**REST API for CRUD TODO list application:**

**Task:**

**Functional Requirement:**

Pointy-haired-boss wants to be more organized.  So, he wants a web application where he can

login and get all the TODOs he has created.  Pointy-haired-boss also wants to be able to

update and delete the TODOs.

The frontend will be built by someone else.

Technical Requirement

Please create the following:

- REST APIs

  - API to authenticate

  - APIs for TODO CRUD

- Protect the API using JWT

- Persist TODOs

- Must have unit test cases

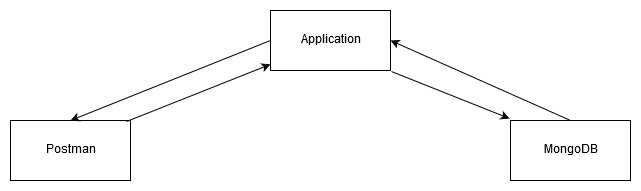
- Must have swagger documentation using Open API 3.0

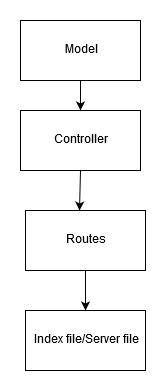
**Solution**

**Tools Used**

1. Node JS
2. Mongo DB
3. JSON Web Token
4. Postman
5. Robo 3T

**Application Workflow**





Interdependency of the Files.

**Instructions**

1. Clone the repository.
2. Run “npm install” to install all the dependencies from the terminal.
3. Run “npm start” to start the server/application.

**Demo**

1. Register the login credentials (username, email and password).

A screenshot of a cell phone

Description automatically generated

1. After registering, login with the same credentials. You will receive the JWT Token. This token is important, and it will used for all the requests in the future

A screenshot of a social media post

Description automatically generated

If you enter the wrong credentials, then it would show an error as follows.

A screenshot of a social media post

Description automatically generated

1. We will now create a few tasks.

A screenshot of a social media post

Description automatically generated

Observe that we must send the JWT token as the header.

A screenshot of a cell phone

Description automatically generated

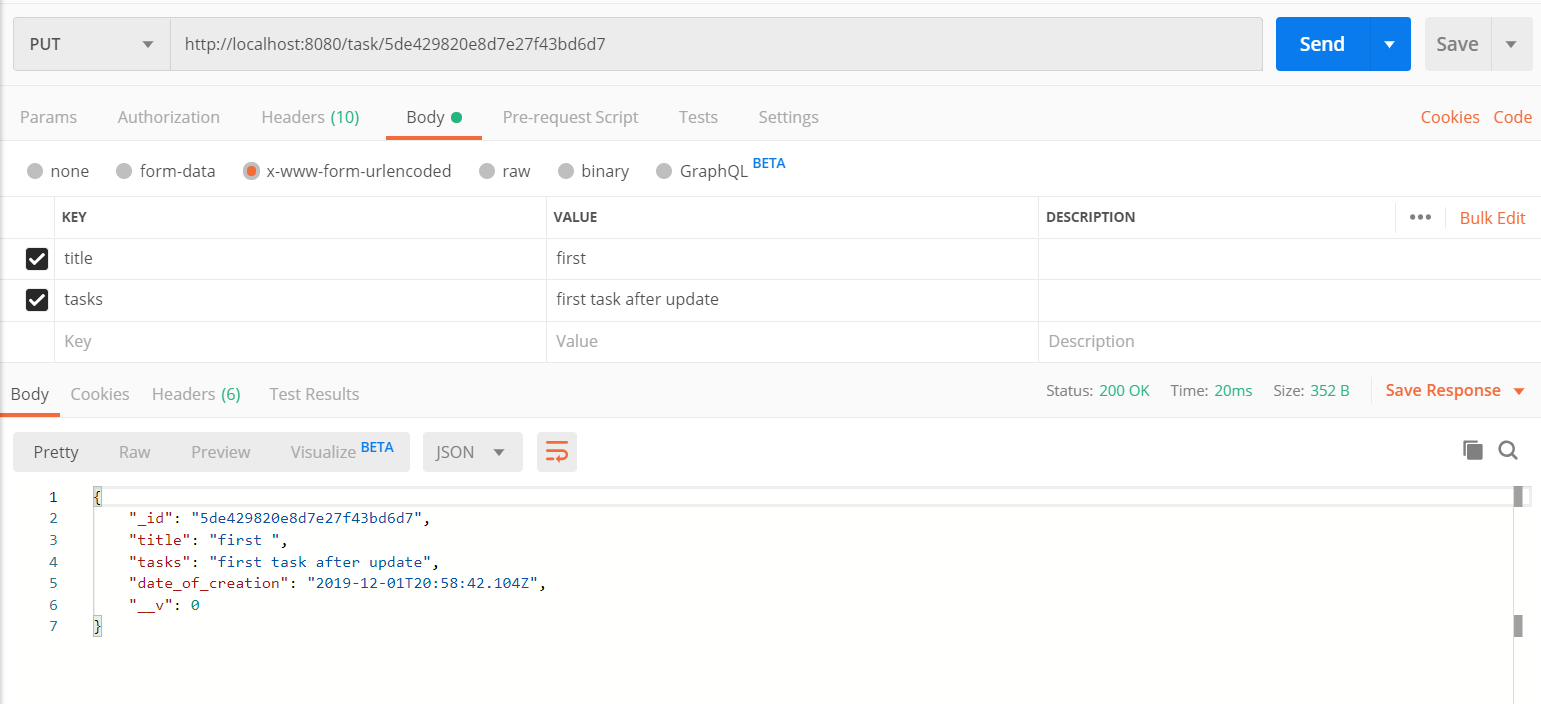
1. We will now read all the to-do lists.

