Friends Management Assignment

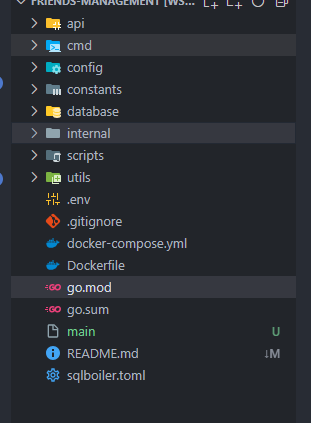
1. Introduction

A RESTful API for managing friend relationships, built using Go and PostgreSQL. This application allows users to add, manage, and interact with their friend lists.

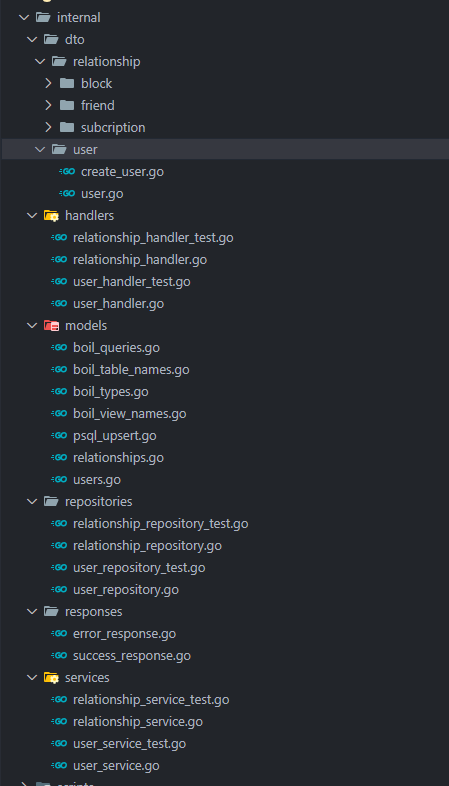
Features:

* Create a friend connection
* Retrieve the friends list for an email address
* Retrieve the common friend list between two email addresses
* Subscribe to updates from an email address
* Block updates from an email address
* Retrieve all updatable email addresses

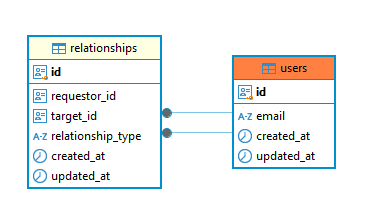
1. Project Structure
   1. Overall



* 1. Internal



1. Database Schema



1. Technologies

The following technologies are used in the provided Go project:

1. Error Handling:

- `github.com/friendsofgo/errors`: Error handling with stack trace support.

2. Routing:

- `github.com/go-chi/chi`: Lightweight, idiomatic routing in Go.

3. UUID Generation:

- `github.com/google/uuid`: For generating universally unique identifiers (UUIDs).

4. Database:

- `github.com/jackc/pgx`: PostgreSQL driver and toolkit for Go.

- `github.com/volatiletech/sqlboiler`: ORM (Object-Relational Mapping) tool for SQL databases.

- `github.com/volatiletech/strmangle`: SQL utility functions.

- `github.com/DATA-DOG/go-sqlmock`: SQL mocking for unit testing.

5. Environment Variables:

- `github.com/joho/godotenv`: Load environment variables from `.env` files.

6. Dependency Injection:

- `go.uber.org/fx`: A framework for building modular, scalable applications using dependency injection.

- `go.uber.org/dig`: A reflection-based dependency injection framework.

7. Logging:

- `go.uber.org/zap`: High-performance logging library.

8. Validation:

- `github.com/go-playground/validator`: Input validation framework for structs and fields.

- `github.com/go-playground/locales`: Language-specific locale information.

- `github.com/go-playground/universal-translator`: Universal translation support.

9. Testing:

- `github.com/stretchr/testify`: Testing utilities for assertions and mocks.

- `gopkg.in/go-playground/assert.v1`: Simplified assertion library.

10. Miscellaneous:

- `github.com/davecgh/go-spew`: Utility to display complex data structures.

- `github.com/spf13/cast`: Easy and safe casting from one type to another.

