## Front-end Testing

User service part：

## User registration

Functional description：

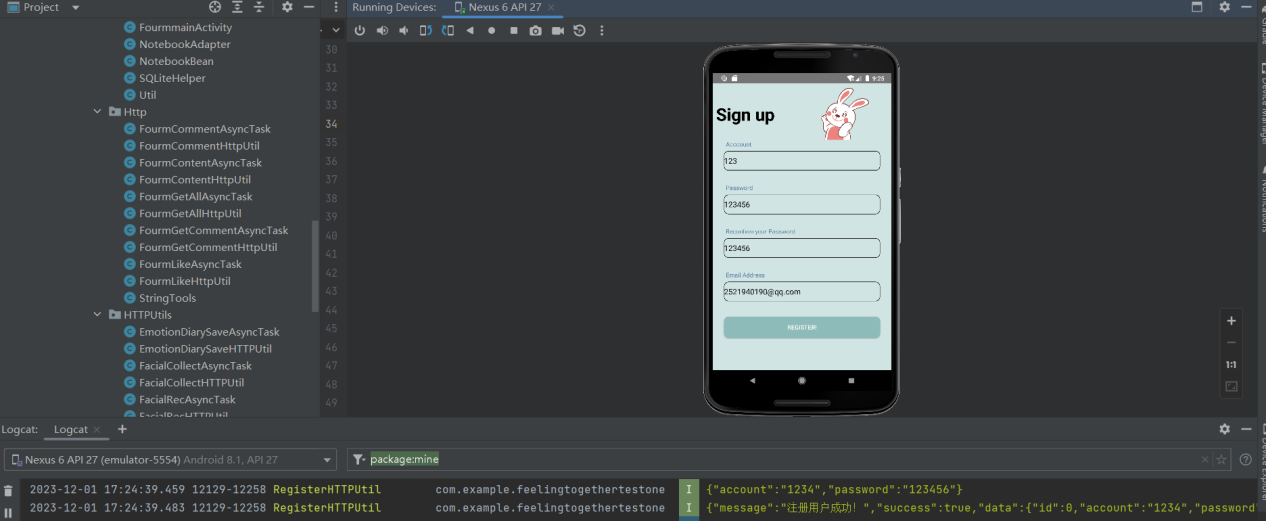
· After the user enters the account, password, confirm password and email address, the backend server is sent the registration request, and the data is transmitted to the server for account registration verification.

· If the "Confirm Password" is different from the "Password", or the account has been registered, the registration fails.

bug or not: None。

The actual interface is the same as the function description, and the test passed.

Interface display:



## Login with the account and password

Function description:

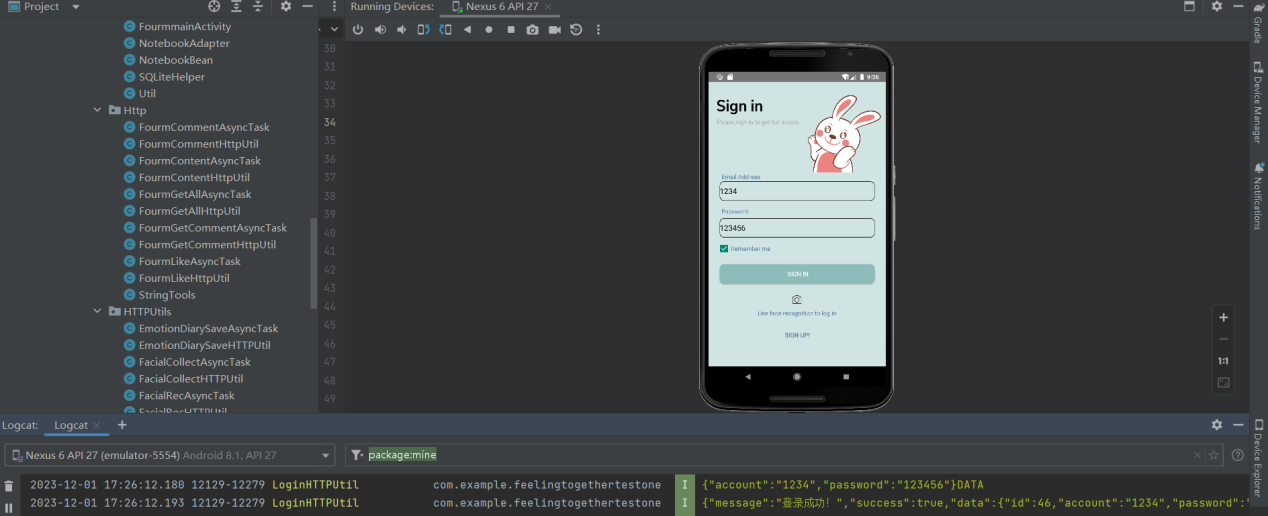
· After the user enters the account password, click the login button, send the login request to the back-end server, and transfer the data to the server for account login verification.

· You can choose "Remember Password" to realize the local cache of the account password, which is convenient to fill in the account password automatically when you log in later.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## User face information is saved

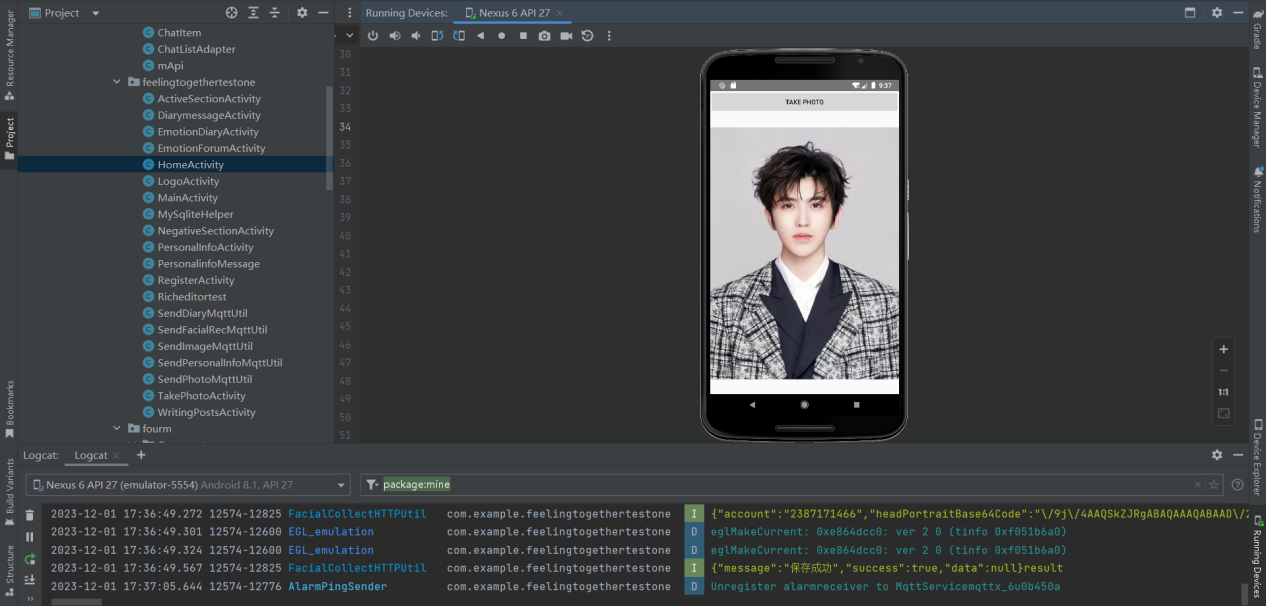
Function description:

· In the personal information interface, the user clicks "Portrait collection" to collect face information. Call the system camera to take pictures, and send a request to the back-end server to transfer the data to the server for portrait information storage. After that, users can log in through face recognition II.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## User face information login

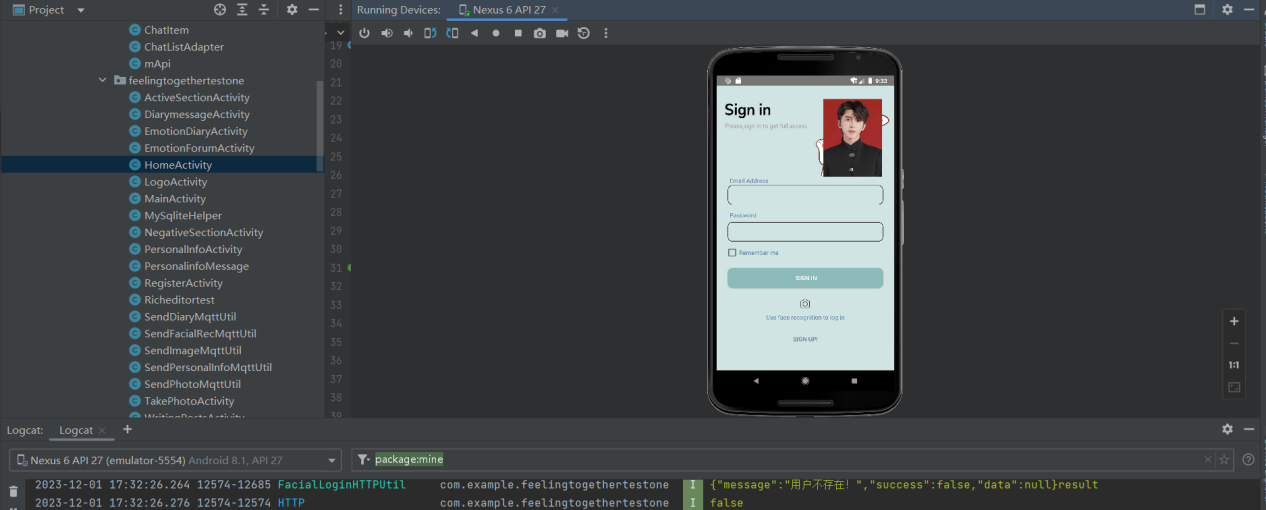
Function description:

· In the login interface, if the user has saved face information, the user can log in through portrait recognition. Call the system camera to take pictures, and send a face recognition login request to the back-end server, and transmit the data to the server for portrait verification.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## Update user information

