**Todo App Using Socket.io**

* Download the project file
* Run **npm install** to install the project dependencies. If you encounter issues with peer dependencies, use **npm install --legacy-peer-deps.**
* Use **npm start** to run the project.
* Import the provided Postman collection**(root/postman collection)** into Postman and execute the API requests.
* Add env file

.env

PORT = 8000

DATABASE=todo\_list

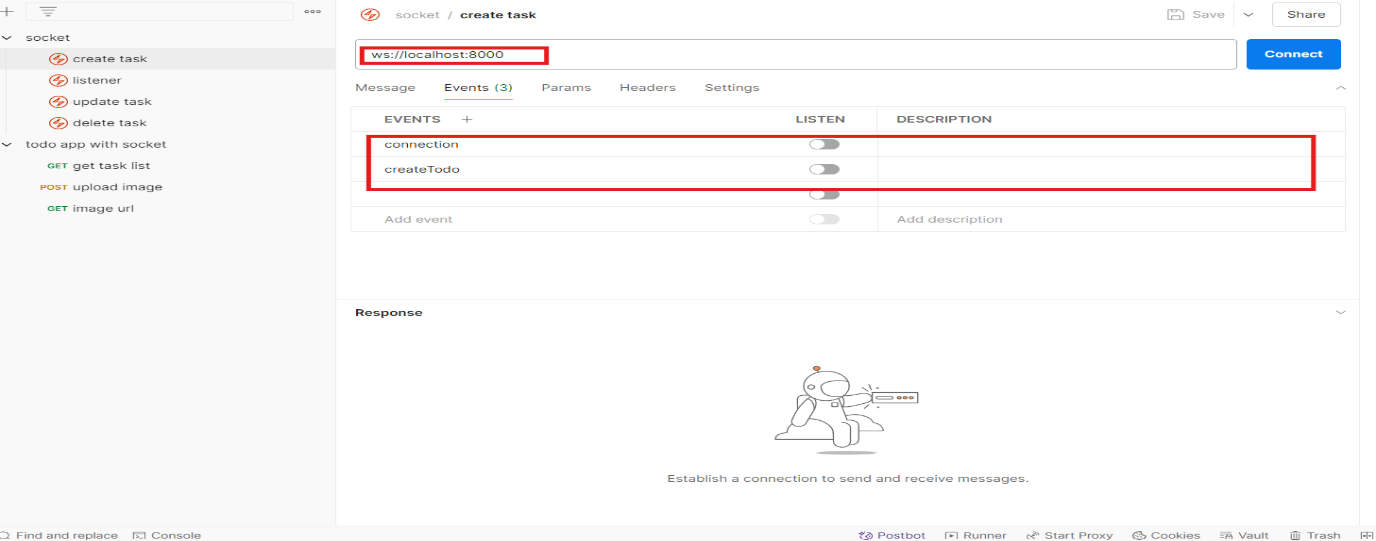
DATABASE\_USERNAME=root

DATABASE\_PASSWORD=

DATABASE\_HOST=localhost

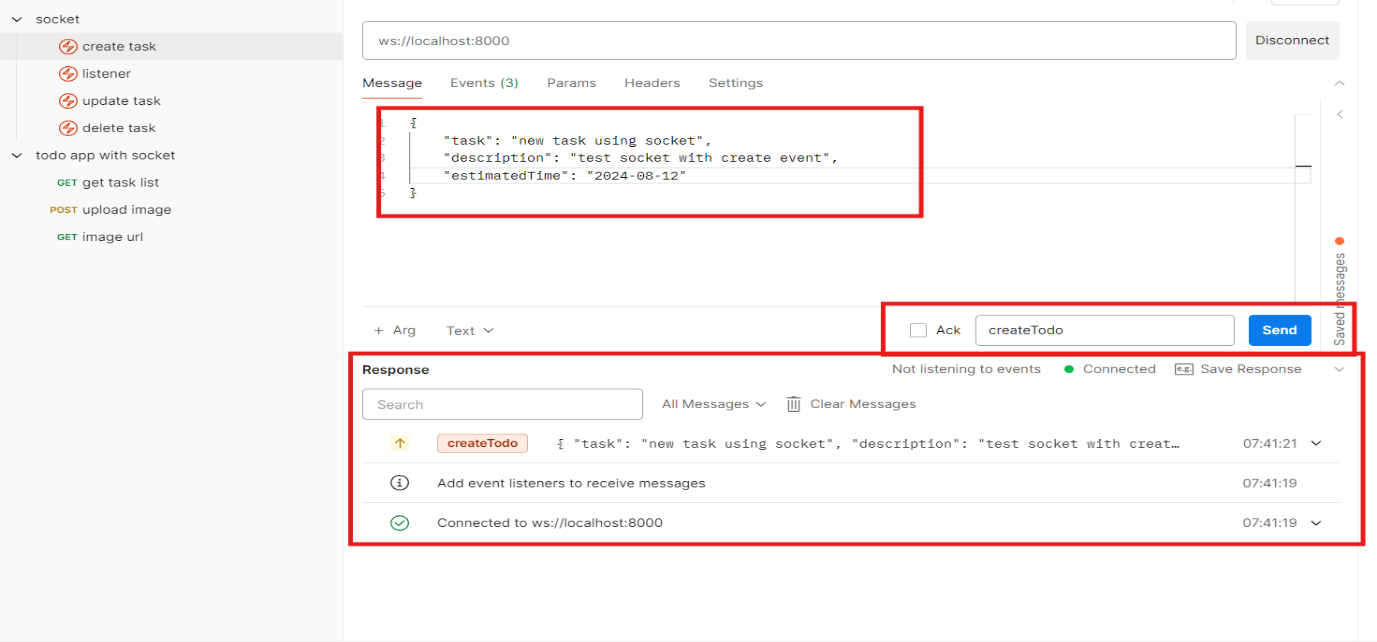
**Connect to a socket using Postman and create CRUD operations.**