- `github.com/pmezard/go-difflib`: A library for computing differences between sequences.

- `golang.org/x/crypto`: Cryptographic algorithms and functions.

- `golang.org/x/sys`: Low-level operating system interaction.

- `golang.org/x/sync`: Additional synchronization primitives.

- `golang.org/x/text`: Text processing, including character encodings and transformations.

- `gopkg.in/yaml.v3`: YAML support for configuration parsing.

- `go.uber.org/multierr`: Combining multiple errors into one.

1. How to run this app
2. Prerequisites

* Go 1.23.2 or later
* PostgreSQL
* Docker (optional, for running PostgreSQL in a container)

1. Installation
   1. Clone the repository:

**git clone https://github.com/koeylp/friends-management.git**

**cd friends-management**

**go mod tidy**

1. Run the application:

**go run /cmd/main.go**

or if you have built the binary:

**./main**

1. Running the Application Using Docker

Start the application:

Run the following command in the root directory of your project:

**docker-compose up –build -d**

This will build the Docker images and start the PostgreSQL database and your application. The application will be available at http://localhost:8080.

1. Stopping the Application

To stop the application, run:

**docker-compose down**

1. API
   1. Create a friend connection

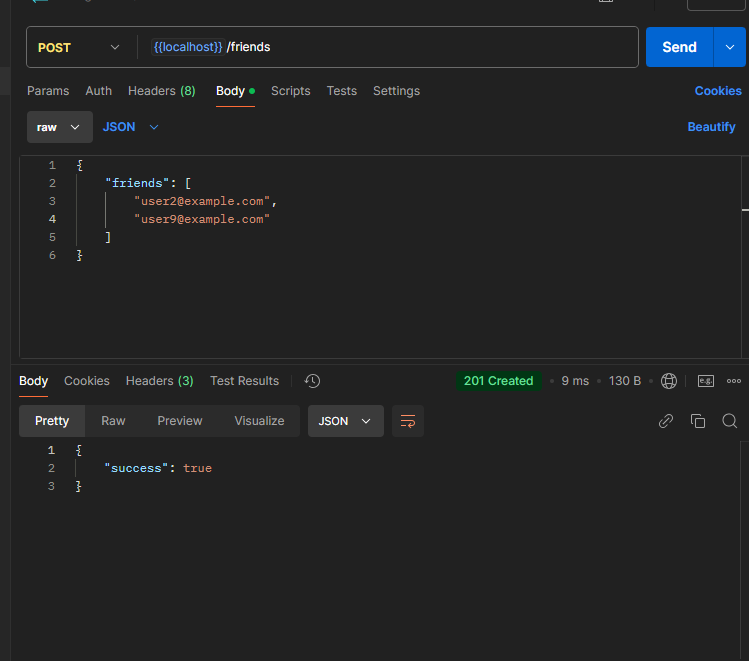


Figure 1. Success\_Case\_1: Create a friend connection.

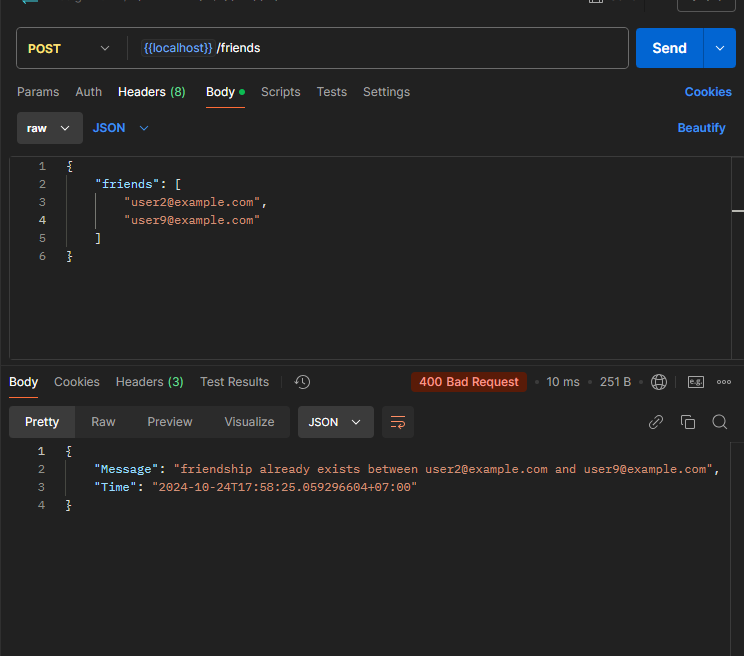


Figure 2. Error\_Case\_1: Create a friend connection.

