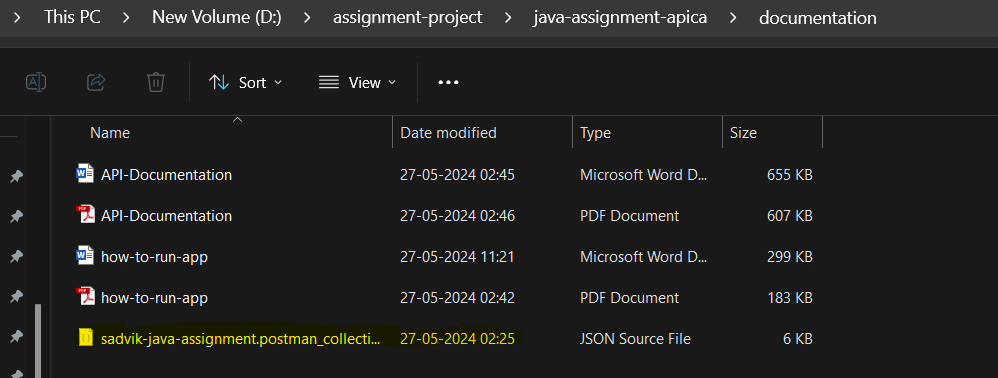
Java Assignment

Two spring boot microservices are created to maintain user details and to log the user events.

The user-service application does the CRUD operations like register, get, update and delete user details and produce the event message to user-event kafka topic. And the other application journal-service consume message from user-event topic and write those messages into database, and Admin user can access those user event messages

Prerequisite:

Please import the postman collect from the below repo.  
GitHub link :- <https://github.com/Sadvik-gowda-BD/java-assignment-apica>  
  


**1)user-service application**

This application does the CRUD operation on user details and publish the message

“**user-event**” kafka topic.

GitHub link for source code: - [GitHub - Sadvik-gowda-BD/user-service](https://github.com/Sadvik-gowda-BD/user-service)   
  
**Api endpoints:-**

Here we are going to save only below fields:

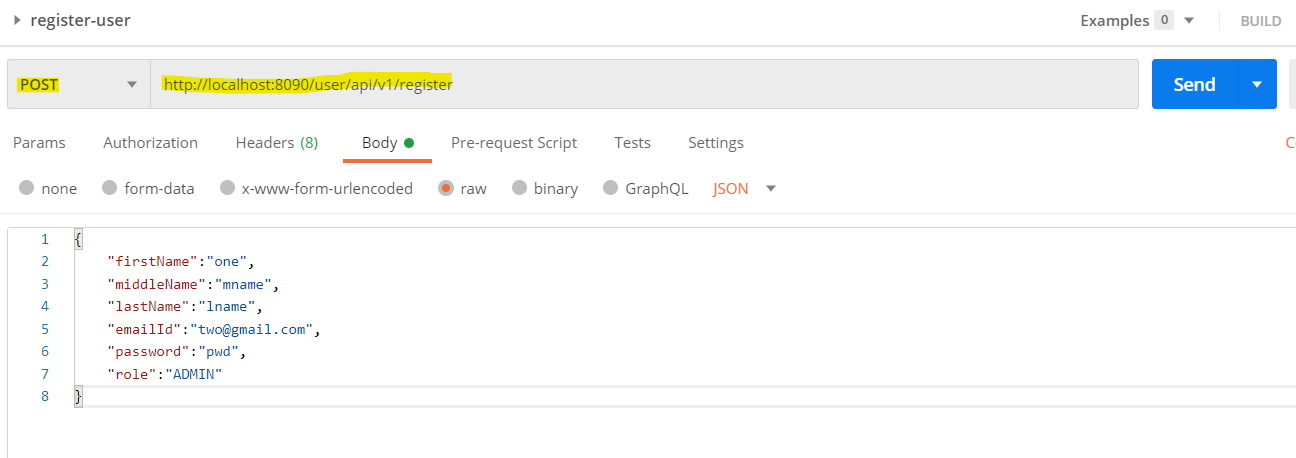
|  |  |
| --- | --- |
| **Field** | **Mandatory/Optional** |
| "userId" |  |
| "firstName":"one", | Mandatory field |
| "middleName":"mname", | Optional field |
| "lastName":"lname", | Mandatory field |
| "emailId":"two@gmail.com", | Mandatory field and email validation check is applied. Email should be unique for all customers. |
| "password":"pwd", | Mandatory field |
| "role":"ADMIN", | Mandatory filed. (Expected value is only **USER** or **ADMIN**) |

Note:- Please import and use postman collection to invoke API.

