

Assignment 1

In this assignment, you will create a REST API using Node.js and Express. You do not need to provide unit tests, but you should specify each API request in a `request.rest` file.

The required API should include the following endpoints:

1. **Add a New Post:** Allows a user to add a new post to the database.
2. **Get All Posts:** Returns all posts in the database as a JSON array.
3. **Get a Post by ID:** Returns the post with the specified ID. The request URL format is:
`/post/<post_id>`
4. **Get Posts by Sender:** Returns all posts published by a specific sender, identified by sender ID. The request URL format is:
`/post?sender=<sender_id>`
5. **Update a Post:** Updates a post with new data, replacing its current content. This operation uses the PUT method, where the post ID is specified in the URL, and the new post data is included in the JSON body of the request.

Additionally, implement a full CRUD API for **Comments**. Comments are messages written by users on specific posts. The Comments API should include endpoints to create, read, update, and delete comments, as well as an endpoint to retrieve all comments associated with a specific post.

Collaboration Requirements:

- Each student is responsible for implementing **half of the API endpoints**.
- Work collaboratively using **Git** with **pull requests** to integrate your work. Each student should submit their changes via pull requests, allowing for review and integration with the team's codebase.

Note: Ensure each API request is clearly defined in the `rest.rest` file, outlining the request methods, URLs, and data formats used.

יש להתחבר למנג'ו כמו שעשינו בכיתה.

חובה לבצע את התרגיל בזוג, על כל אחד מהפתחים להוסיף מחצייה מה- API הנדרשים ומהחצייה מהטסיטים הנדרשים - את הטסיטים יש לבצע באמצעות rest request באופן ידני כמו שעשינו בכיתה. יש ליצור branch ו-pull request עבור כל אחד מה- API. הסטודנט השני נדרש לאשר את ה- pull request לפני ה- merge.

יש להגיש קובץ אחד ובתוכו:

1. קישור לגיט עם הקוד של הפרויקט (חייב להיות ב- public).
2. סרטן קצר המדגים את המערכת רצה.

3. צלום מסך ה- Network Graph מתוך הגיט כך שניתן לראות גם את שם בעל חשבון הגיט בהתאם לדוגמה הבאה:

The screenshot shows the GitHub Insights interface for the repository `menachi / app_a`. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Security, Insights (which is highlighted), and Settings. A search bar and various icons are also present. On the left, a sidebar lists metrics: Pulse, Contributors, Community, Community Standards, Traffic, Commits, Code frequency, Dependency graph, Network (which is selected and highlighted in orange), Forks, Actions Usage Metrics, and Actions Performance Metrics. The main area is titled "Network graph" and contains a sub-header: "Timeline of the most recent commits to this repository and its network ordered by most recently pushed to." Below this is a table with two columns: "Owners" and "Nov". The "Owners" column lists "menachi". The "Nov" column shows a timeline of commits. One commit from "menachi" is shown with a green arrow pointing to another commit from "mernachim". Other commits from "menachi" are shown with blue arrows pointing to them. The timeline ends with a commit from "mernachim" at the bottom right.

בhasilha