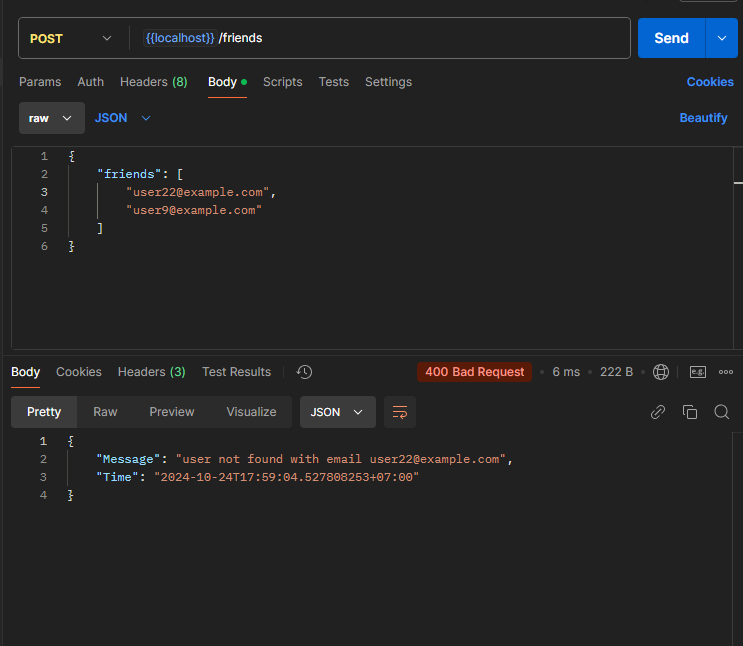


Figure 3. Error\_Case\_2: Create a friend connection.

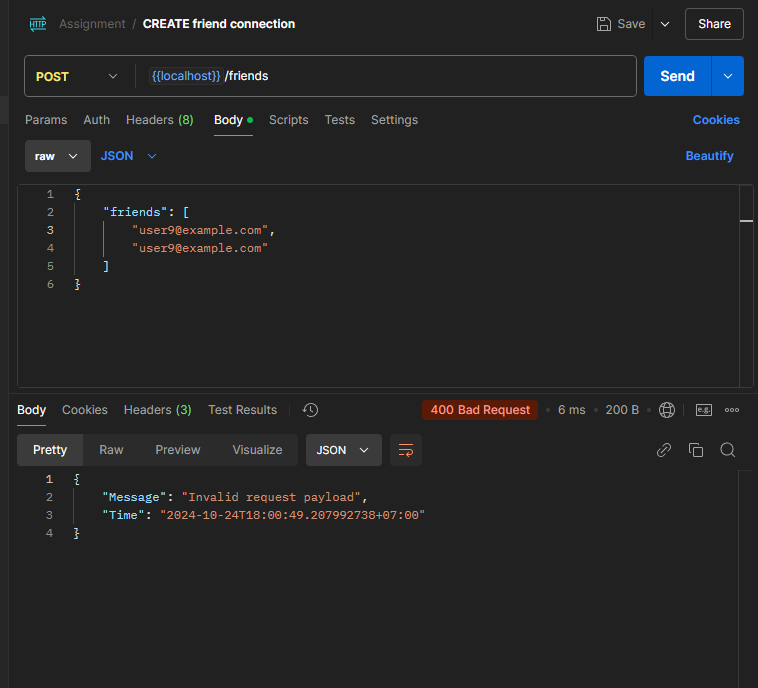


Figure 4. Error\_Case\_3: Create a friend connection.