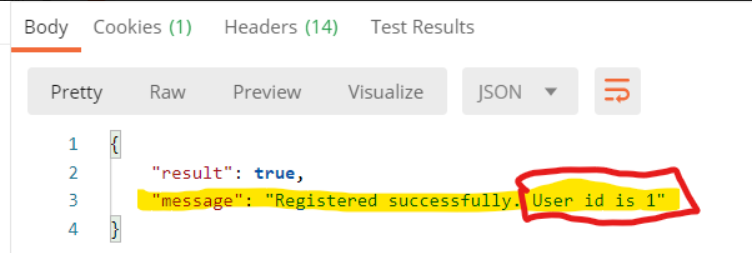
**1)Register:**

This API is used to register the user.

**Endpoint**: POST: <http://localhost:8090/user/api/v1/register>   
  
**Request**: JSON object user details



**Authorization**: No auth  
  
**Response**: user id, this user id will be used for authentication and identify the user.  
In below image 1 is the user id.

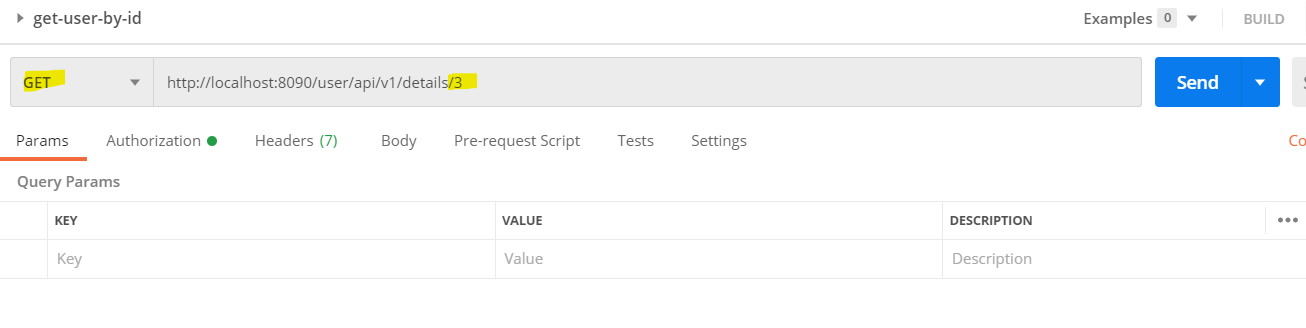


**2) get-user-details-by-id**

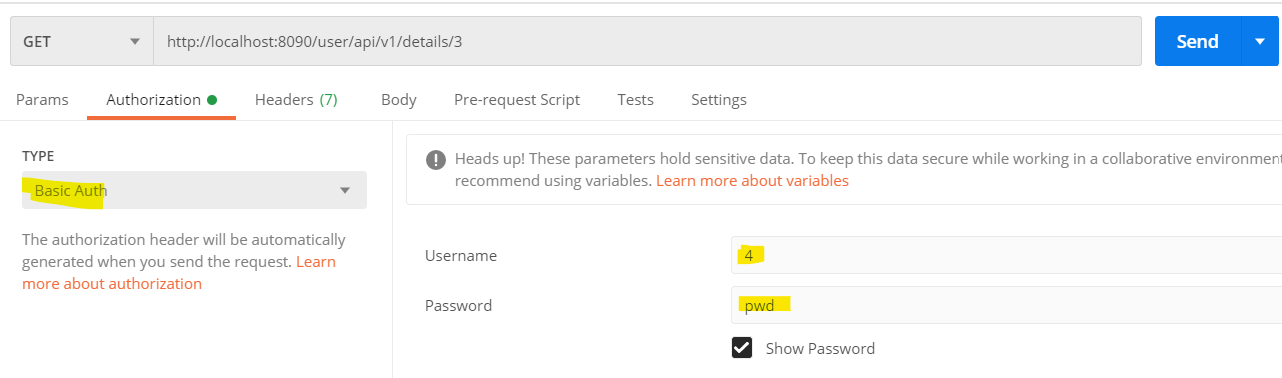
This API is used to get user details by user id (user id is returned after user registration). Only ADMIN can access this endpoint.