A screenshot of a social media post

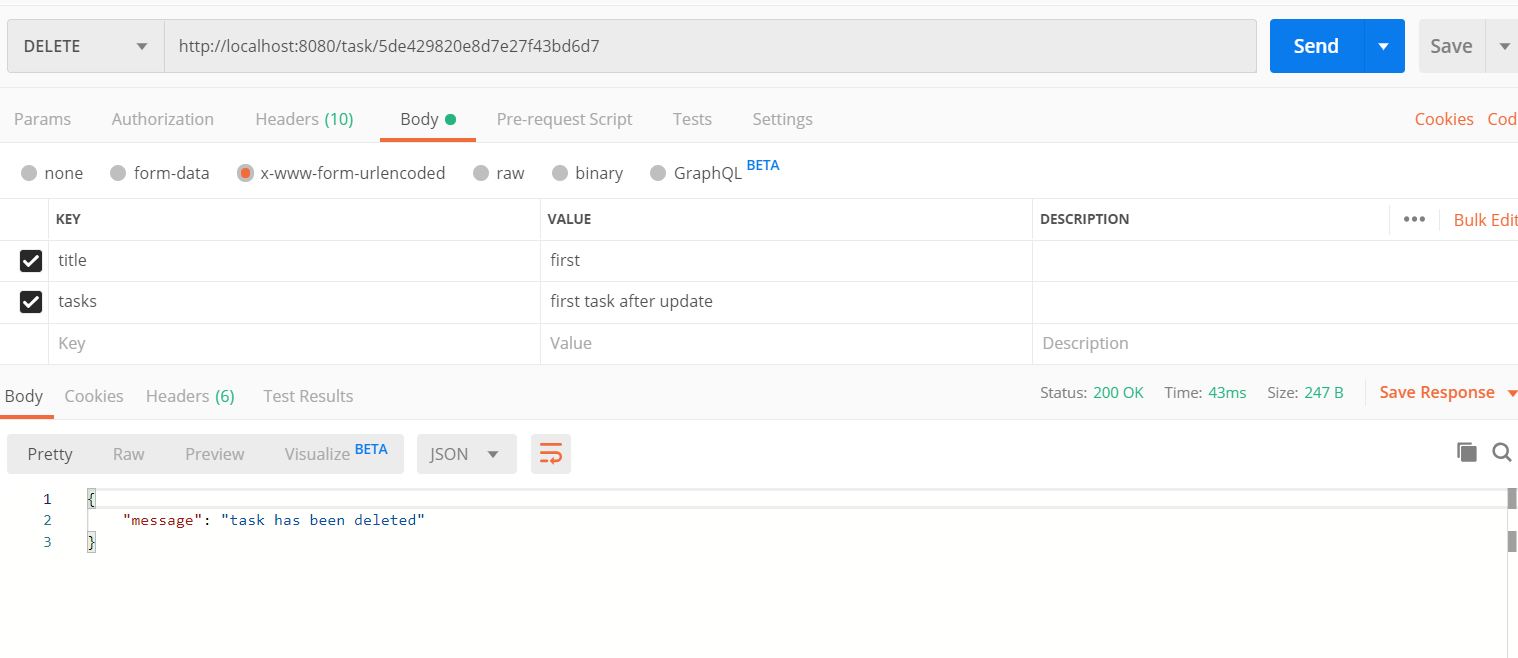
Description automatically generated

1. We will now read the first to-do list. We can read it by calling a GET