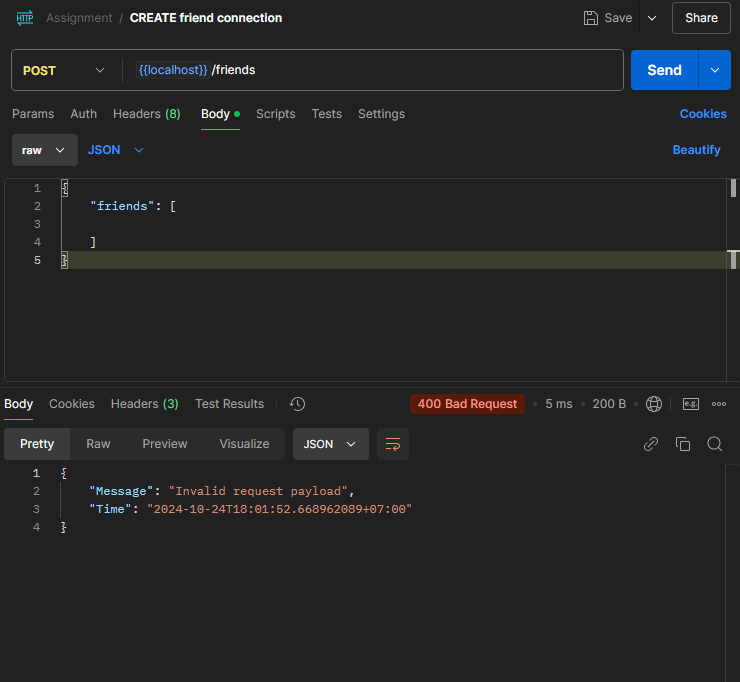


Figure 5. Error\_Case\_4: Create a friend connection.