**Endpoint**: GET <http://localhost:8090/user/api/v1/details/3>

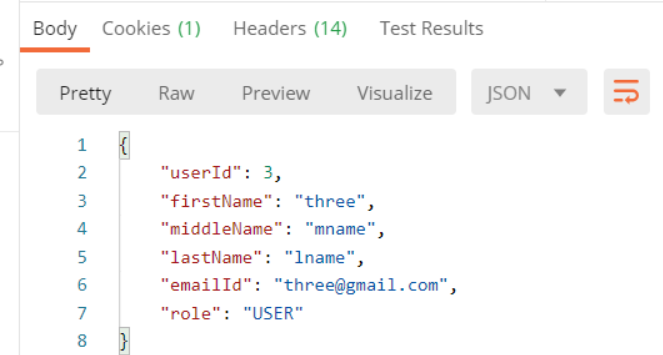
**Request**: User id in URL(path variable)  
Here 3 is the user id.



**Authorization**: Admin user id and password



**Response**: user details.

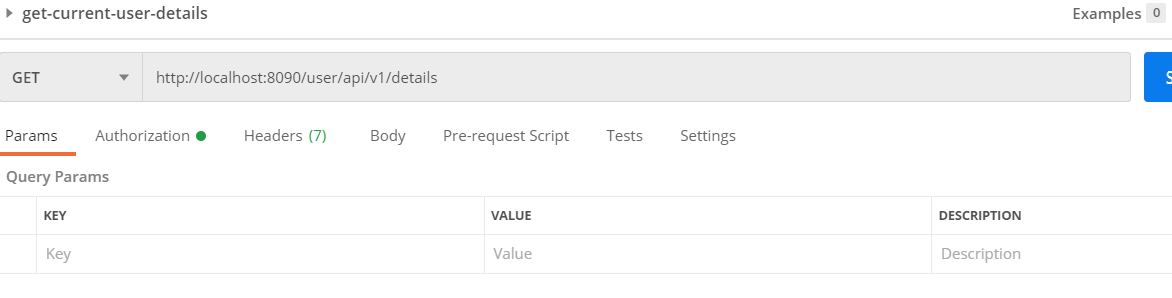


**3)get-current-user-details**

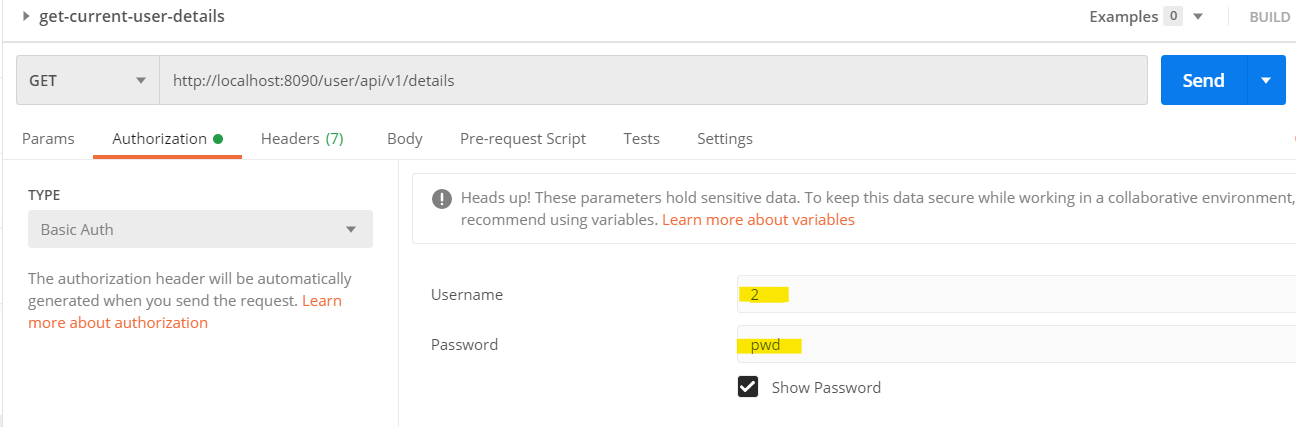
This API is used to get the current login user details. Any user can access this endpoint with user id and password.