request and addressing the id.
2. We dsdsA screenshot of a social media post

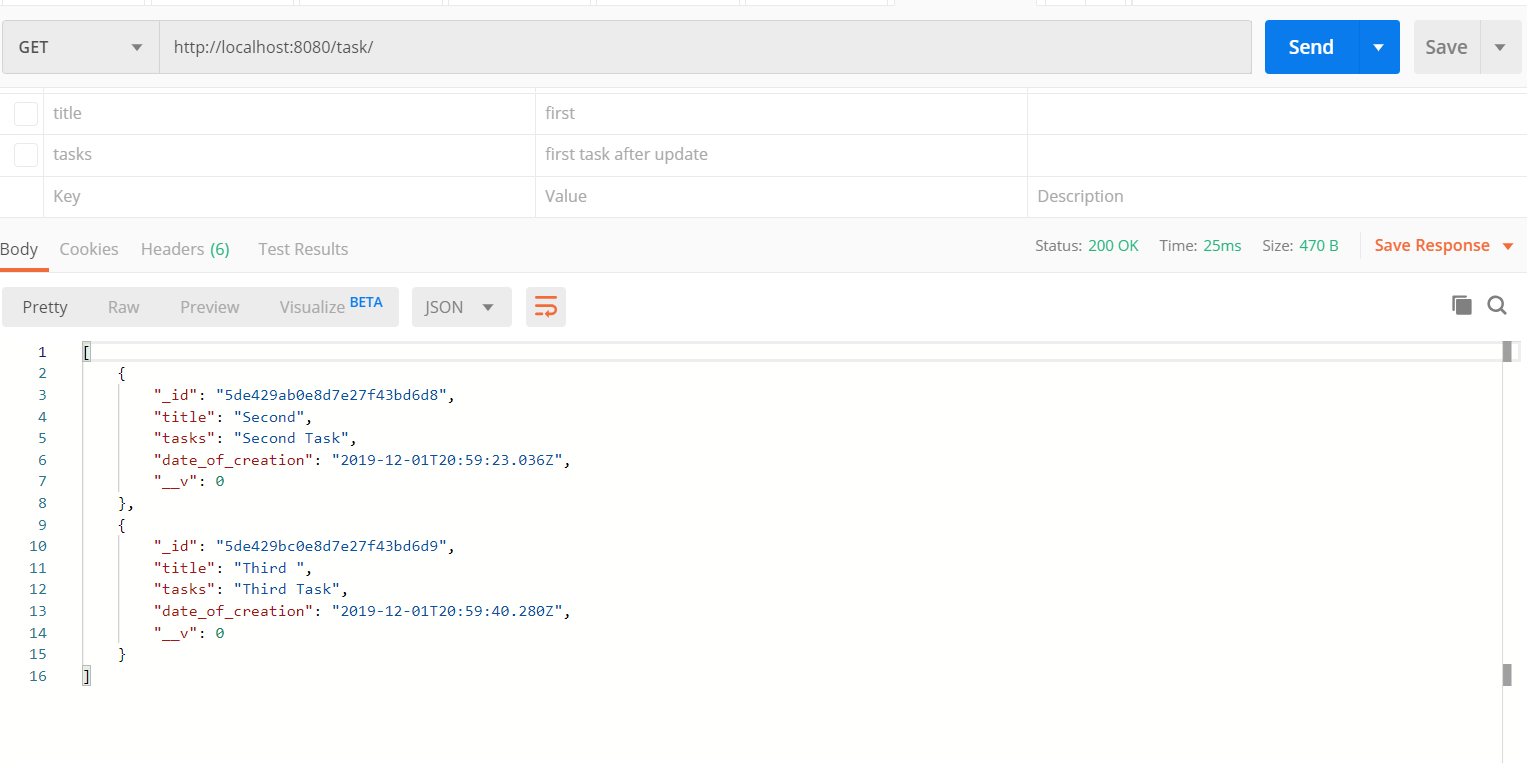
   Description automatically generated
3. We will now update a task.



