## **Author Information**

**Name:** Luis Alejandro Reyes Morales  
**Role:** Senior Software Engineer  
**Date:** 25/11/2024  
**Project:** Test-A-Safe API Development  
**Contact:** inglreyesm@gmail.com

**Task 7: Monorepo and Separate Packages (25 points)**

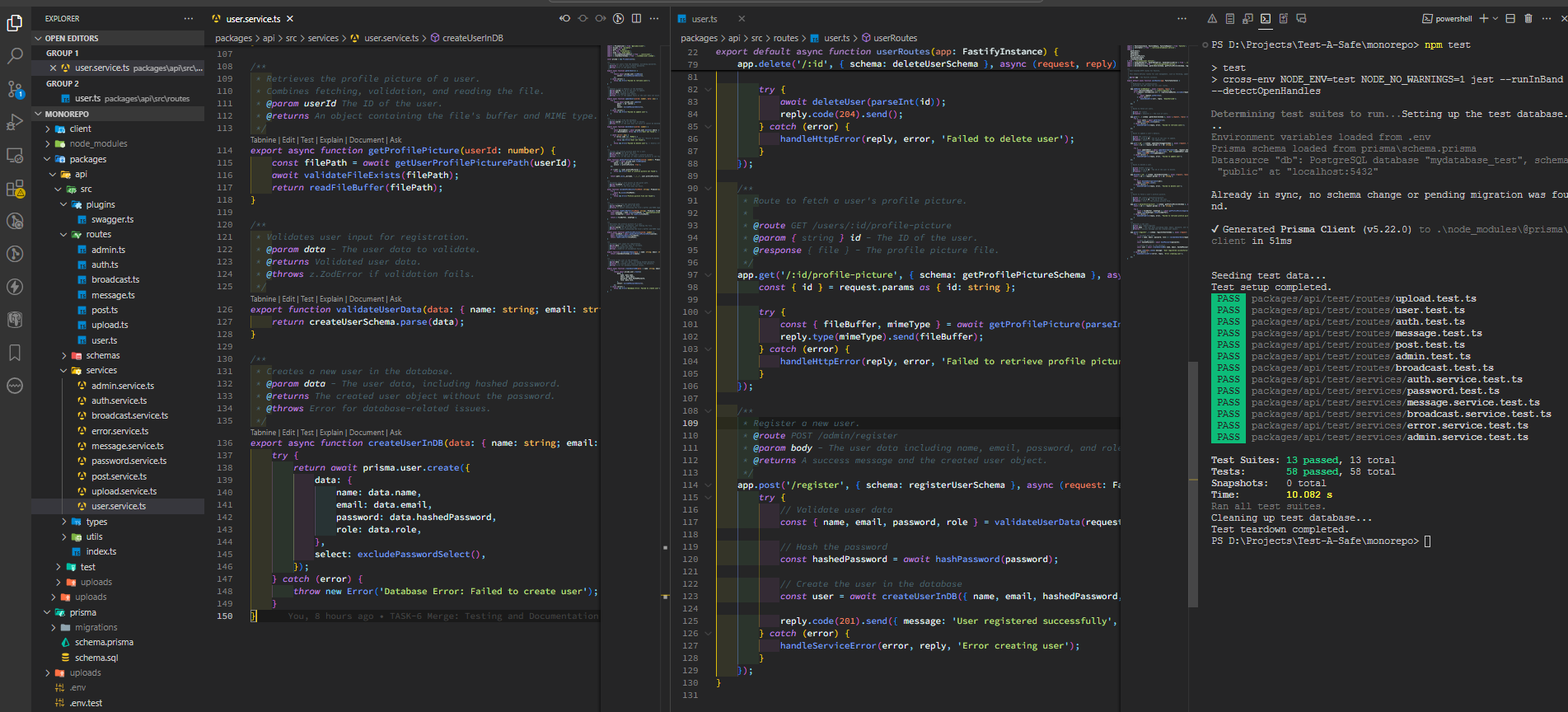
**[Optional]**

● Demonstrate how changes in one package affect others.

### Initial State: Tests Before Adding New Methods

The following figure shows the state of the tests before implementing the changes. These tests validate the existing functionality of the application and ensure that the previously implemented endpoints work correctly.

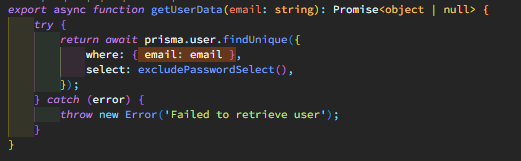
The main goal before starting the changes is to maintain this stable and functional state, ensuring that no regressions are introduced during the implementation of the new method and endpoint.



### Addition of the getUserData Method

The getUserData method has been added to encapsulate the logic for retrieving a user from the database using their email address as the key. This method is essential for the new endpoint as it:

* Centralizes the database interaction for user retrieval.
* Facilitates the reuse of this logic in other contexts or future endpoints.
* Improves code maintainability by separating business logic from route logic.