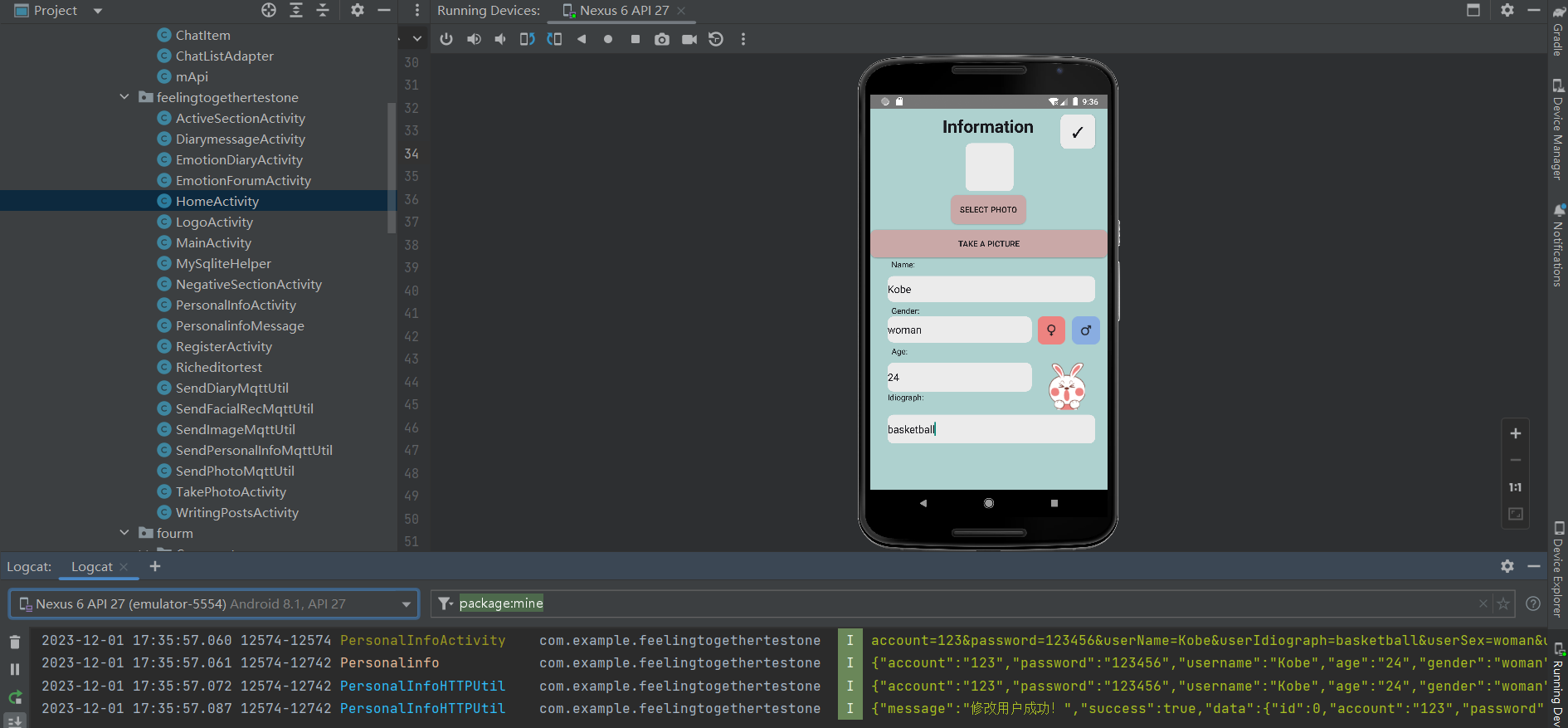
Function description:

· In the personal information module, the user can set the basic information of the account such as "nickname", "personal signature", "age", "gender", "avatar", etc. After clicking the Save button, the user can send the request for updating the user information back end, and the data can be transferred to the server for updating

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



Emotion forum section：

## Post a post and identify its emotional polarity

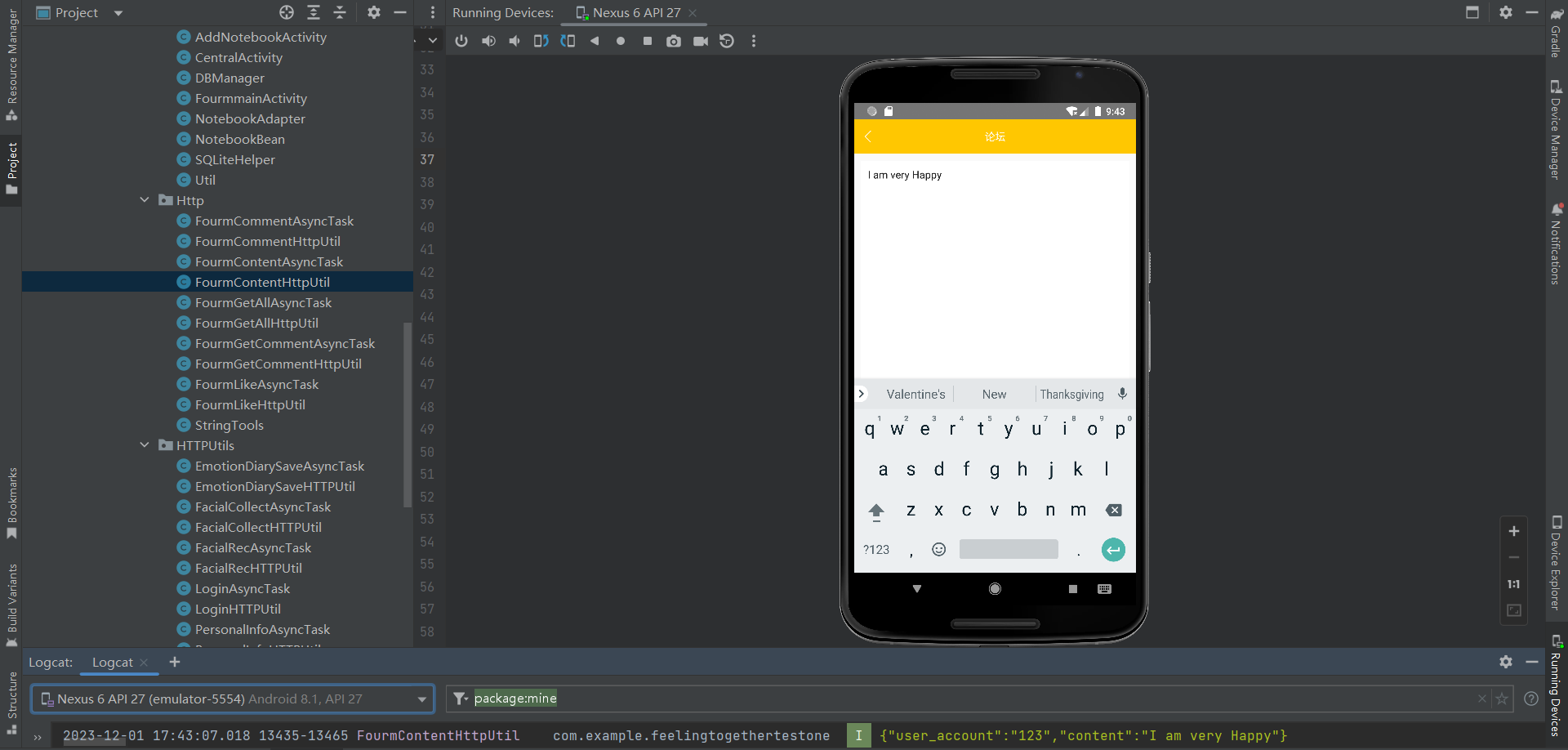
Function description:

· Post in the emotion forum, edit the content of the post and publish it. Post requests are sent to the back-end server, and the content of the post is transmitted to the server for sentiment analysis. Posts are placed in different emotional zones based on the emotional polarity returned by the server.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:

Posting

## Pull posts of both poles

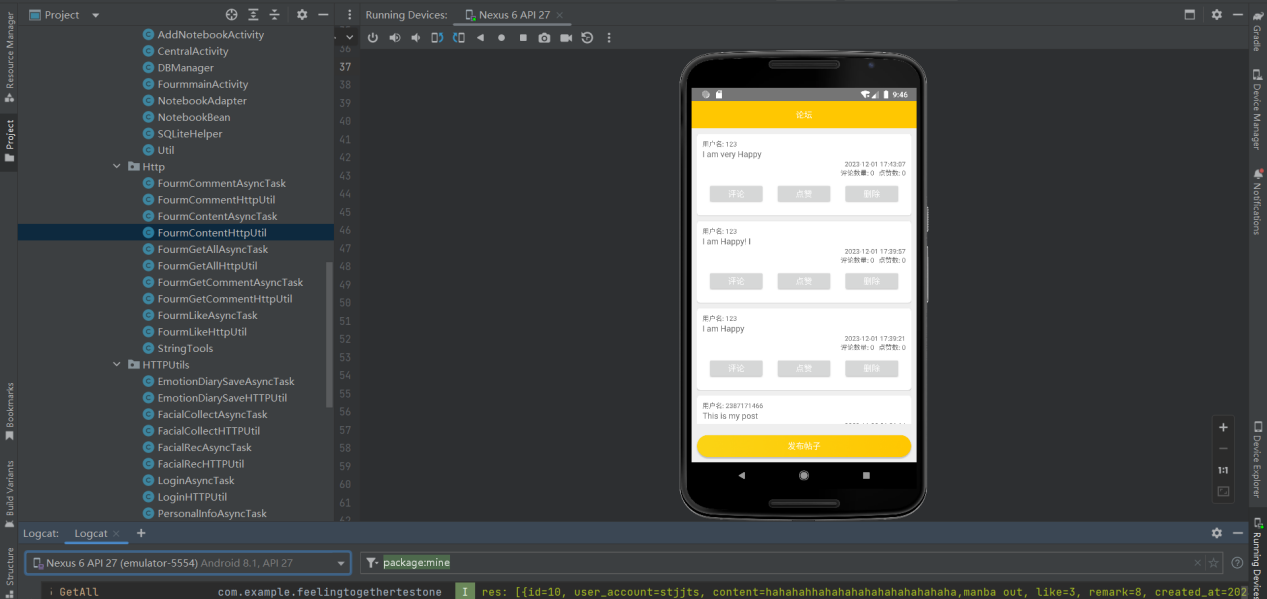
Function description:

Users can enter post partitions with different emotional polarities (currently only positive and negative partitions are implemented). When entering the partition, the backend server sends a request to retrieve posts, retrieve and update all posts of both poles, and display them on the front-end interface.