***Add task***



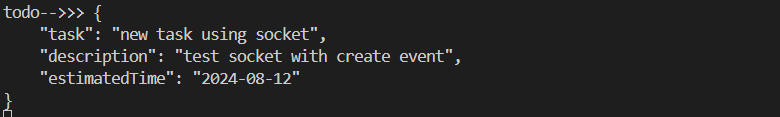
Socket url - ws://localhost:8000

Add events  - **connection, createTodo**



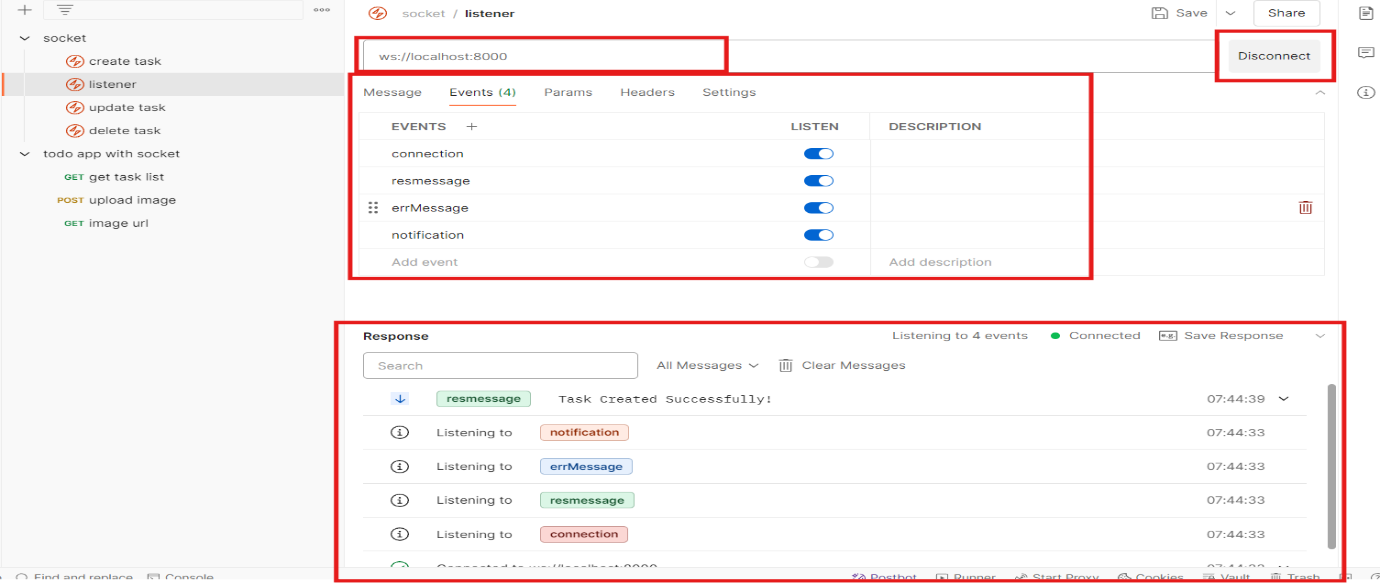
Add a task as a **object** in message field, enter the event name and hit the send message will emit and receive in server handler