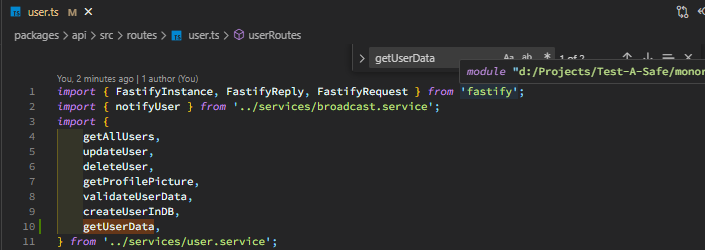


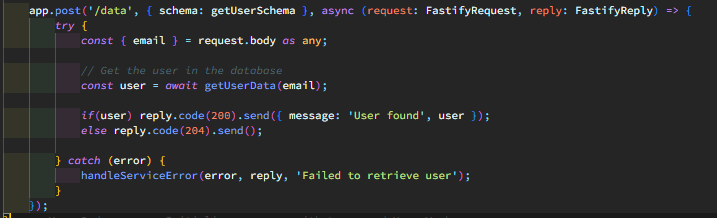
### Integration of the Method into the GET /users/data Route

The new getUserData method has been integrated into the user.ts file within the GET /users/data route. This route allows querying information about a specific user by providing an email address in the request body.

#### Route Process:

1. **Request Validation**:
   * Uses the getUserSchema to ensure the provided email is valid.
2. **Call to the getUserData Method**:
   * Passes the email to the method to search for the user in the database.
3. **Response**:
   * **200 OK**: If the user is found, their information is returned along with a success message.
   * **204 No Content**: If the user is not found, a message is returned indicating they do not exist.
   * **500 Internal Server Error**: If an error occurs during execution, an appropriate error message is returned.





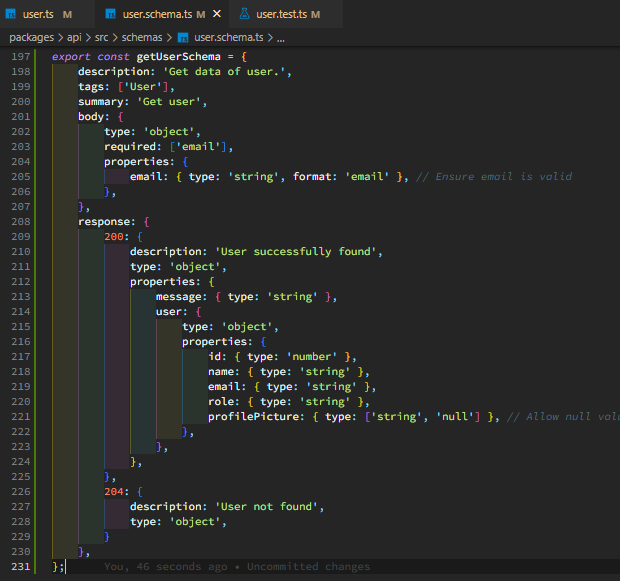
### Creation of the getUserSchema

The getUserSchema was created to validate the inputs and outputs of the new route. This schema ensures:

* That the email provided in the request body is required and has a valid format.
* That responses are consistent with the status codes and have the expected structure.

#### Schema Components:

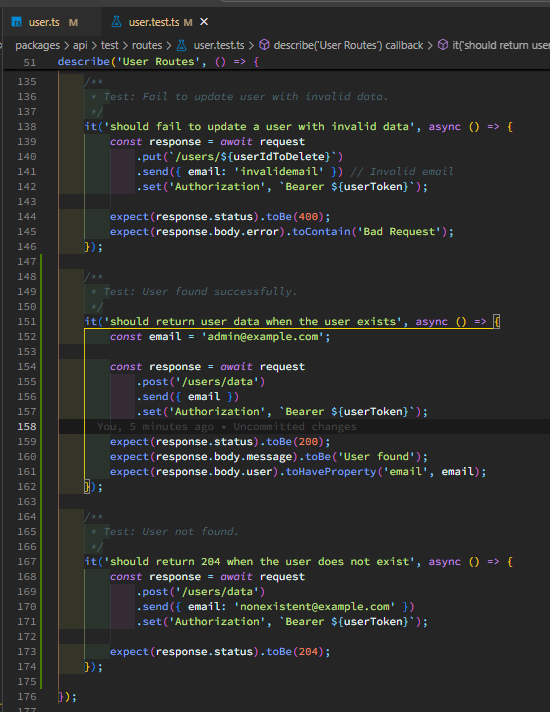
1. **Request Body (body)**:
   * Requires an email field with a valid format.
2. **Responses**:
   * **200 OK**: Returns the found user and a success message.
   * **204 No Content**: Indicates that the user was not found.
   * **500 Internal Server Error**: Provides a message in case of a server-side error.



**New Tests for the Endpoint**

Tests were added to cover the new GET /users/data endpoint, ensuring it works correctly in different scenarios. These tests include:

1. **User Found**:
   * Sends a valid email of an existing user.
   * Verifies that the response includes a 200 status code and the user data.
2. **User Not Found**:
   * Sends a valid email of a user that does not exist in the database.
   * Verifies that the response is a 204 status code with no additional data.
3. **Service Error**:
   * Simulates a failure in the getUserData method.
   * Verifies that the endpoint returns a 500 status code with a clear error message.
4. **Invalid Input**:
   * Sends an email in an incorrect format or an empty email.
   * Verifies that the response is a 400 status code, indicating that the request is invalid.



**Execution and Validation of the New Tests**

Finally, the new tests were executed to validate the behavior of the new endpoint. During this process:

* It was confirmed that all tests for the new endpoint passed successfully.
* It was verified that the existing tests were not affected, ensuring no regressions were introduced into the system.

In the following figure, the test results after implementing the changes are displayed. All tests, including the new ones and the existing ones, execute successfully, validating the success of the integration.