**Endpoint**: GET <http://localhost:8090/user/api/v1/details>

**Request**: not required any addition data to send because user id can be got from authentication.



**Authorization**: user id and password



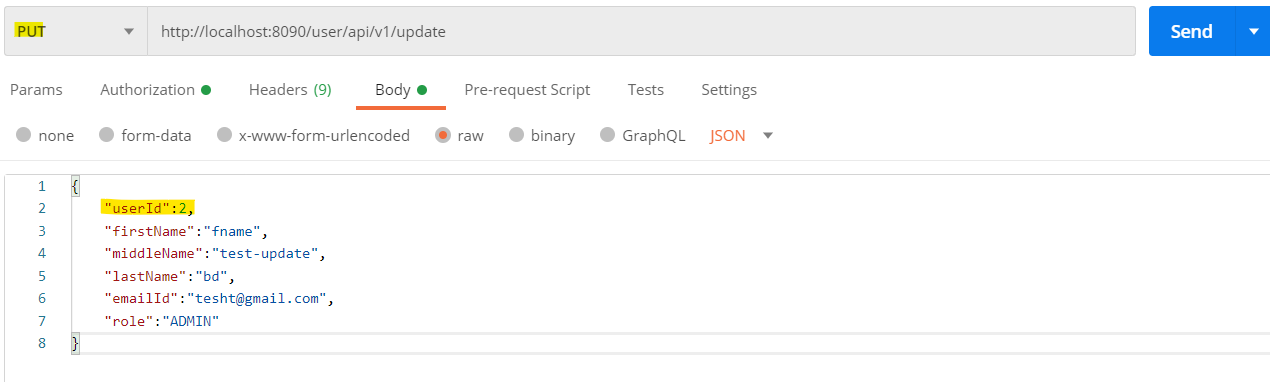
**Response**: user details  
  
**4)update-user**

This API is used to update any user details. Only ADMIN user can access this API.

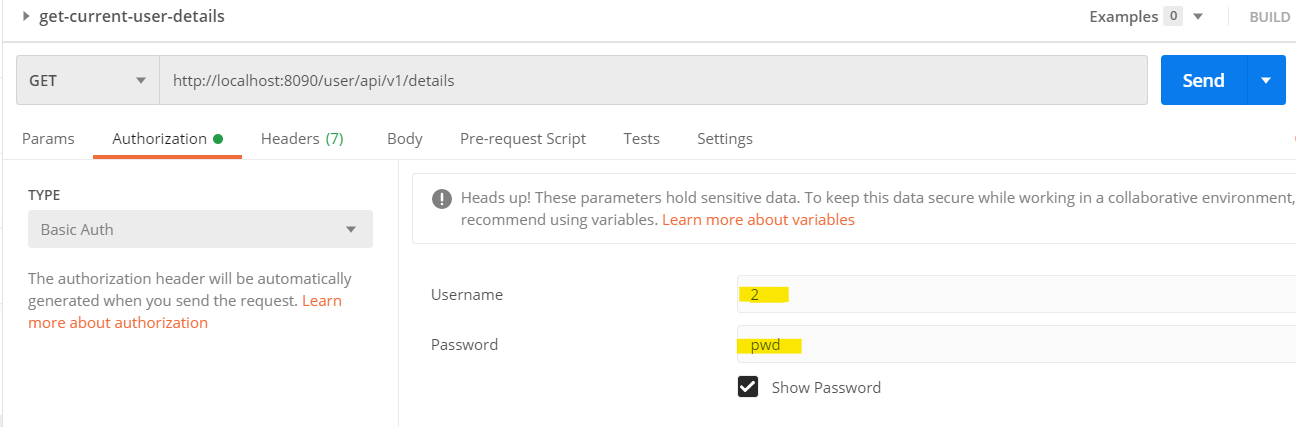
**Endpoint**: PUT: <http://localhost:8090/user/api/v1/update>