**Receive Events in socket handler**

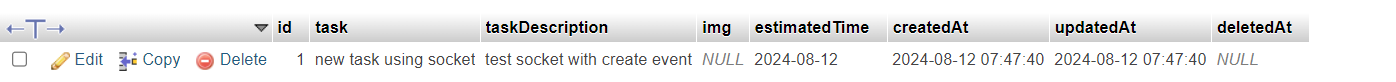


**Create Lister for get the result/error**

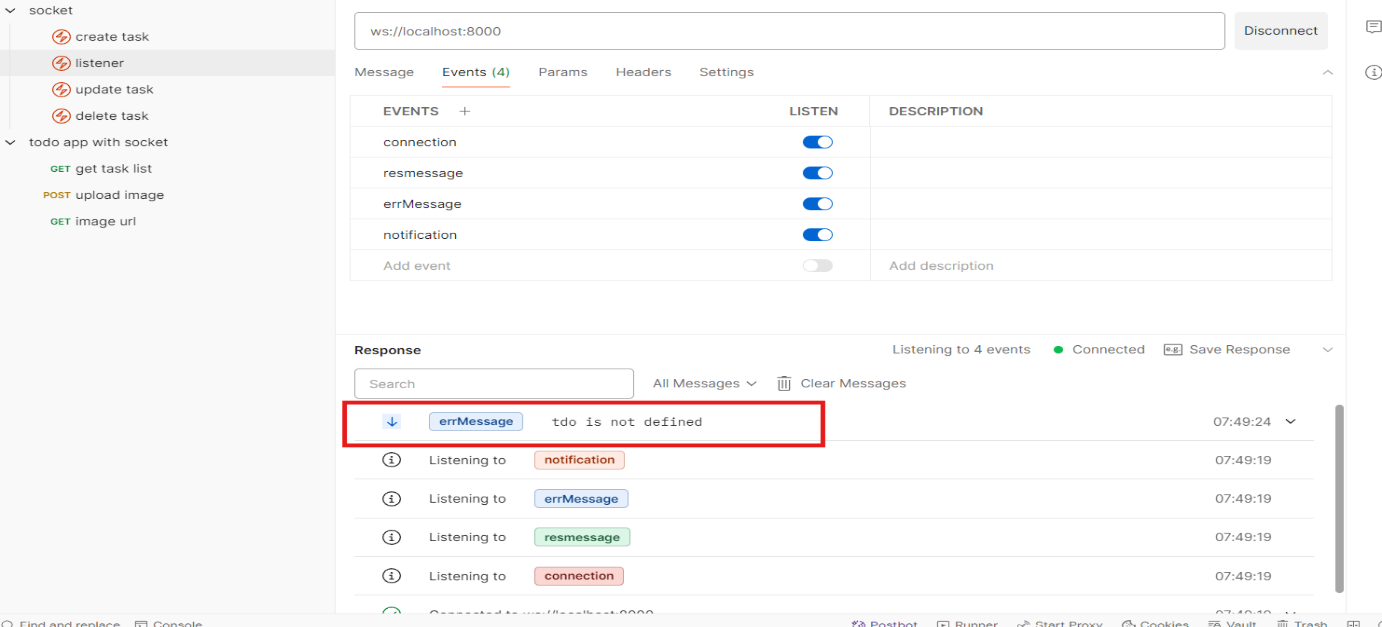
**Add listener events** in event tab and active the listen toggle button , it will receive all the emitted messages



New record **added** in Database

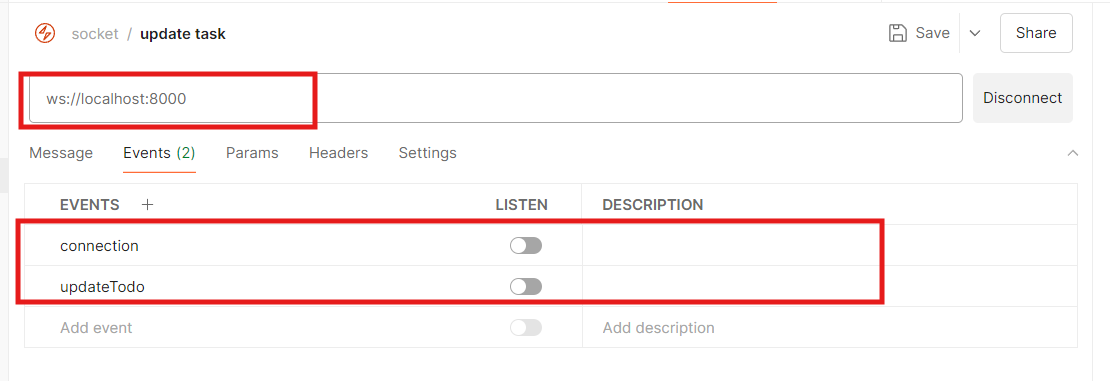


**Error Handling using socket**



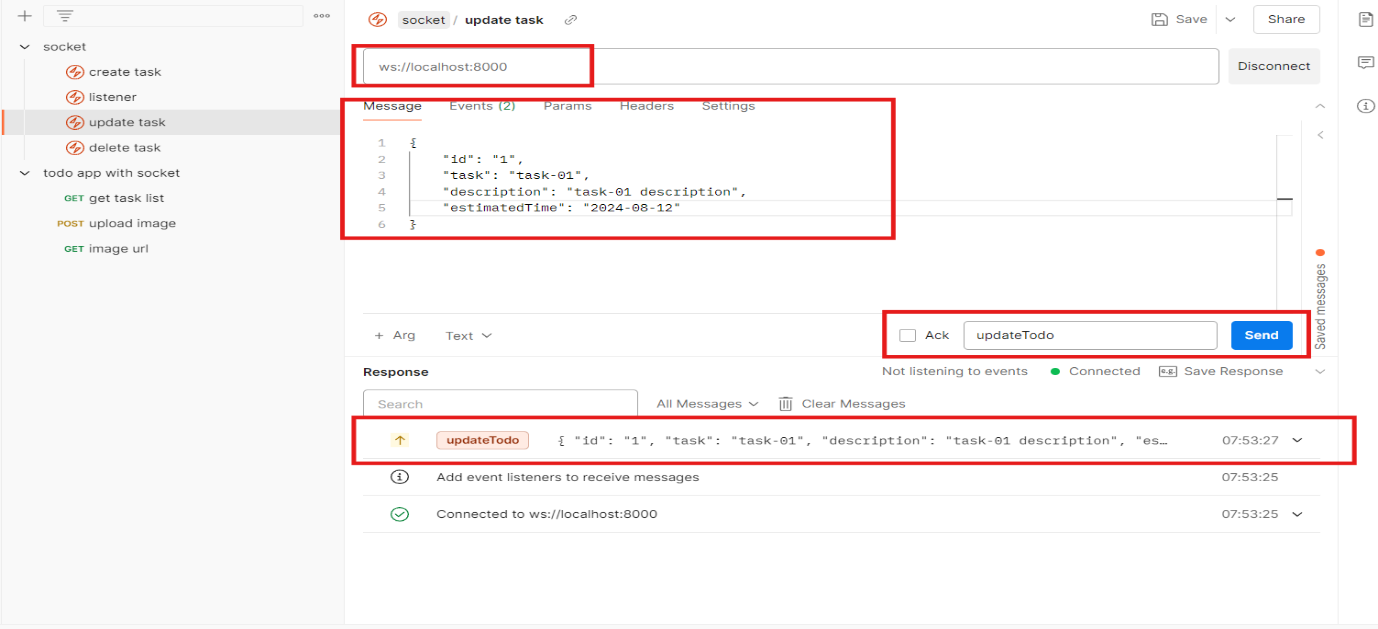
**errMessage event catch all the errors** and send it to the client