* 1. Retrieve the friends list for an email address

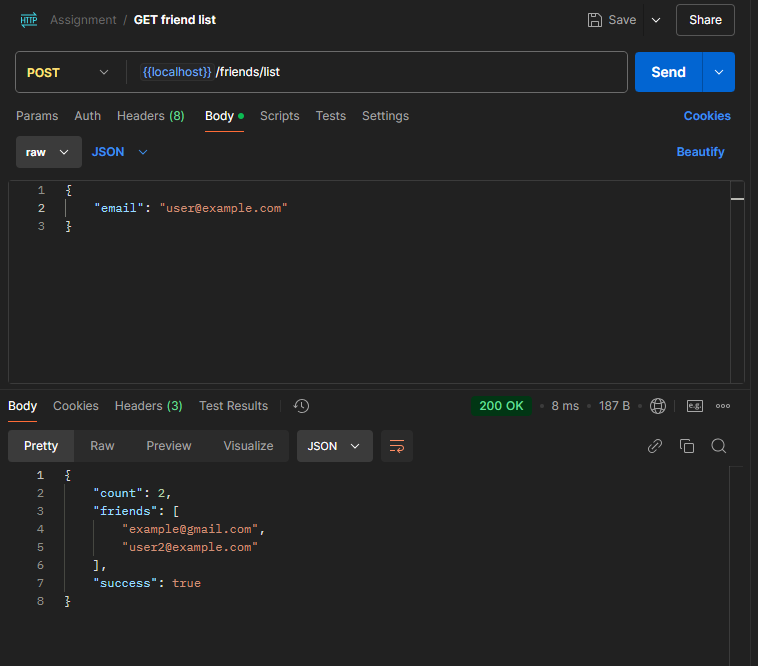


Figure 6. Success\_Case\_1: Retrieve the friends list for an email address

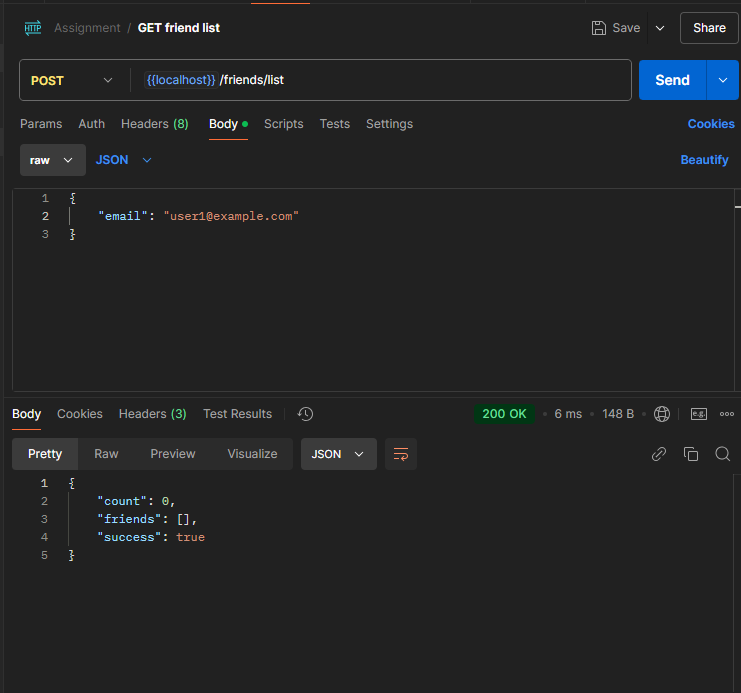


Figure 7. Success\_Case\_2: Retrieve the friends list for an email address

* 1. Retrieve the common friend list between two email addresses

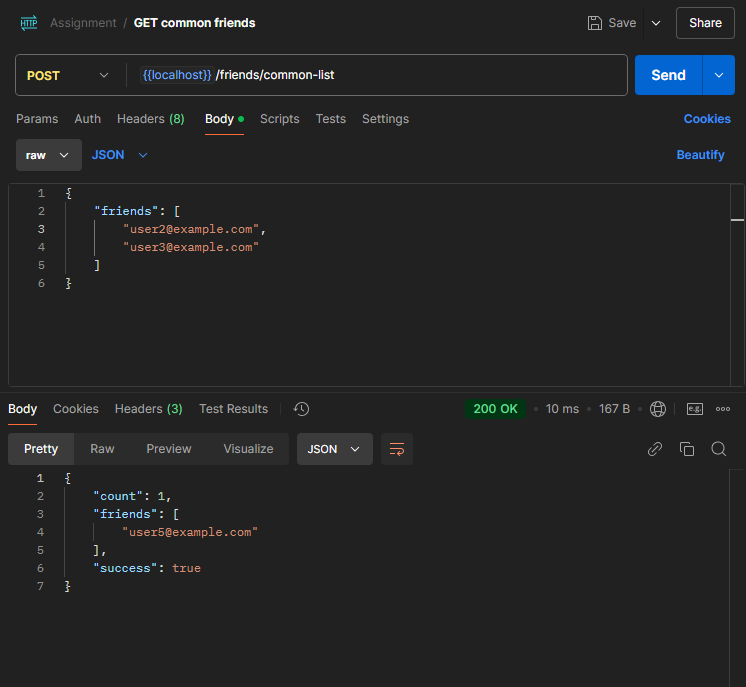


Figure . Success\_Case\_1: Retrieve the common friend list between two email addresses