**Request**: JSON object with user details.

Note:- update will happen for the given user id.



**Authorization**: Admin user id and password



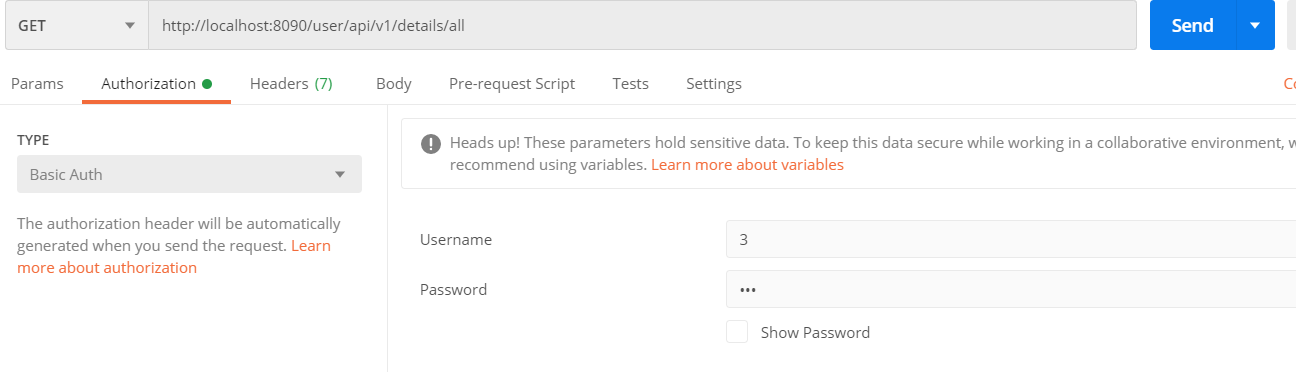
**Response**: user details

**5)get-all-user-details**

This API is used to get all user details. Only ADMIN can access this endpoint.