***Update Task***



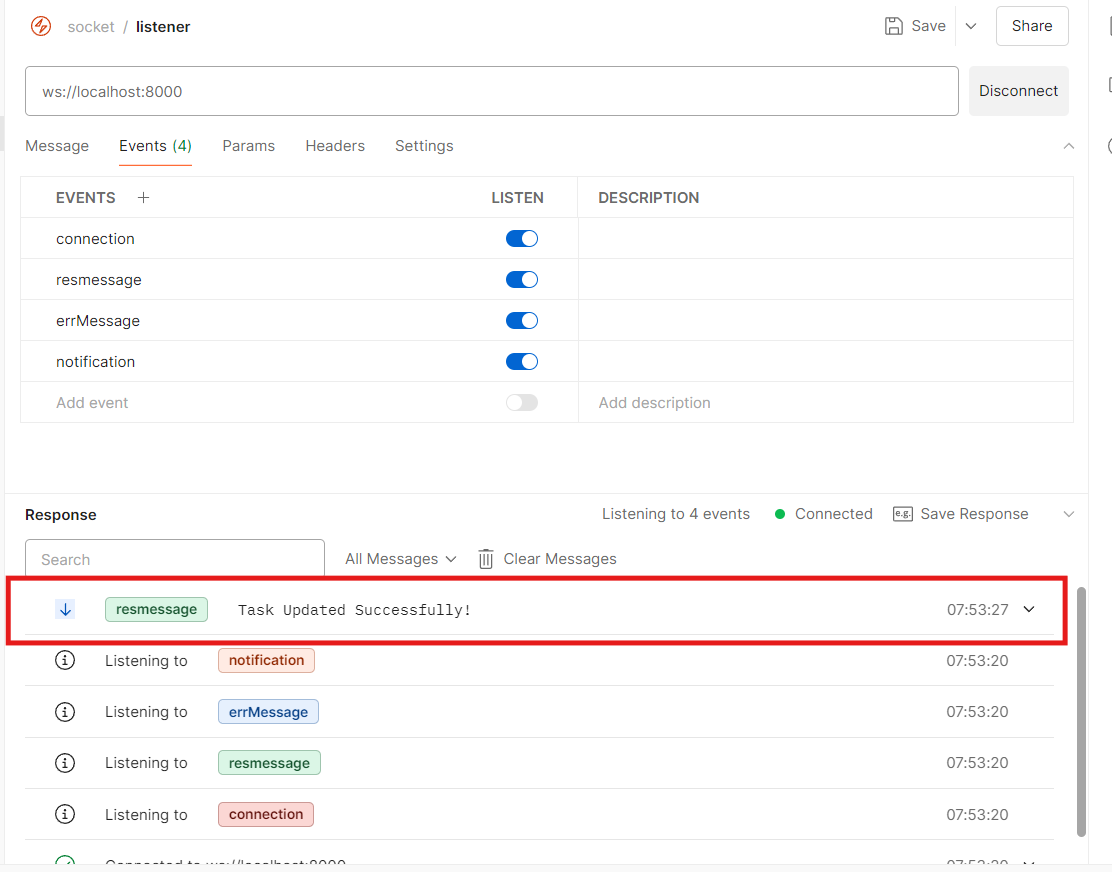
Socket url - ws://localhost:8000

Add events  - **connection, updateTodo**

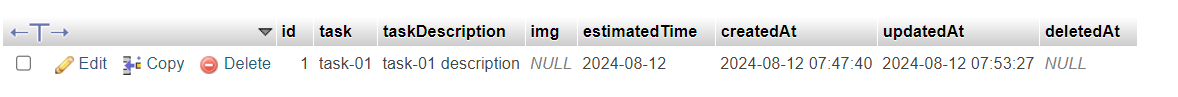


Add a task as a **object** in message field, enter the event name and hit the send message will emit and receive in server handler