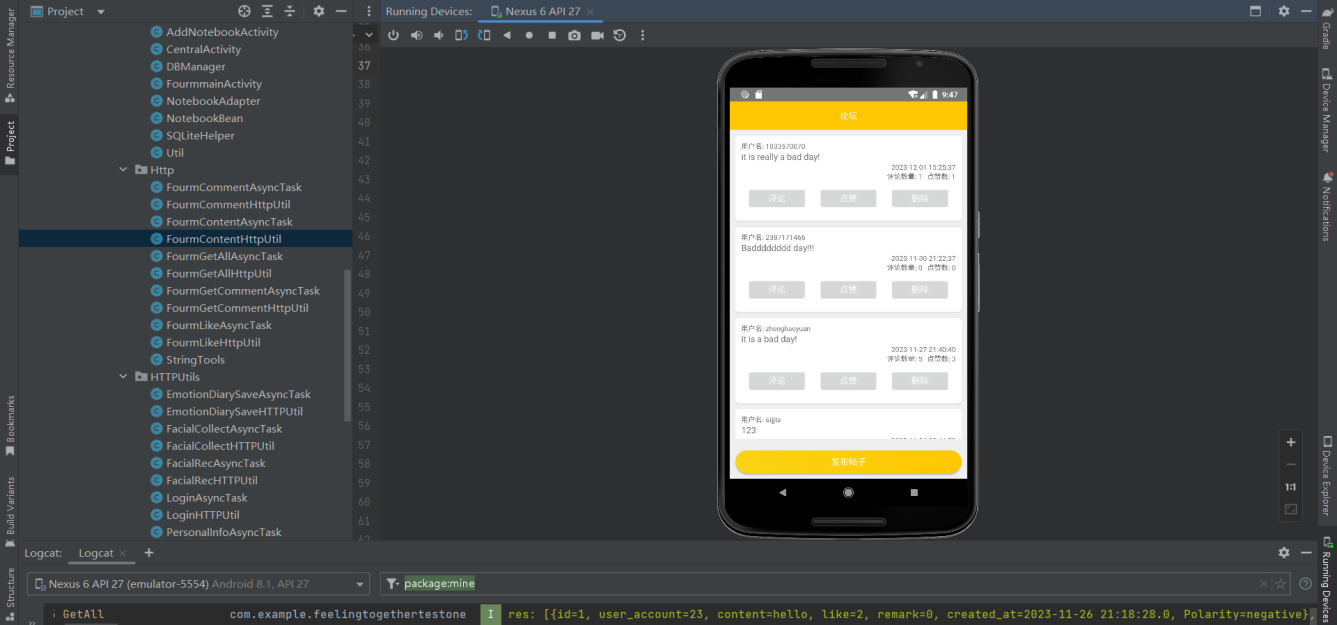
bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:

Positive:

Negative:



## Like and cancel posts

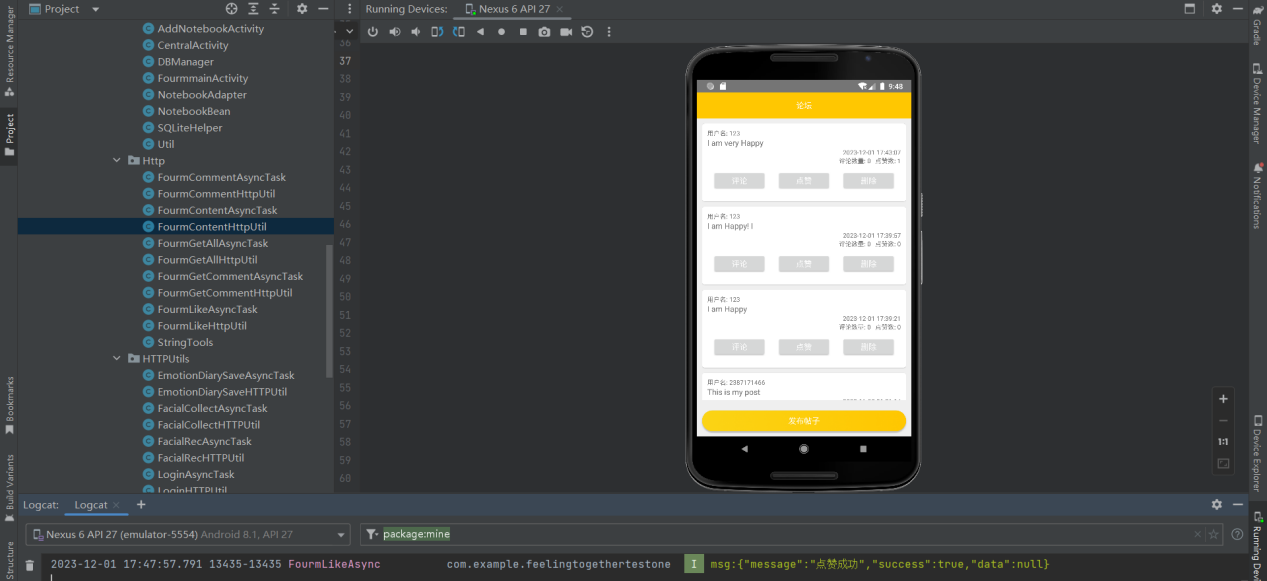
Function description:

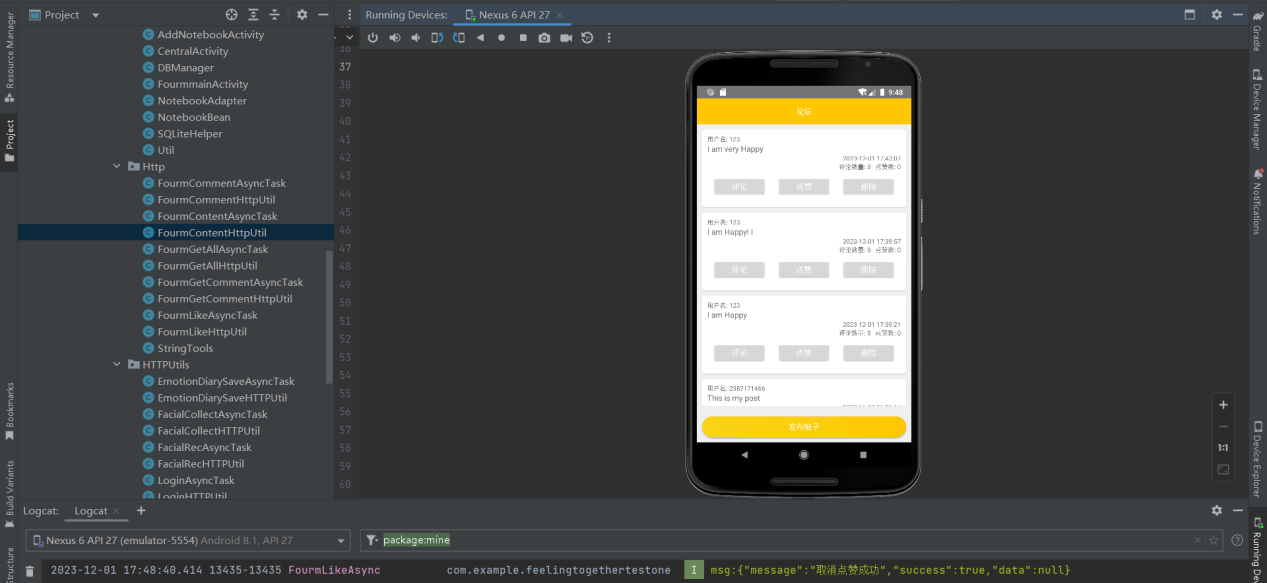
· Users can support likes for posts they like, send a like request to a backend server, and transfer data to the server for information changes.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:





## Post comments

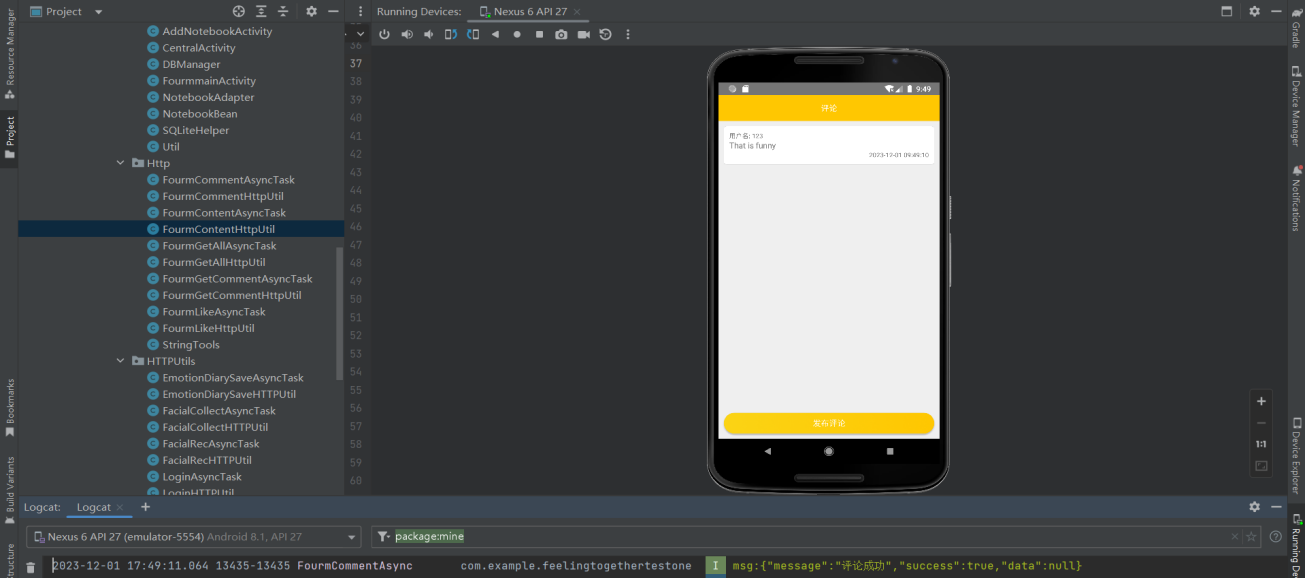
Function description:

· Users can comment on posts to provide emotional support. The back-end server sends a comment request and transfers the data to the server for comment storage.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## Get comments on posts