**Endpoint**: GET : <http://localhost:8090/user/api/v1/details/all>

**Request**:- Not required any additional data.  
Authorization:- Admin user id and password

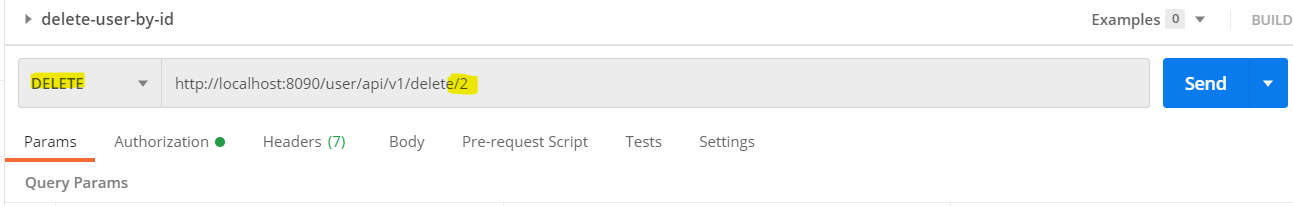
  
  
**Response**: List of all user details

**6)delete-by-user-id**

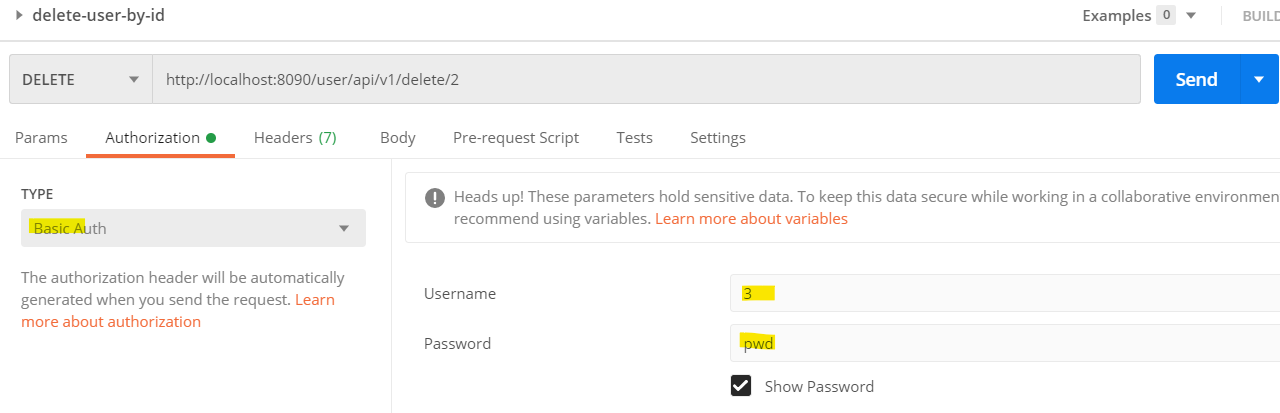
This API is used to delete the user by id. Only ADMIN can access this API.

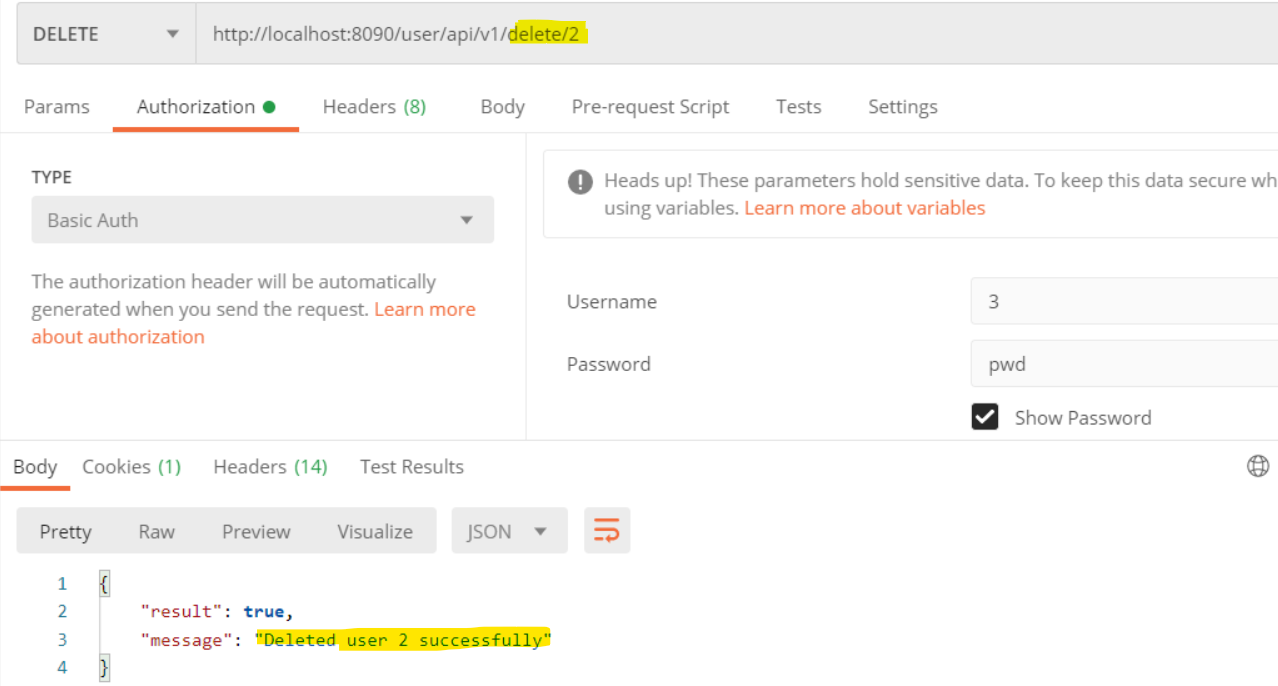
**Endpoint**: DELETE: <http://localhost:8090/user/api/v1/delete/2>

**Request**: user id in URL(path variable)  
Here 2 is the user id for deletion.



**Authorization**: Admin user id and password

  
  
**Response**: Delete successful message



**2)journal-service application**

This application consume message from **“user-event”** kafka topic and save those details into database, and it has API to get all events.

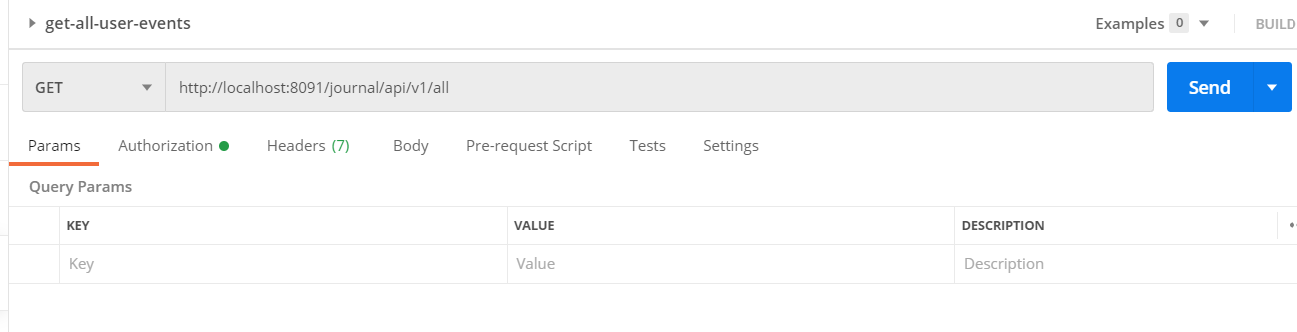
GitHub link for source code: - [GitHub - Sadvik-gowda-BD/journal-service](https://github.com/Sadvik-gowda-BD/journal-service)