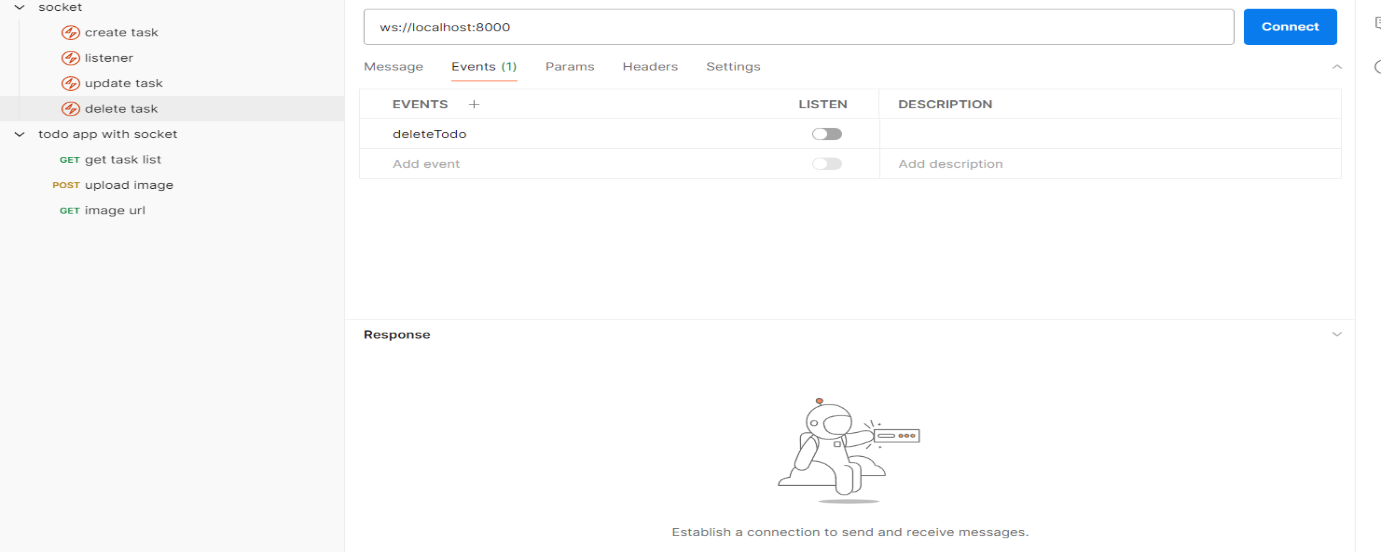
Update task event response



One task was updated

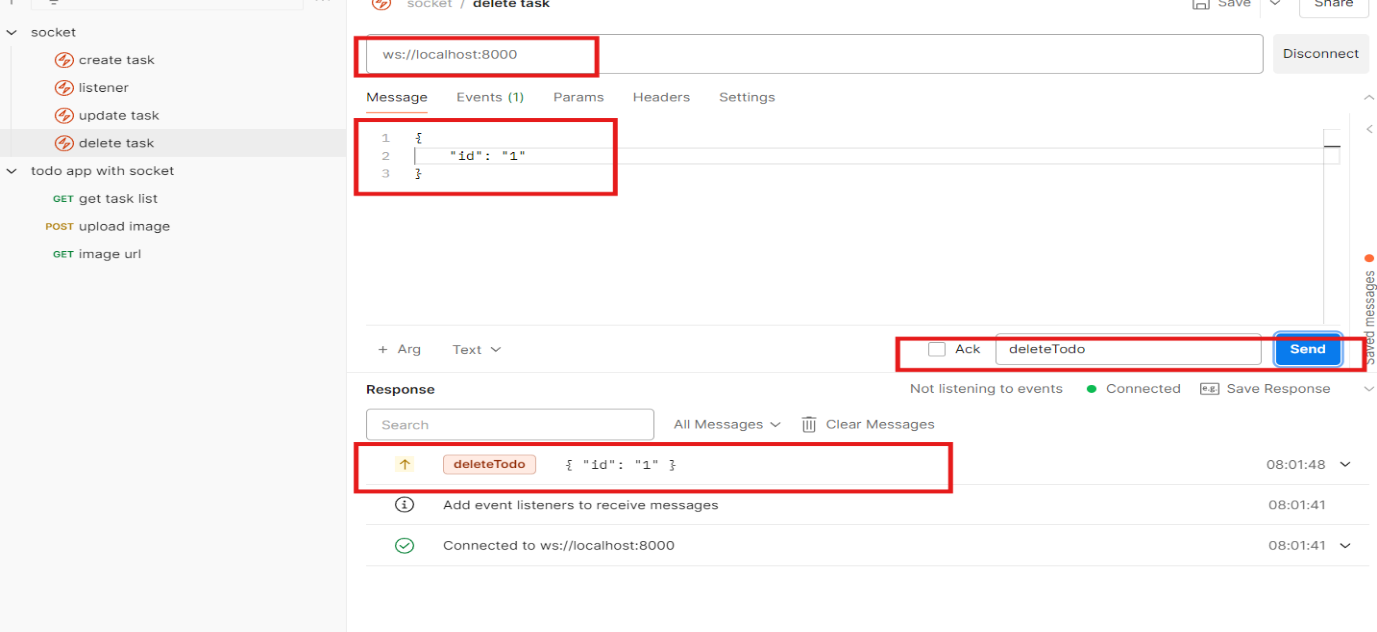


***Delete Task***



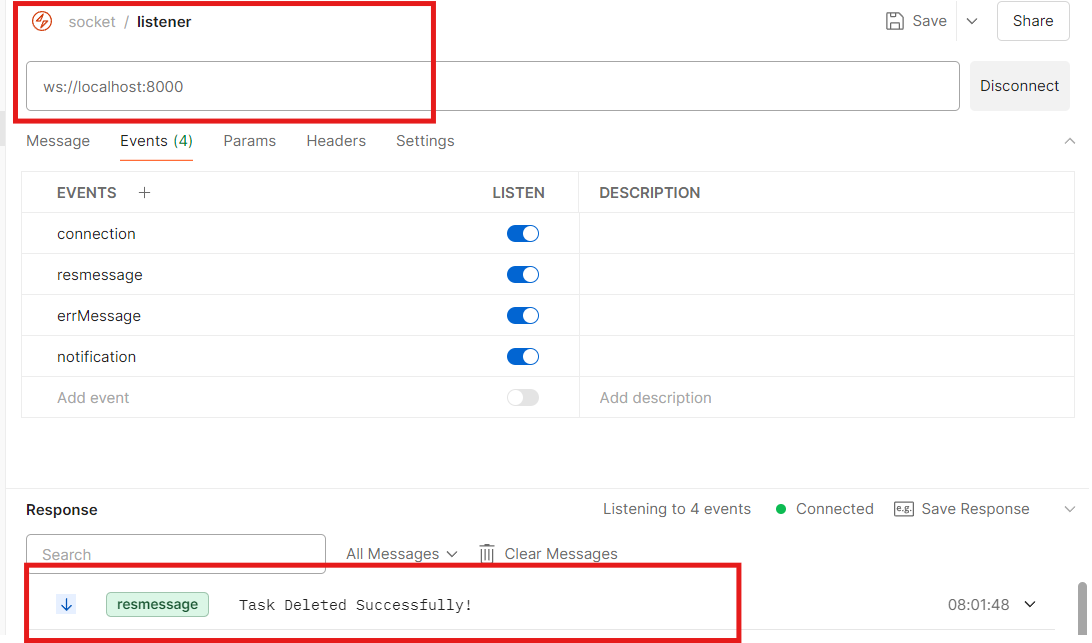
Socket url - ws://localhost:8000

Add events  - **connection, deleteTodo**

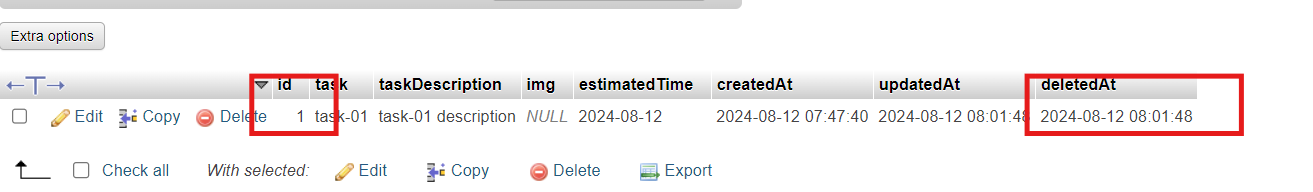


Add a task as a **object** in message field, enter the event name and hit the send message will emit and receive in server handler