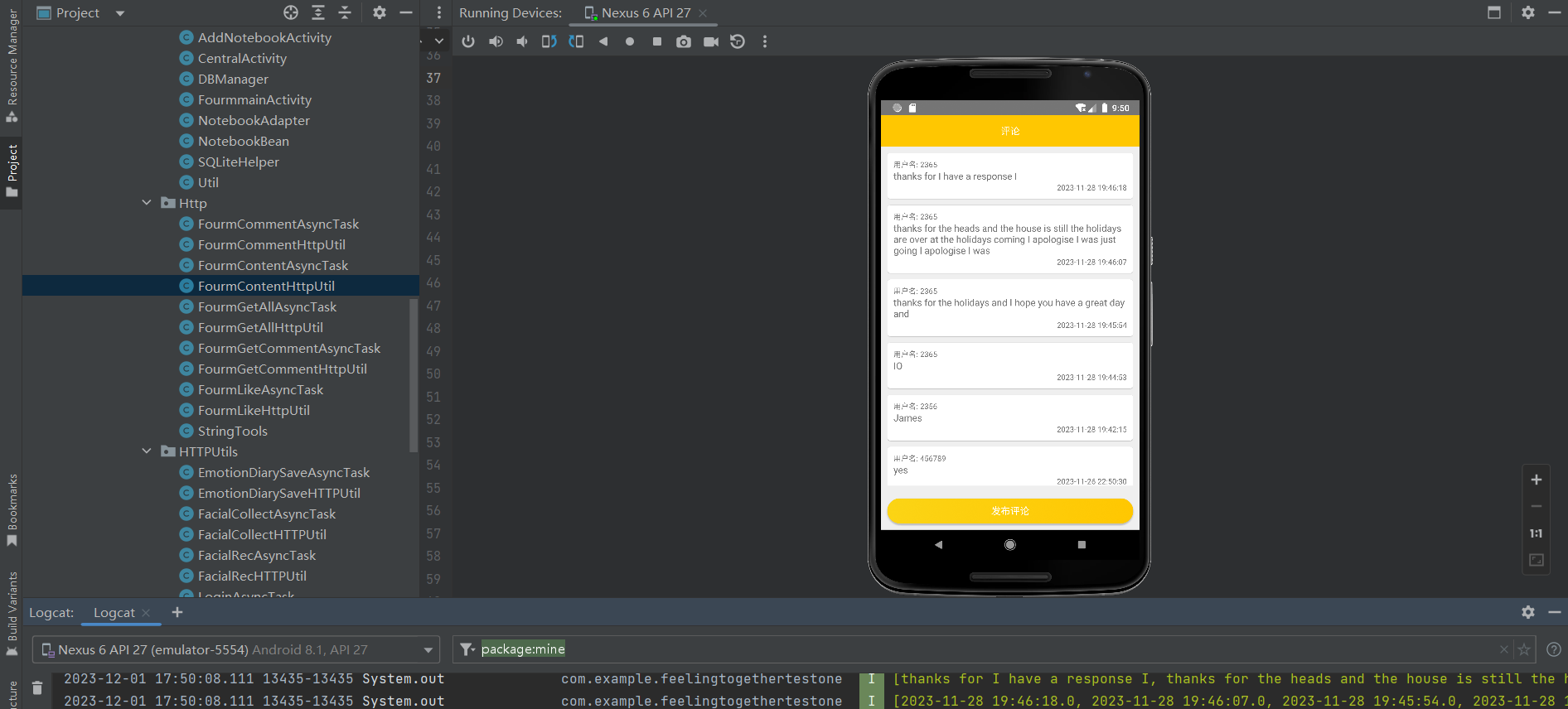
Function description:

· Users can view comments on any post. When viewing comments, the backend server sends a request to retrieve post comments, retrieve and update all comments under a post, and display them on the front-end interface.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



Emotional Diary：

## Identify journal emotional polarity

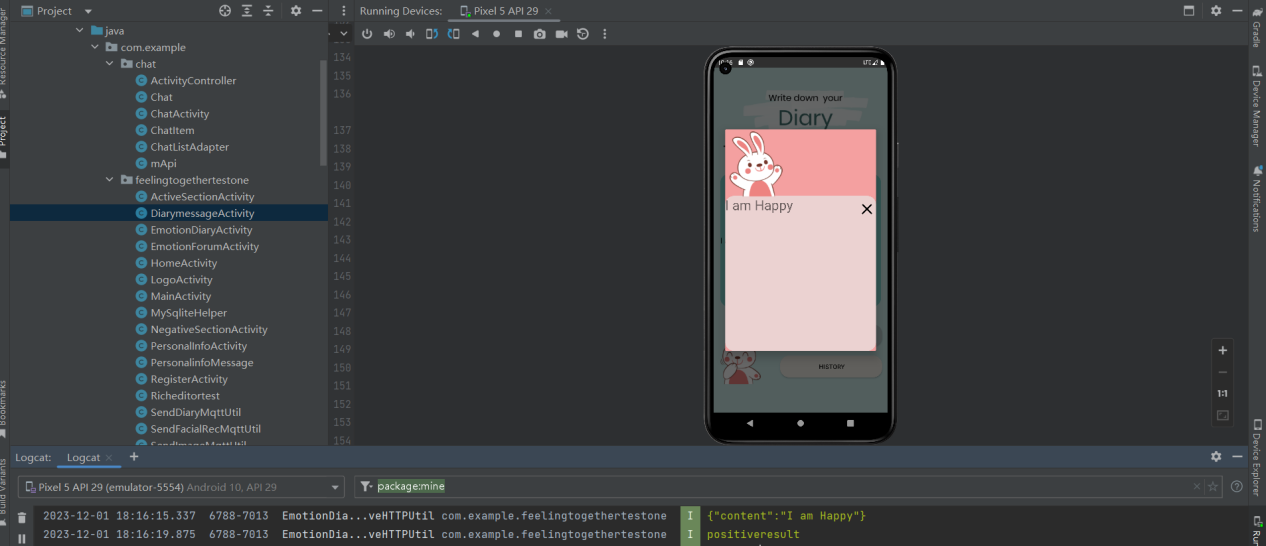
Function description:

· Users can write and save emotional diaries. When the user finishes writing and clicks Save, the back-end server sends a diary emotion recognition request and transmits the diary contents to the server for emotion analysis. Different Save diary UI interfaces are displayed depending on the emotional polarity returned.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## Get your relationship journal history

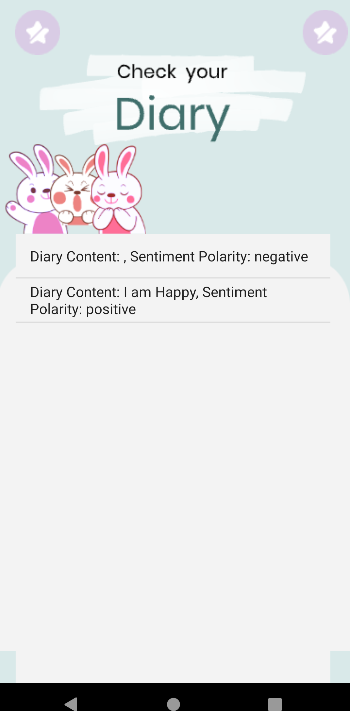
Function description:

· Users can write and save emotional diaries. When the user views the history record, the diary content and its emotional polarity are retrieved from the database, and different historical diary UI interfaces are displayed according to the emotional polarity.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



Emotional calendar：

## Display and record function of emotional calendar

Function description:

· When the user enters the interface, the calendar of the current month is displayed, and the current diary is orange

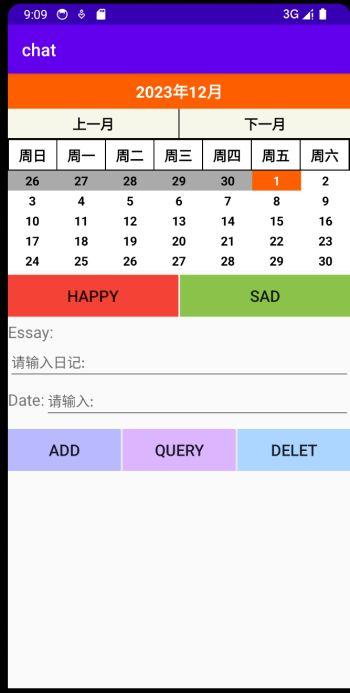
· Users can choose happy or sad according to the mood of the day. The date color changes accordingly.

· Users can enter a daily diary, and save it, and check the emotional essay of the day or before every day. It also provides the function of adding and deleting queries.

bug or not: None.

The actual interface is consistent with the function description, and the test passes.

Interface display:



Rich Editor Part ：

## Special font and image input

Function description:

· Realize three different functions of font bold, italic and underline.

· Realize the display of pictures

bug or not: None.

The actual interface is consistent with the function description, and the test passes.

Interface display:



Back-end Testing

User service part：

## User registration

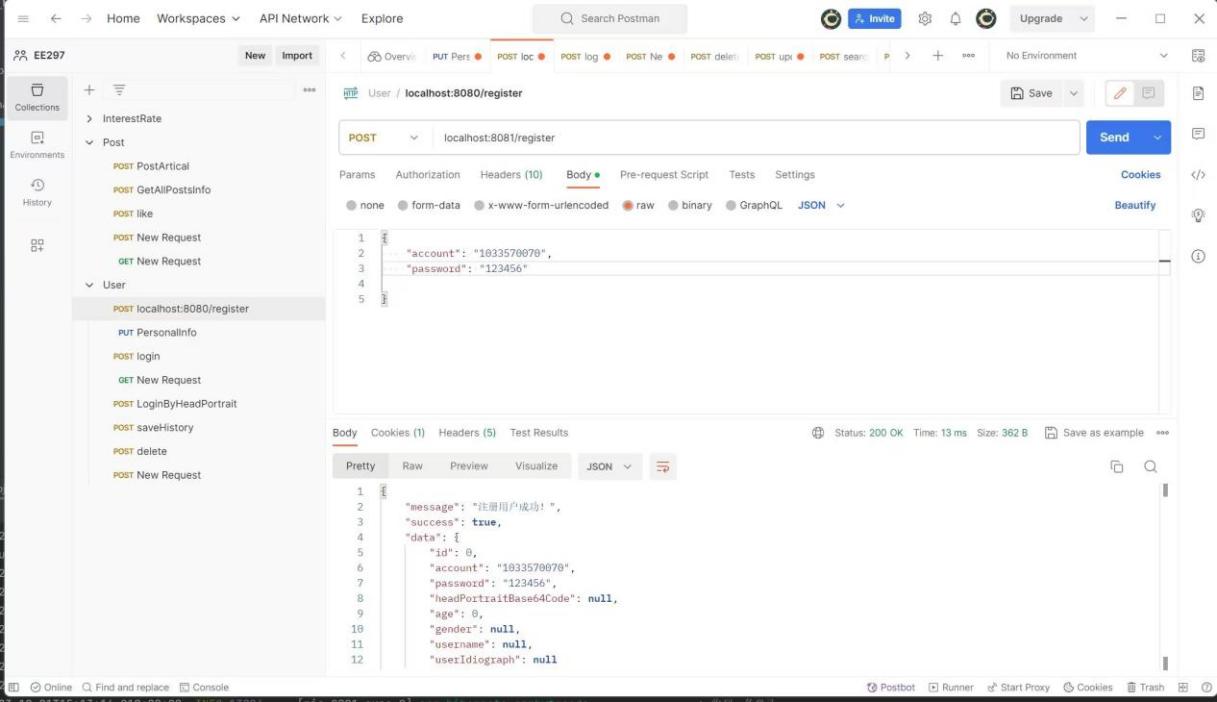
Function Description:

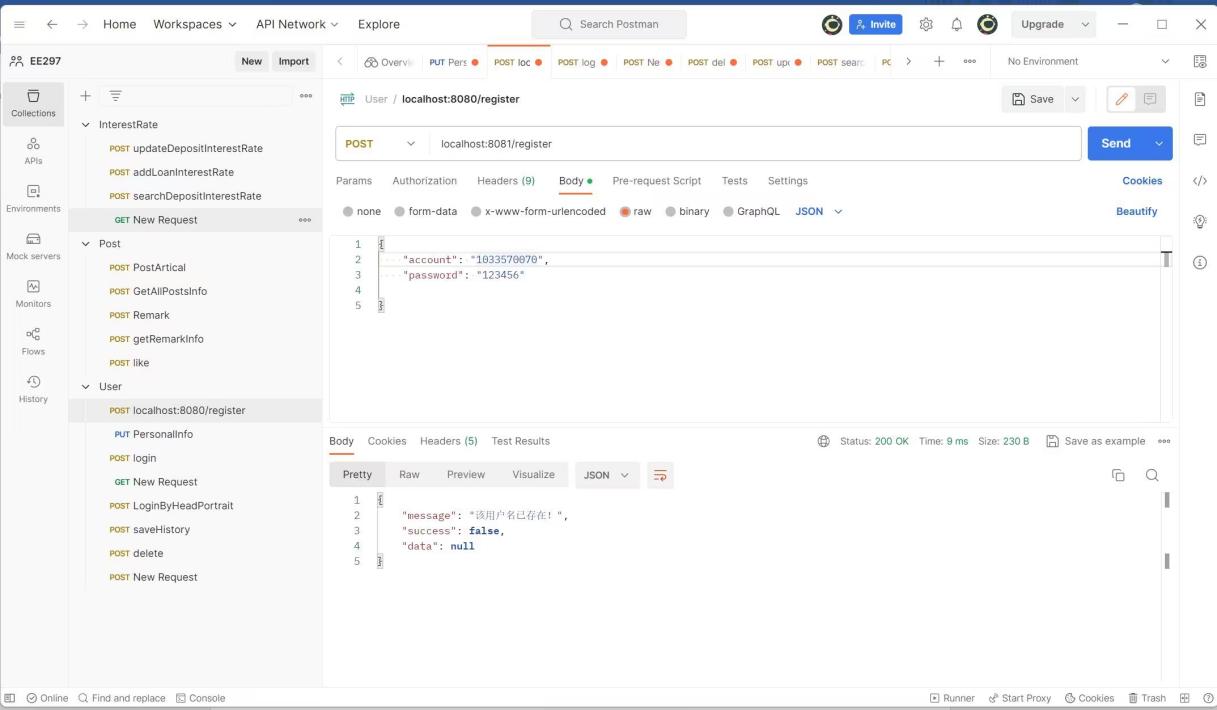
· Users fill in registration information on the front-end interface and send registration requests to the back-end. After receiving the registration request, the back end verifies whether the information provided by the user meets the requirements. If the requirements are met, the back end returns the response indicating that the registration is successful to the front end. Otherwise, the back end returns a message indicating that the registration fails.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:





## User login

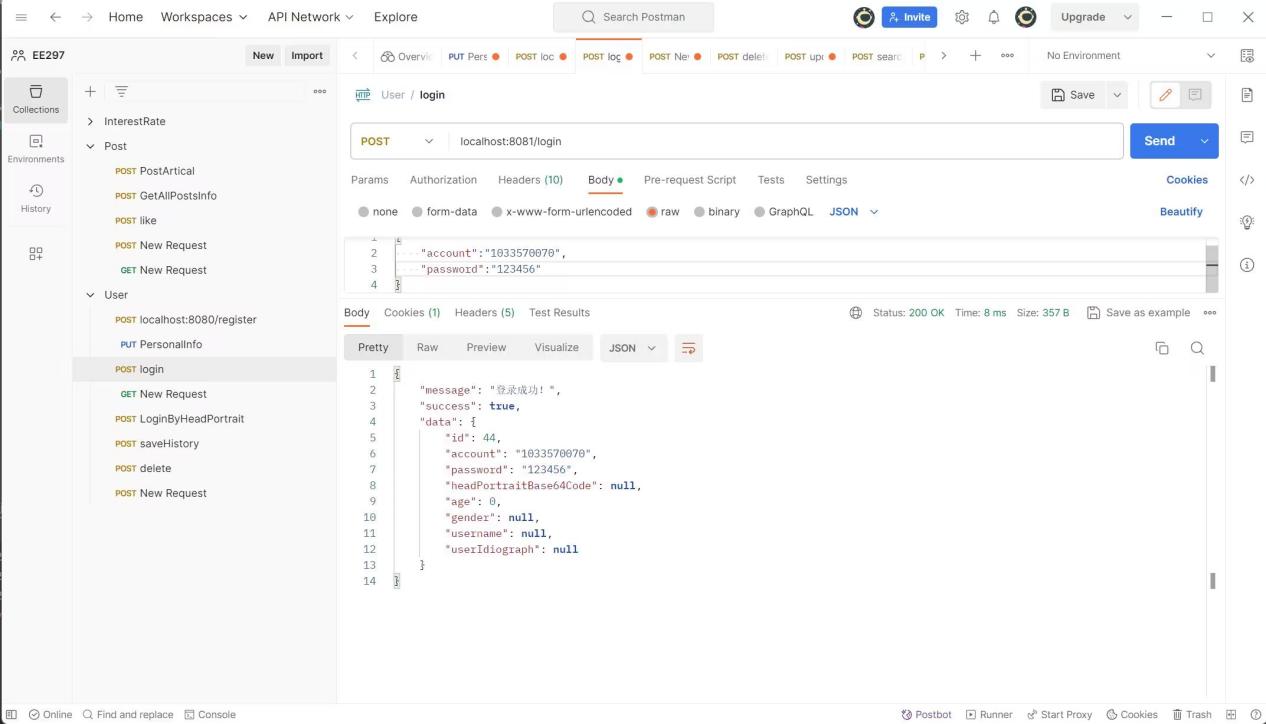
Function description:

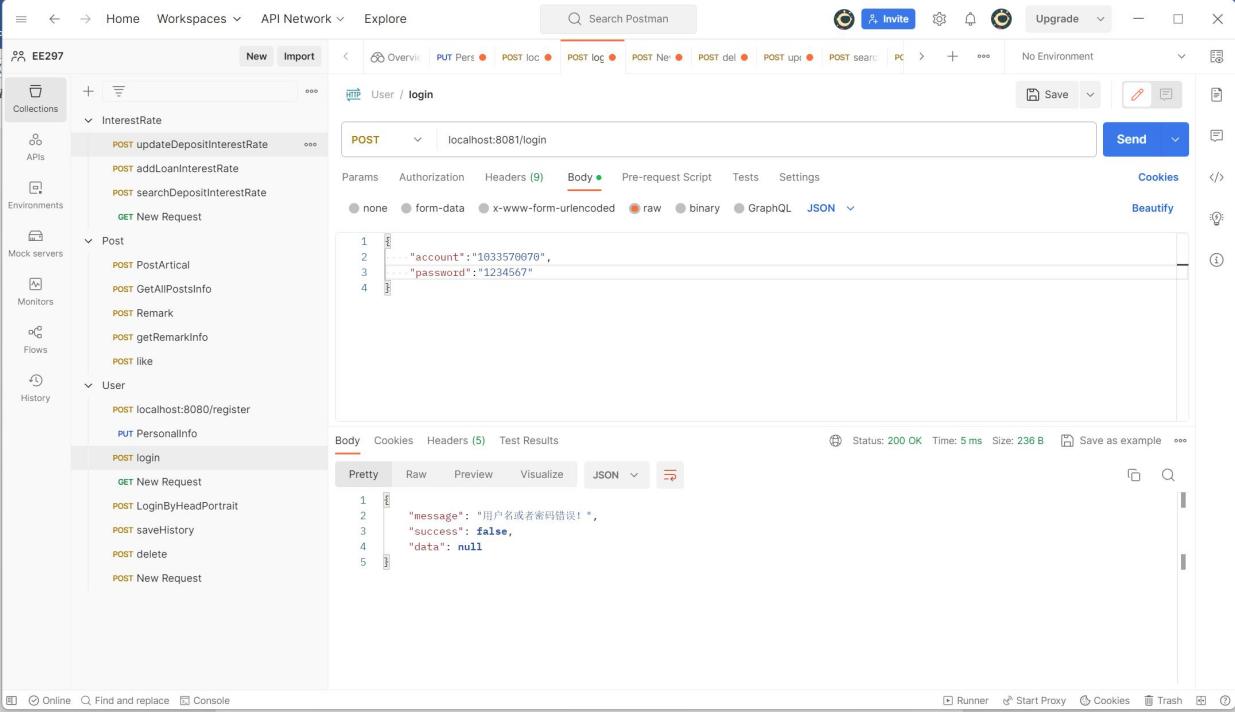
· The user enters the login data at the front end and sends the login request to the back end. After receiving the login request, the back end verifies that the data provided by the user is valid and correct. If the data is valid and correct, a response of successful login is sent to the front end and the user is allowed to access the appropriate function or resource; Otherwise, a login failure message is sent to the front-end, preventing users from accessing restricted resources.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:





## User face information is saved

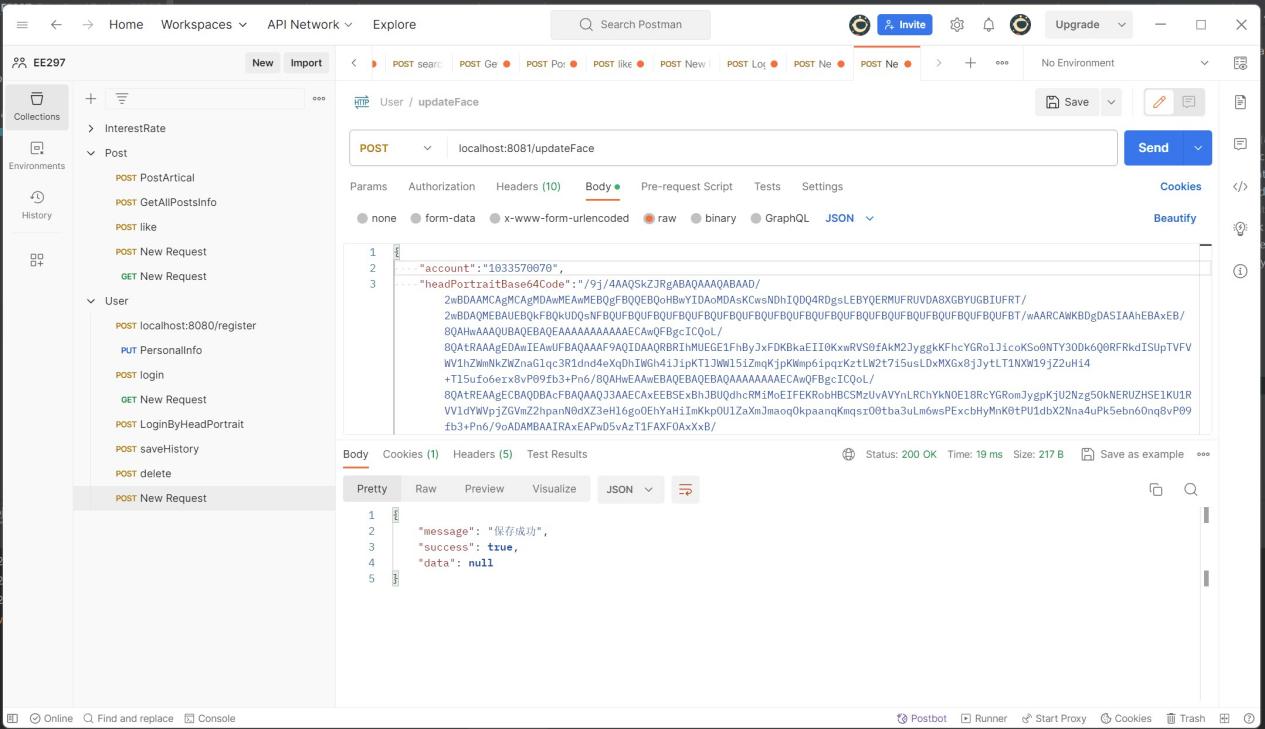
Function Description:

·After the user successfully registers, the user's face information is sent to the back-end through the front-end interface. After receiving this information, the back end processes and validates the face information, and returns the verification results to the front end. If the face information is successfully saved, the back end sends a successful response to the front end; If the save fails or the verification fails, the back end sends a message of the save failure to the front end, requiring the user to re-provide or verify the face information.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## User face information login

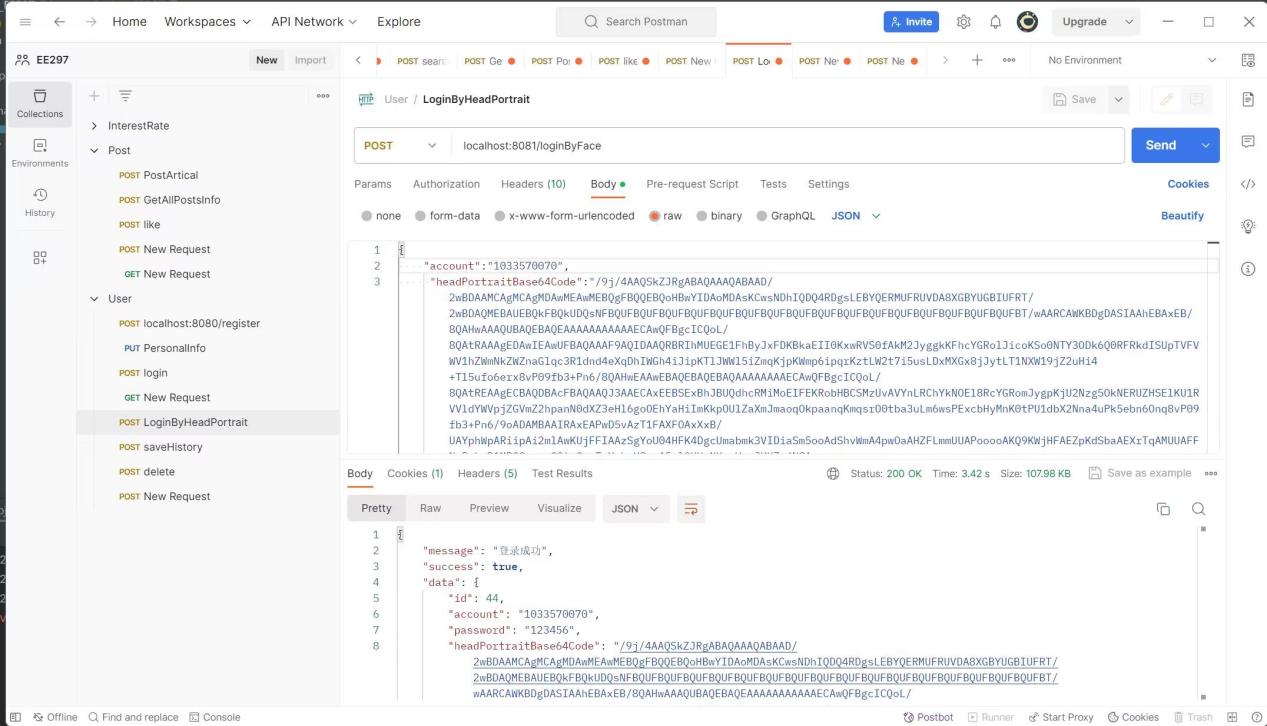
Function description:

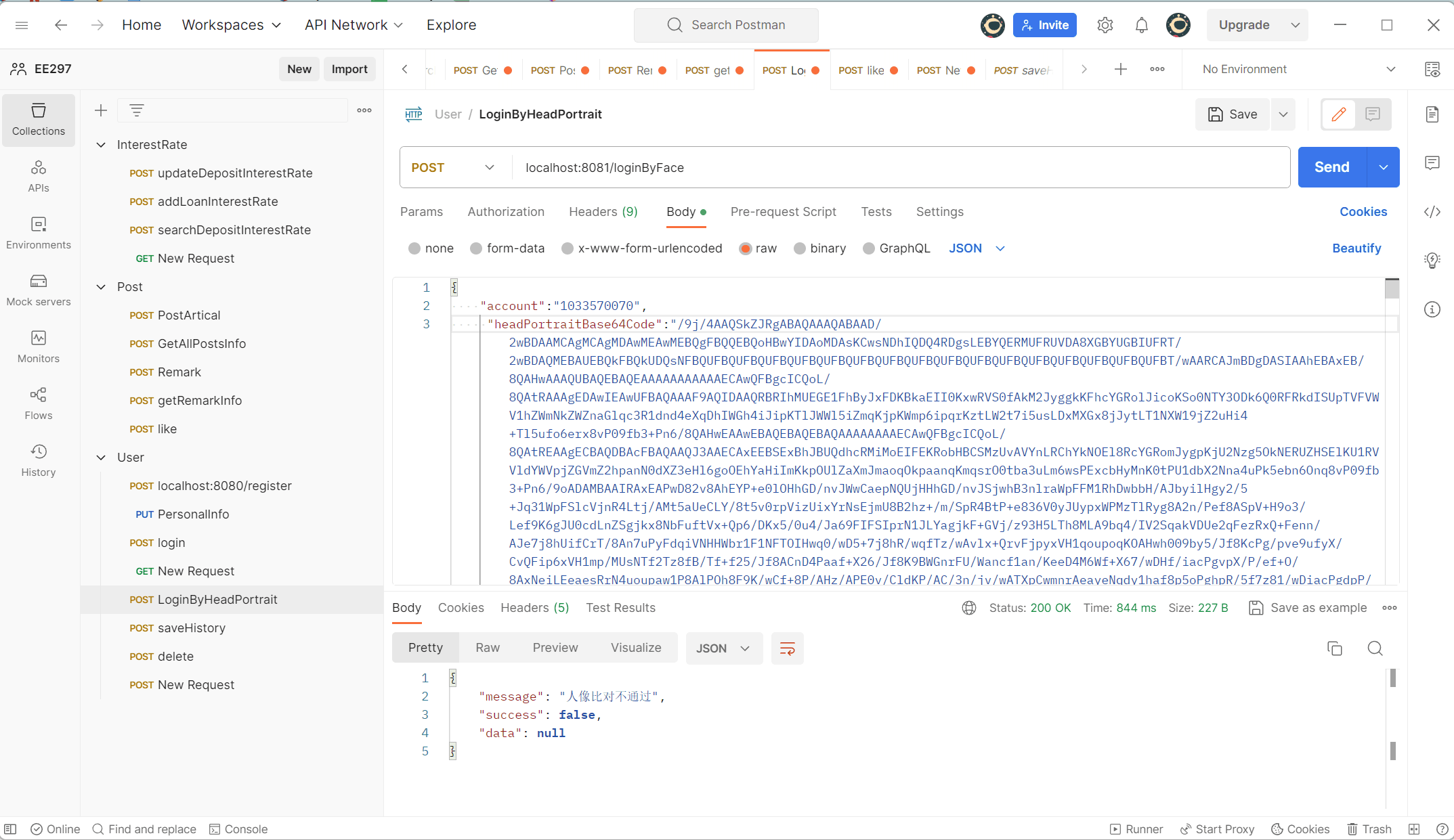
· The user uses the front-end interface for identity verification through face recognition, and sends the verified information to the back-end for login request. After receiving the user's face recognition information, the back end verifies and recognizes it. If the identification is successful and the verification passes, the response of successful login is sent to the front end, and the user is allowed to access the corresponding function or resource; If the identification fails or the authentication fails, a login failure message is sent to the front-end, denying the user access to restricted resources, and requiring the user to use other methods to log in.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:





## Update user information

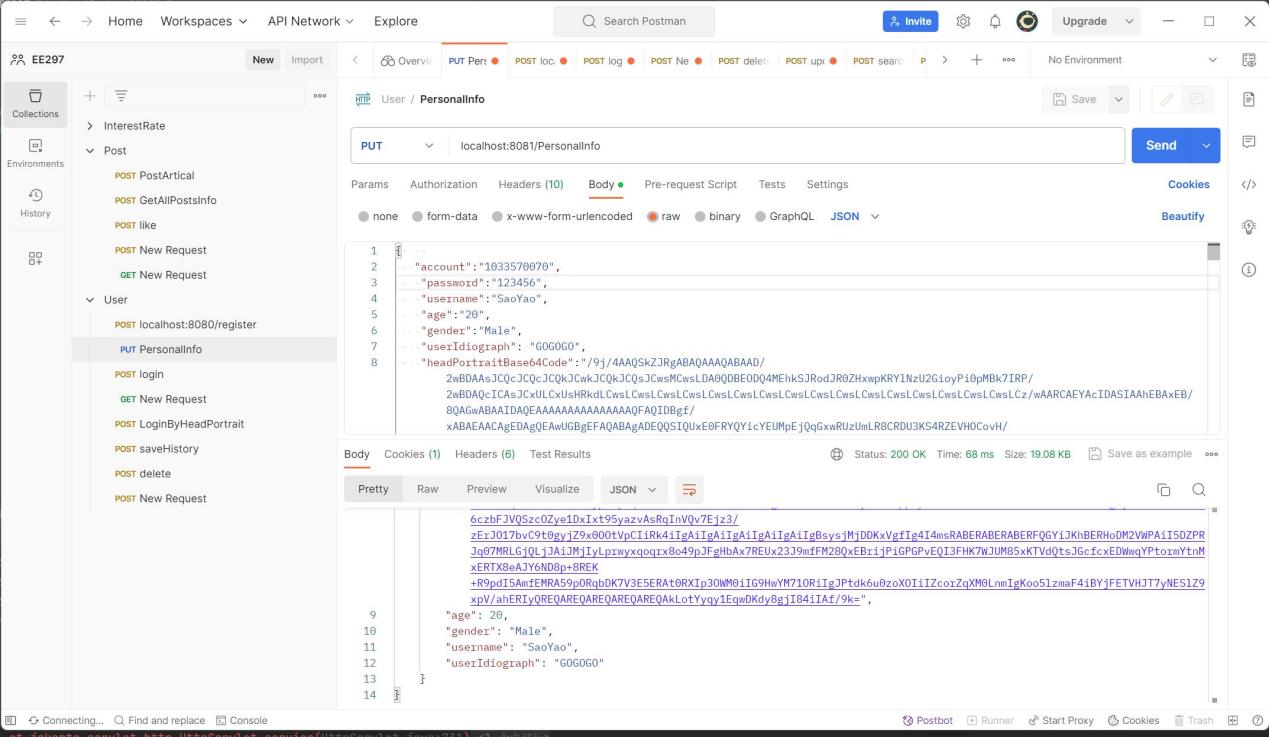
Function description:

· The user sends a request to update the user information through the front-end interface. After receiving the request, the back end validates and updates the information provided by the user. Once the information is updated, the back-end will confirm the success of the update and send a successful response to the front-end; If there is a problem during the update process or the information provided does not meet the requirements, the back-end sends a message that the update failed to the front-end.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



Emotional Forum section：

## Post a post and identify its emotional polarity

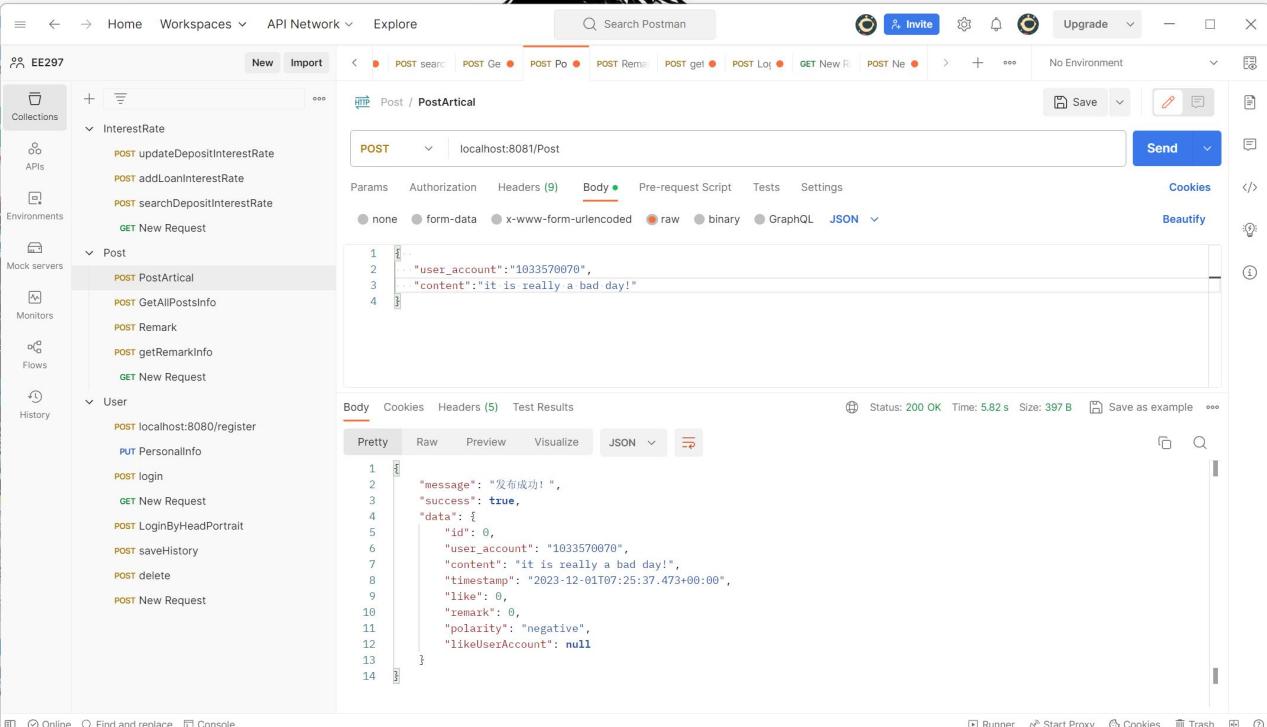
Function description:

· The user sends a post request to the backend through the front-end interface. After receiving the request, the back end processes the content of the post provided by the user and performs the corresponding action. If the operation is completed successfully, the back end sends a confirmation message to the front end that the post is successful, along with the emotional polarity information of the post; Emotional polarity information may include the positive or negative emotions of the post and is used to describe the emotional tendencies of the content of the post. If the operation fails or an error occurs, the back end sends a message to the front end that the publication failed, which may require the user to retry the publication or make necessary corrections.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## Pull posts of both poles

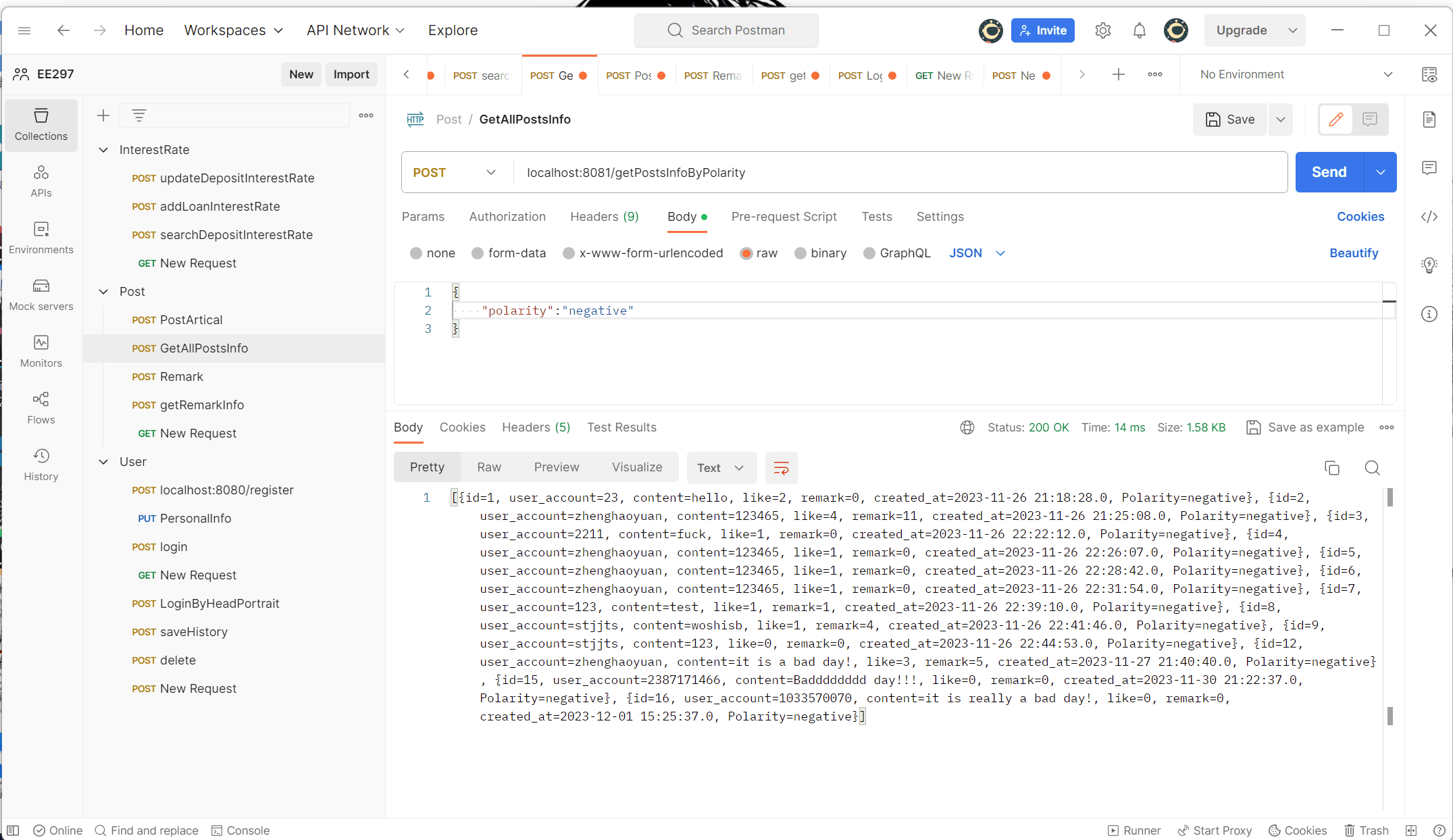
Description of Function:

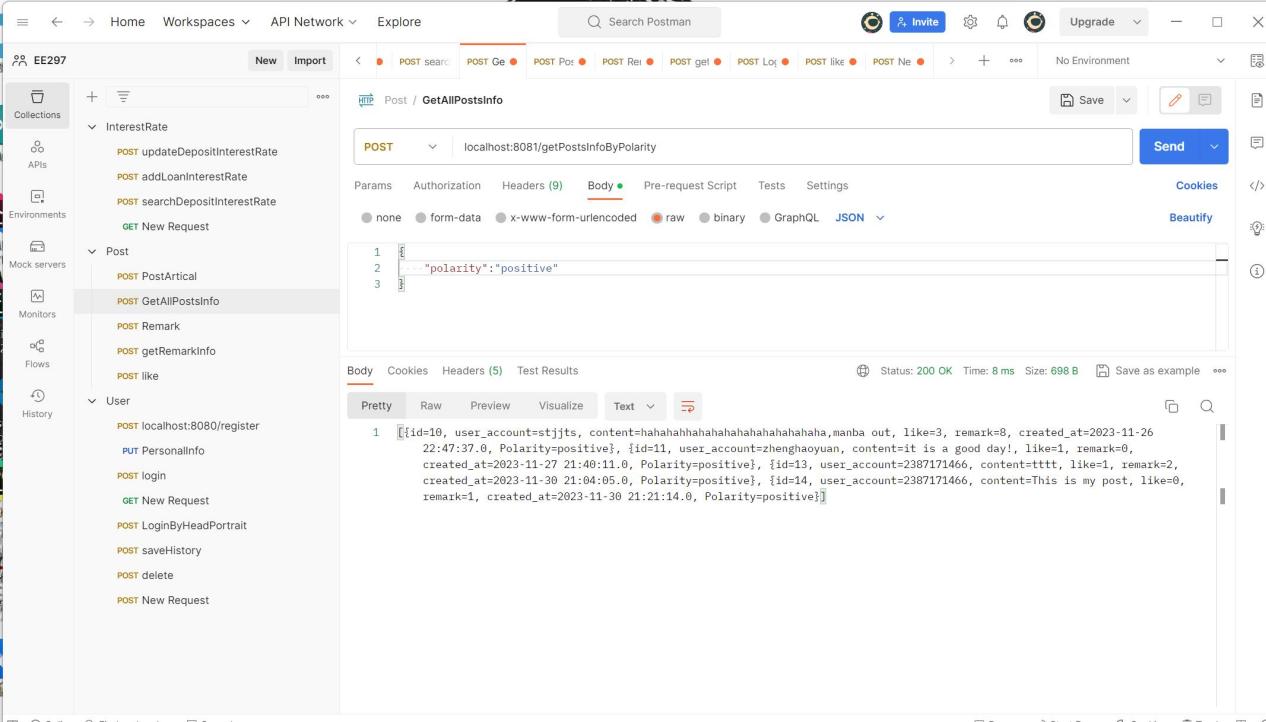
·The user sends a request through the front end to the back end, asking for all posts in one of the two polarities. After receiving the request, the back end filters and retrieves all post information of the corresponding type according to the specified polarity. The back end then sends the post data that meets the requirements to the front end as a response. This response may contain information such as the content of the post, the author, a timestamp, and be provided to the user by polarity. If the request is successfully executed, the back end replies to all posts on one of the two poles specified; If there is an error or if the post with the specified polarity is temporarily unavailable, the back end will either reply with a failed request or return an empty list of posts.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:





## Like posts

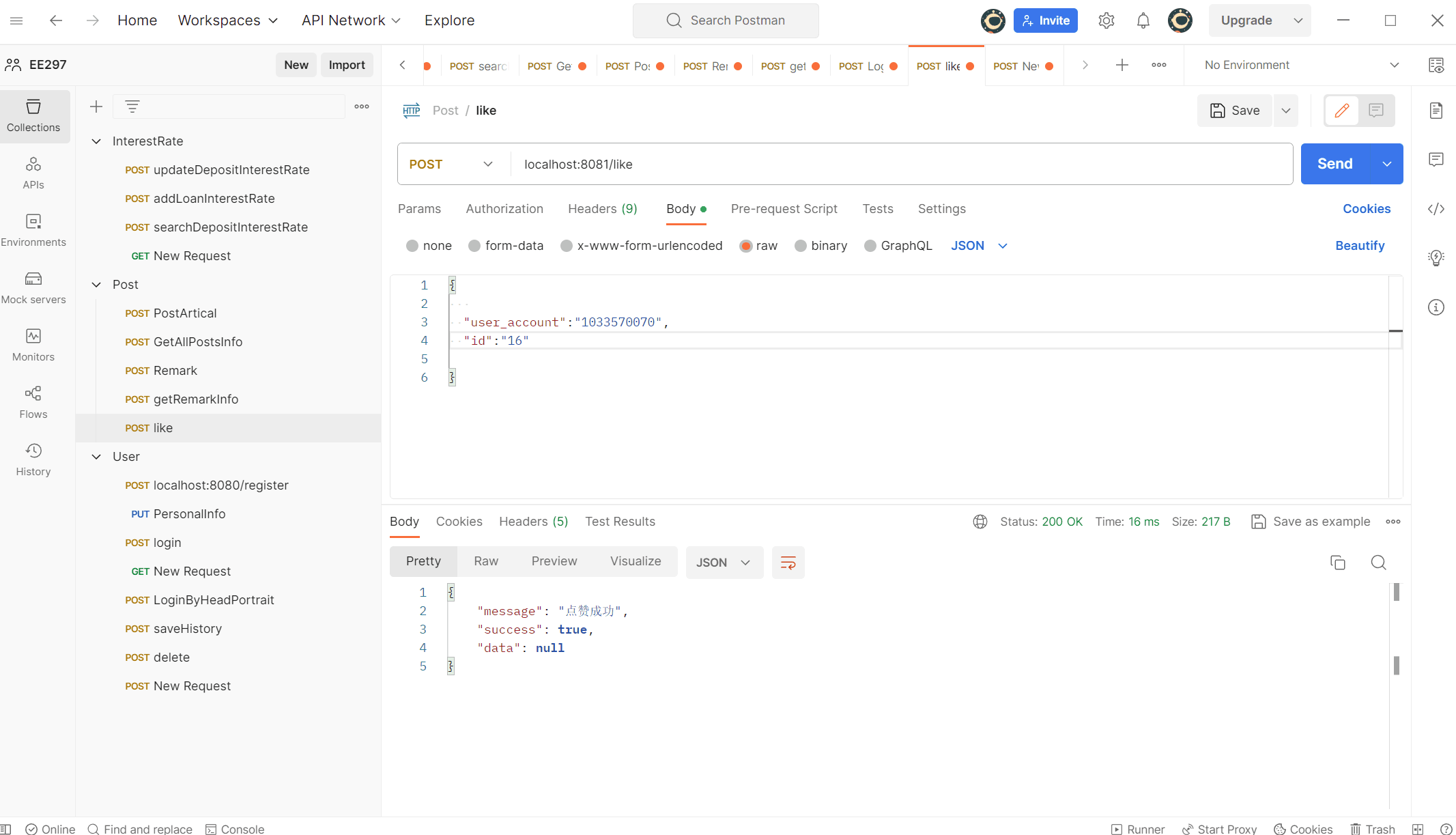
Function description:

· The user clicks a "like" on the front-end interface and then sends a "like" request to the back-end. After receiving the request, the backend processes the request and verifies the validity of the request. If the "like" operation meets the requirements and is successfully completed, the back end sends a confirmation message of successful "like" to the front end; Otherwise, the back end sends a response with a failed like to the front end.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



## Post comments

Function description:

· Users comment on specific posts in the front-end interface and send the comment content to the back-end. After receiving a comment request, the back end verifies the legitimacy of the comment and associates or stores the comment information with the corresponding post. If the comment operation is successfully completed, the back end sends a confirmation message to the front end that the comment is successful. If the comment operation is abnormal or fails validation, the back end sends a response to the front end that the comment failed. A successful response to a comment usually contains relevant information, such as comment content, comment time, commenter information, etc., and updates the number of comments on the post or other relevant statistics.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:

![5D4%[]()_~$VY`WC_[@376S](data:image/jpeg;base64,)

## Get comments on posts

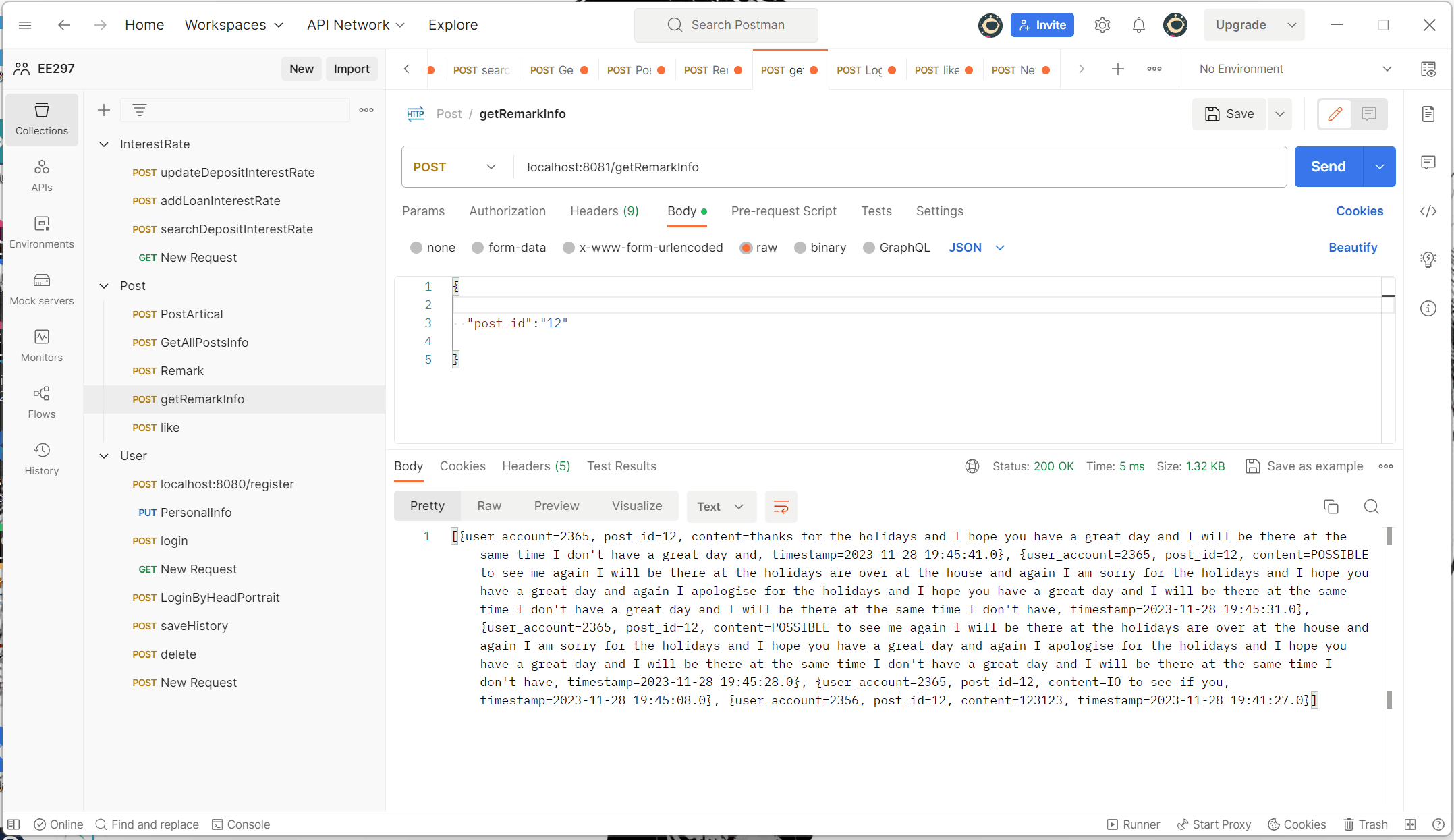
Function description:

· The user sends a request through the front end to get all the comment information under a particular post. After receiving the request, the backend retrieves all comments under a particular post in the database or store based on the post identifier or other relevant information. The back end then sends all the comments it gets as a response to the front end. This response may contain the content of the comment, the time of the comment, the information of the reviewer, etc., and is provided to the user in a certain format or structure. If the request is successfully executed, the backend will reply to all comment information under a particular post; If an error occurs or there are no comments under the specified post, the backend will either fail to respond to the request or return an empty list of comments.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:



Emotional Diary：

## Identify journal emotional polarity

Function description:

·The front end sends the diary contents to the back end, and the back end uses the pre-training model to identify the emotion polarity of the diary contents. After receiving the diary content, the back end feeds it into a pre-trained sentiment analysis model for processing and analysis. Then, according to the analysis results of the model, the back end will interpret the emotional polarity of the diary content and send the corresponding emotional polarity reply to the front end. This response contains the emotional tendencies of the diary content, such as positive or negative emotions, so that the user can understand the emotional tendencies expressed in the diary content.

bug or not: None.

The actual interface is the same as the function description, and the test passed.

Interface display:

