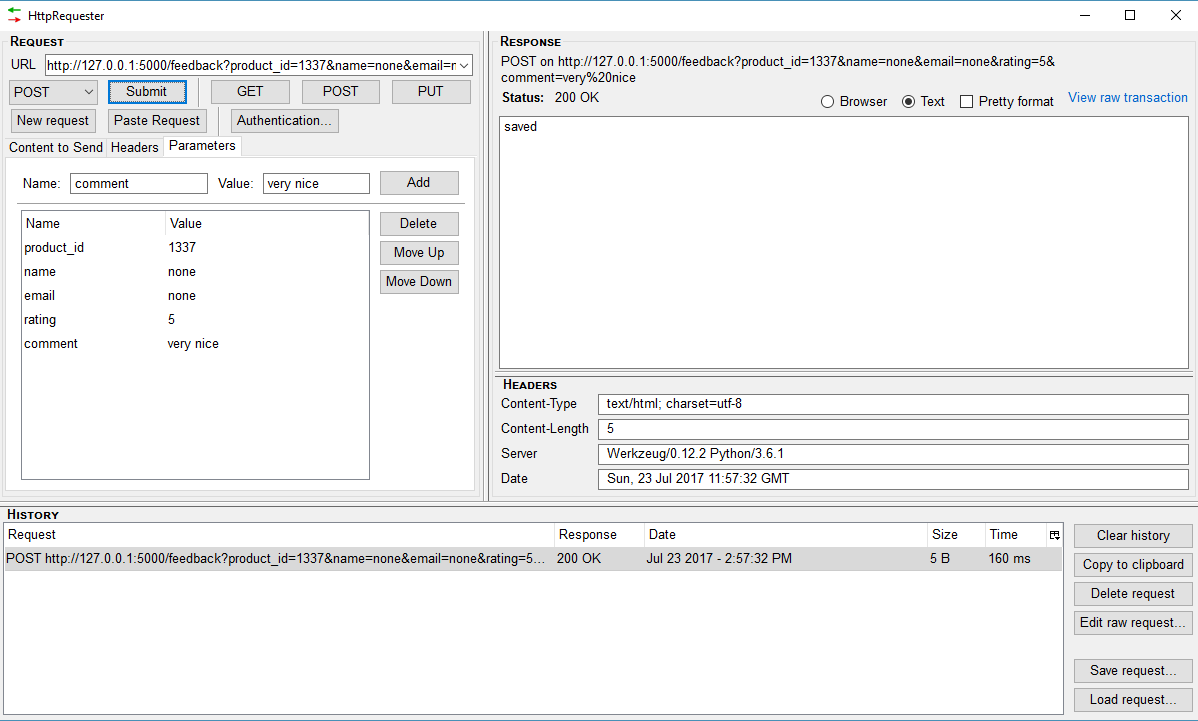


Figure 2 POST user feedback using HttpRequester.

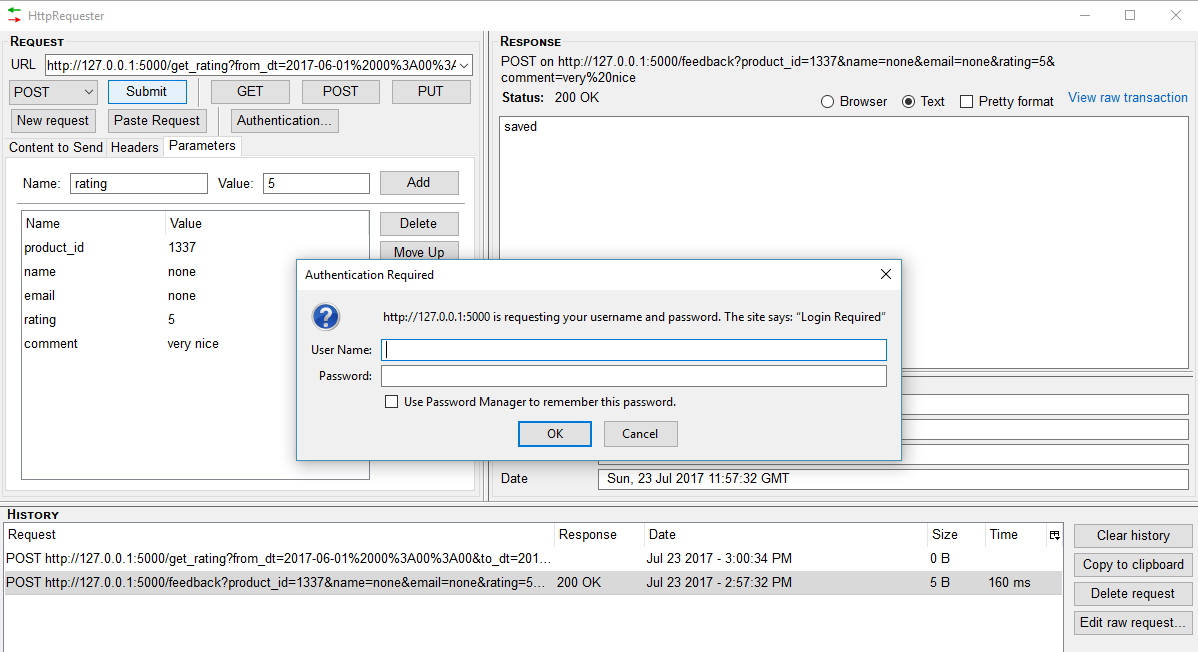


Figure 3 Admin authentication required before posting.

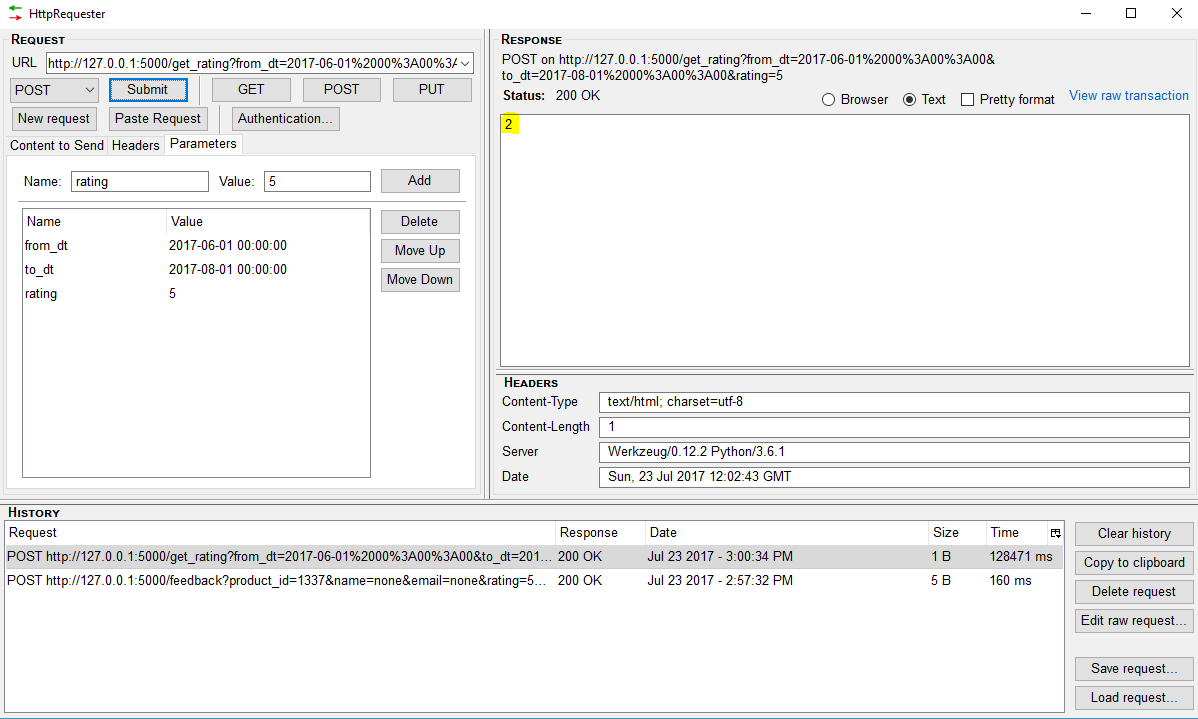


Figure 4 get\_rating POST response.

This POST method gives value 2 (yellow marked) because there are two feedback where users give rating 5.

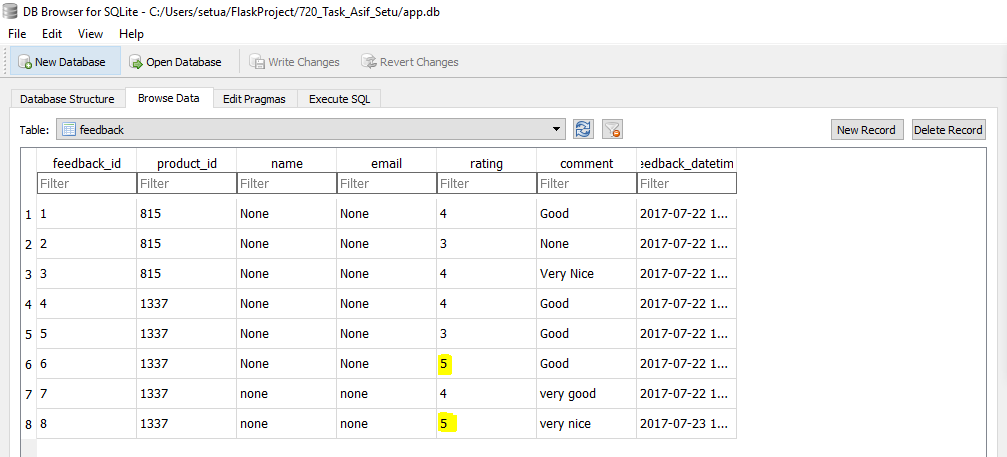


Figure 5 feedback table where number of rating 5 is two.