Socket listener catch the response



One record was deleted



***Task List API***

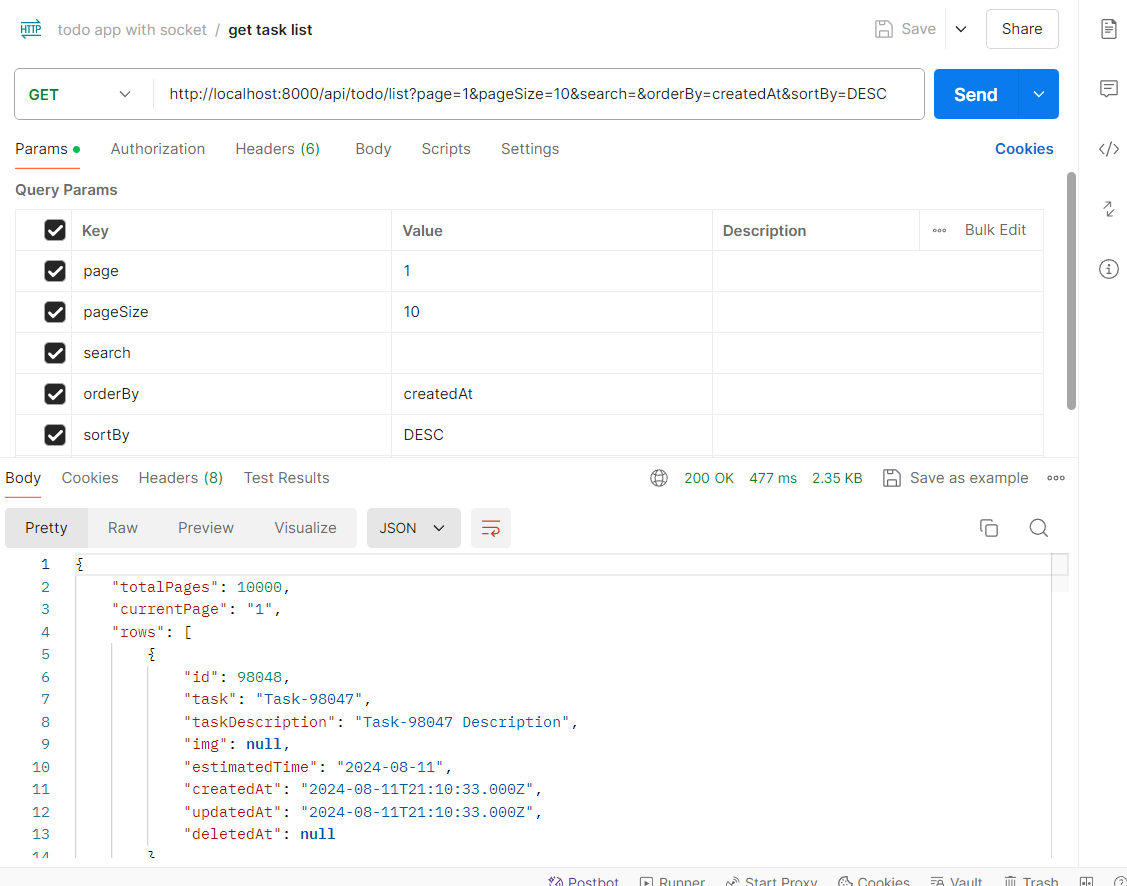
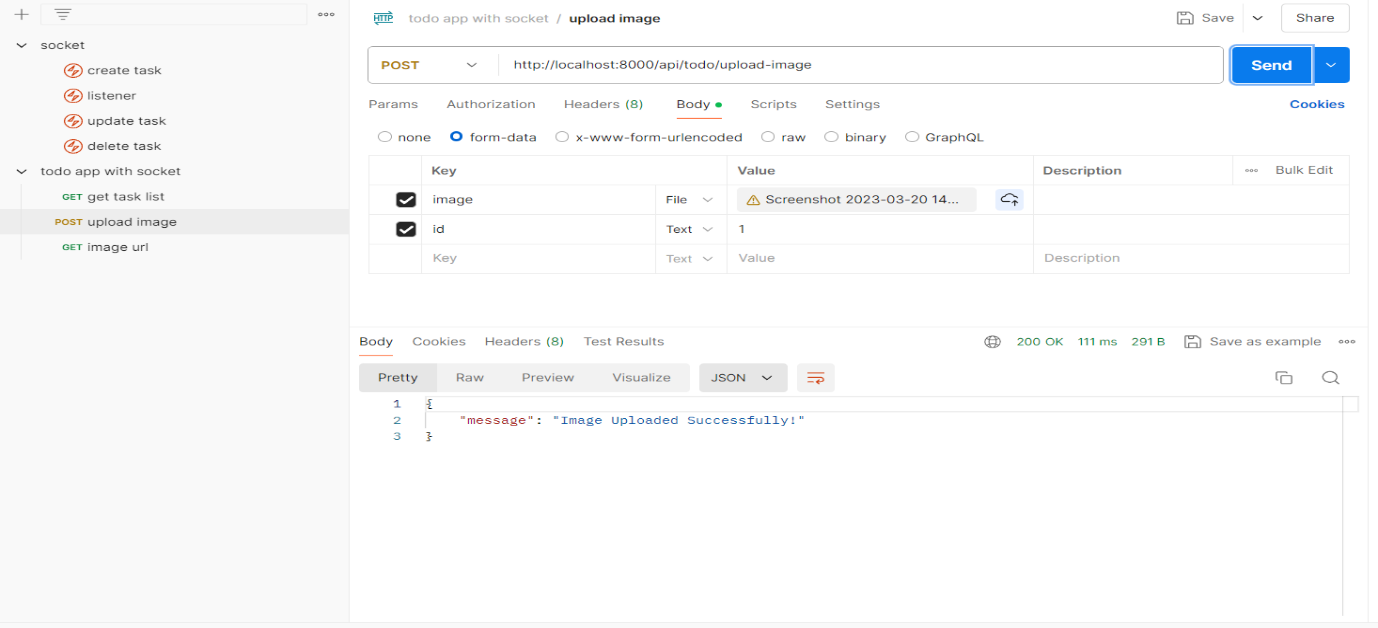