**Api endpoints:-**

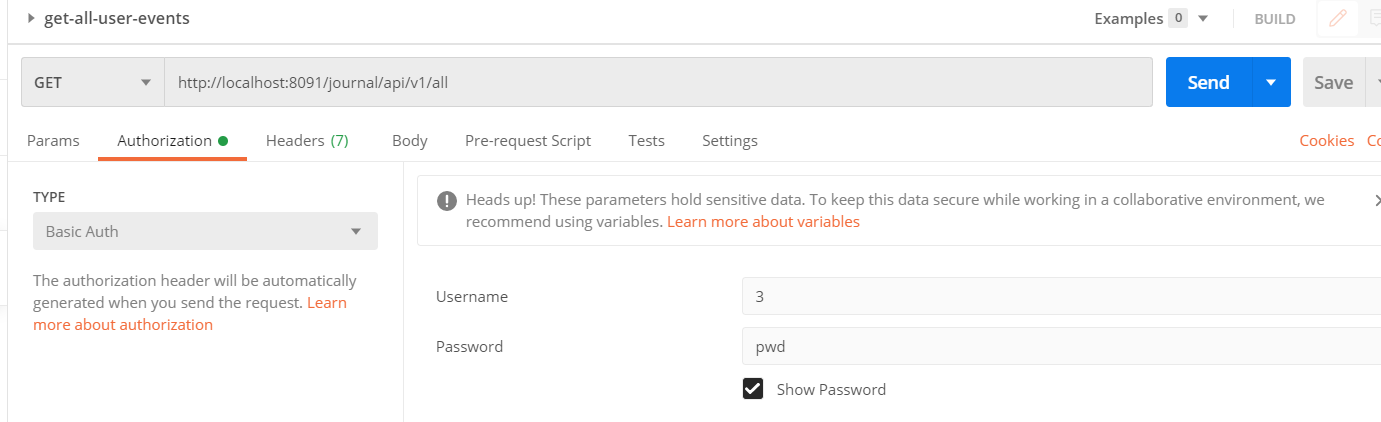
1)get-all-user-events

**Endpoint**: GET: <http://localhost:8091/journal/api/v1/all>

**Request**: Not require any additional details.



**Authorization**: Admin user id and password



**Response**:All user events.  
Sample response object.  
{

        "eventId": 9, // event id

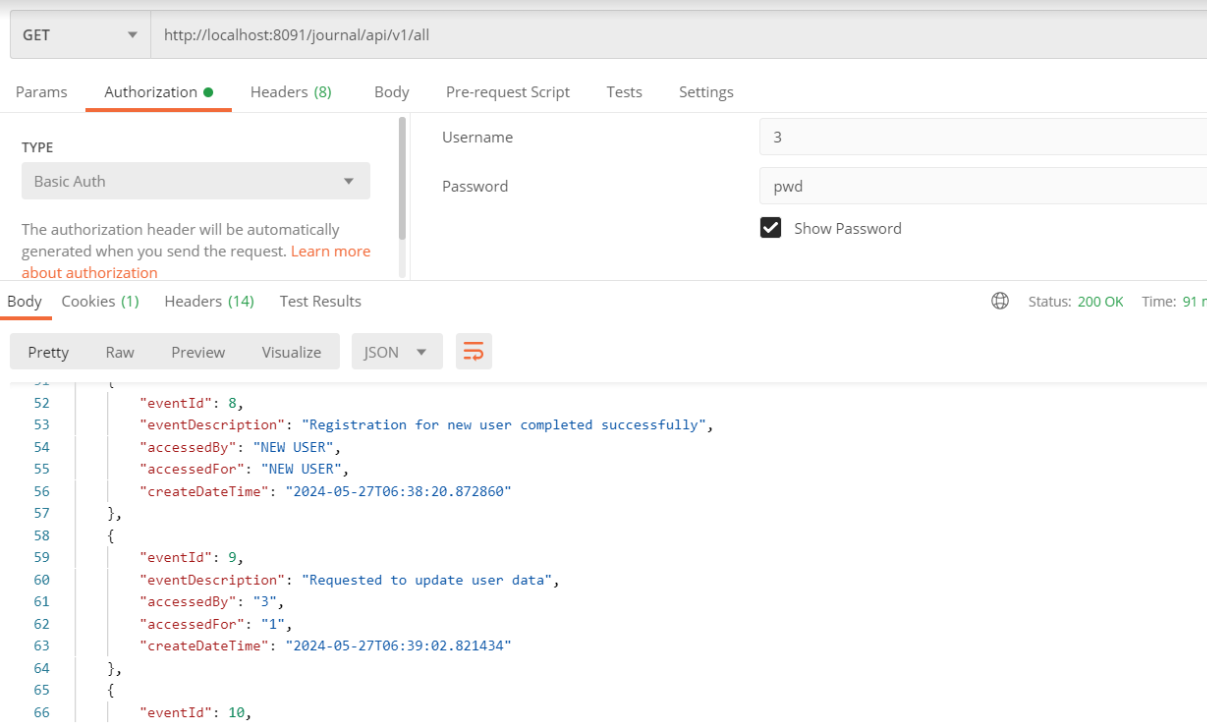
        "eventDescription": "Requested to update user data", // event description

        "accessedBy": "3", //user id of the accessed by user

        "accessedFor": "1", //user id of the accessed for user

        "createDateTime": "2024-05-27T06:39:02.821434" //Event create date time

}



Contact number:

Please call to following mobile number for any doubts: +91 9741471445

Email-id: sadvikgowda3333@gmail.com