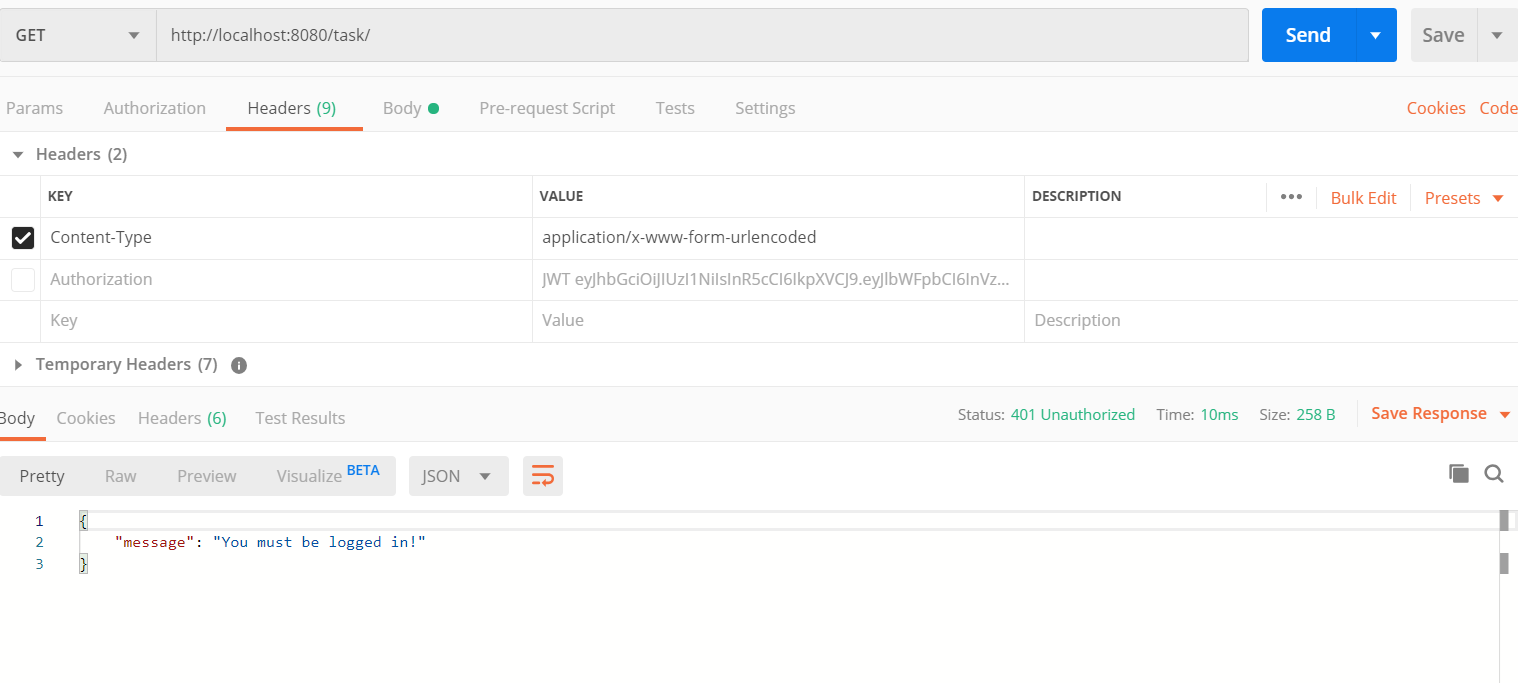
1. We will now delete the first task.



When we now retrieve all the tasks, you will see that the first task is deleted.



1. We have now completed all the operations. Let us now see what happens when we do not add the JWT token.



Thus, the API is also secure from any threats of remote login.

1. Following are the database schema explored with Robo 3T.