Image uploader

Use **form data** to upload image

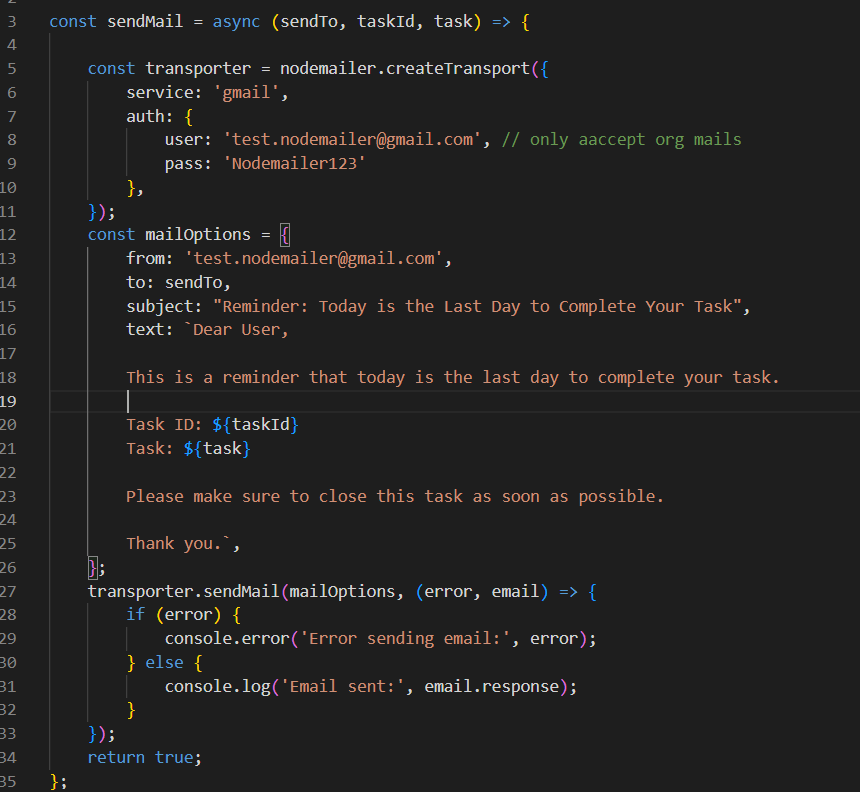


Sample image



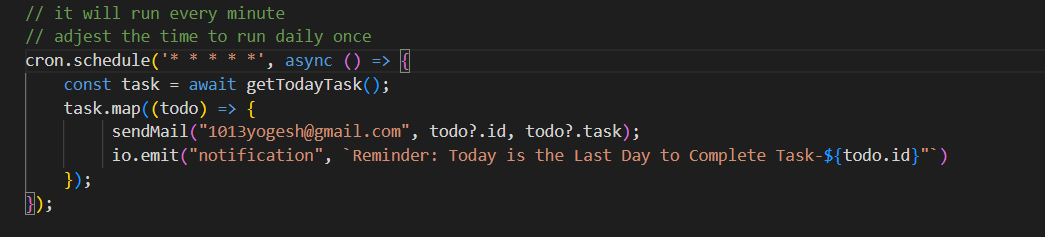
***Remainder Mail/Notification***

Send **remainder** using **Socket** & **Nodemailer**

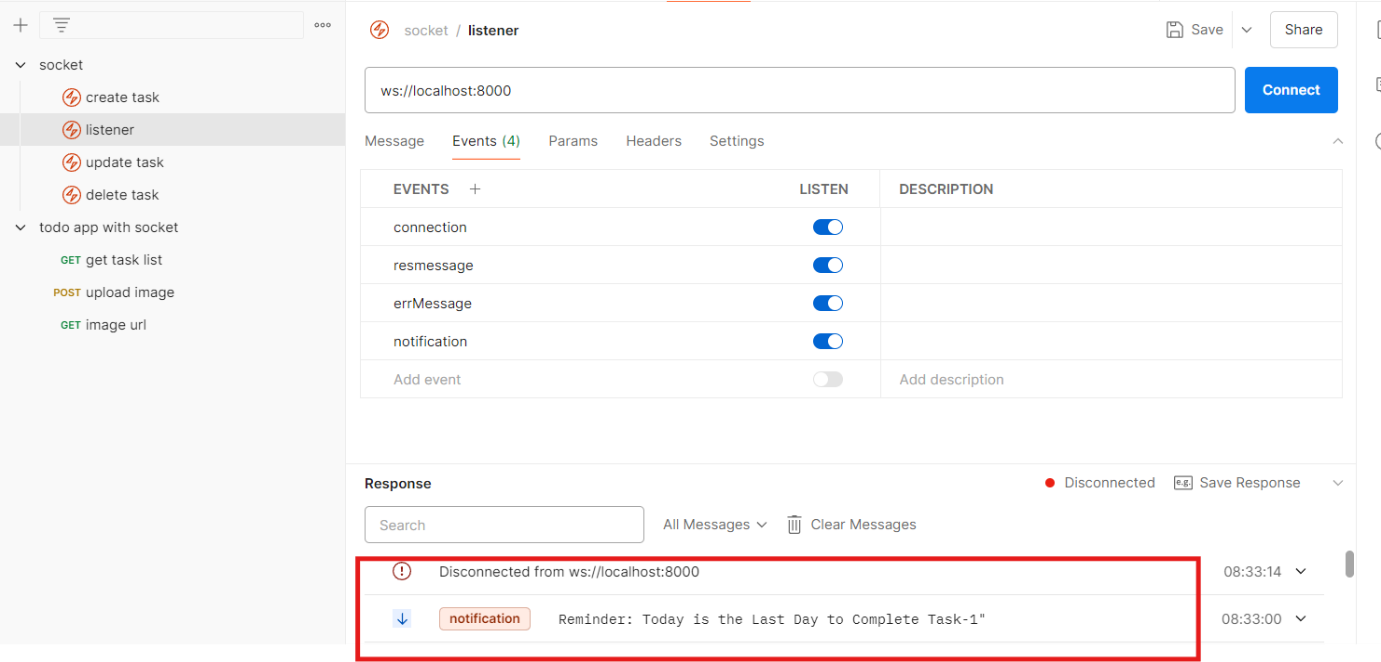


Replace the auth to you **organisation mail** and enable **permission** in you email

***Cron Job***



Add **cron job** to send mail and socket notification



Socket listener receive the notifications

***Create Sample Records(100000 records)***

Create 100000 test records using seed (batch process)

Cmd :- npx sequelize-cli db:seed:all