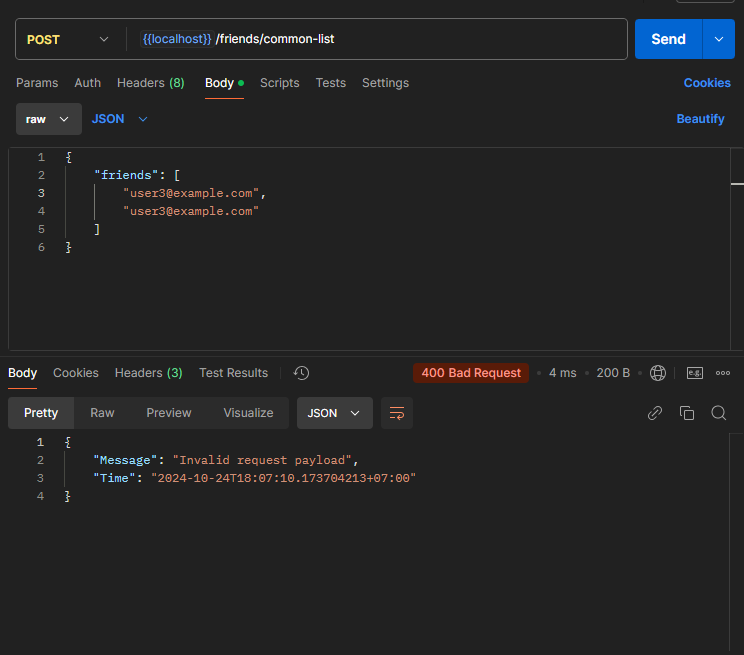


Figure . Error\_Case\_1: Retrieve the common friend list between two email addresses

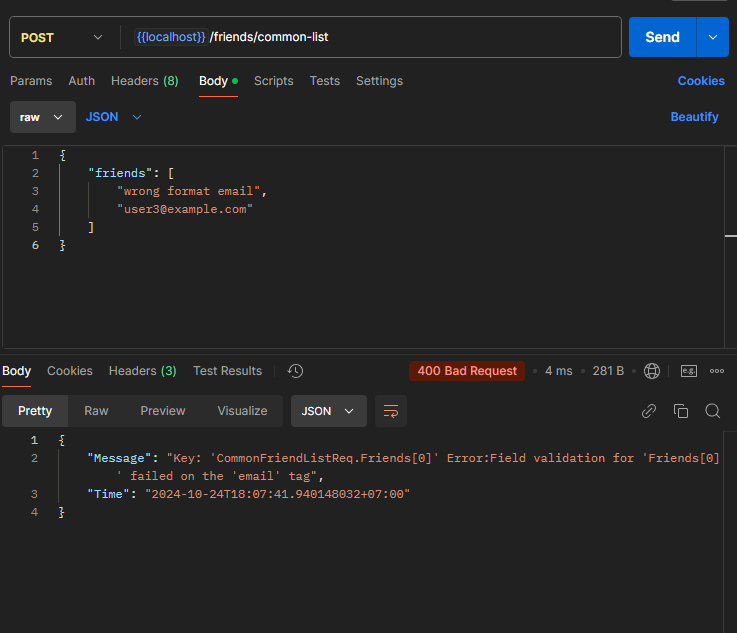


Figure . Error\_Case\_2: Retrieve the common friend list between two email addresses

* 1. Subscribe to updates from an email address

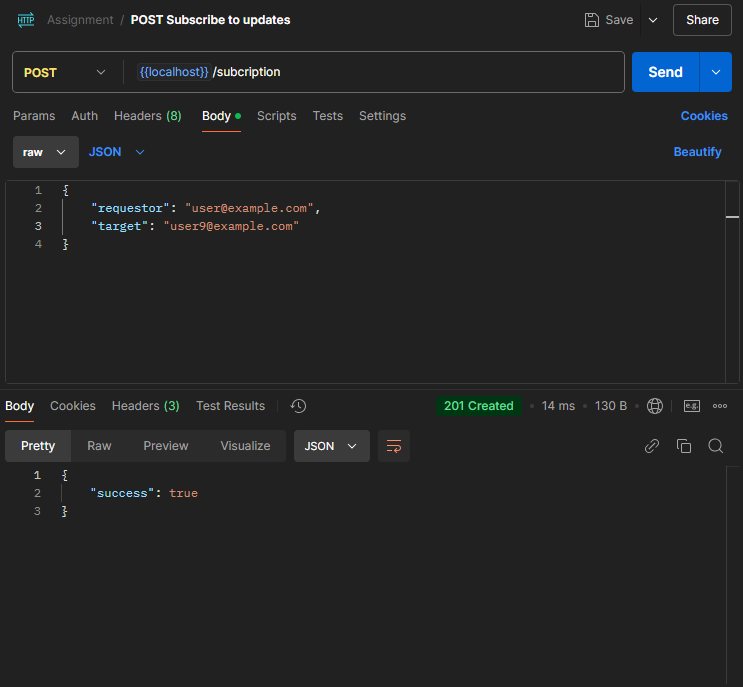


Figure . Success\_Case\_1: Subscribe to updates from an email address

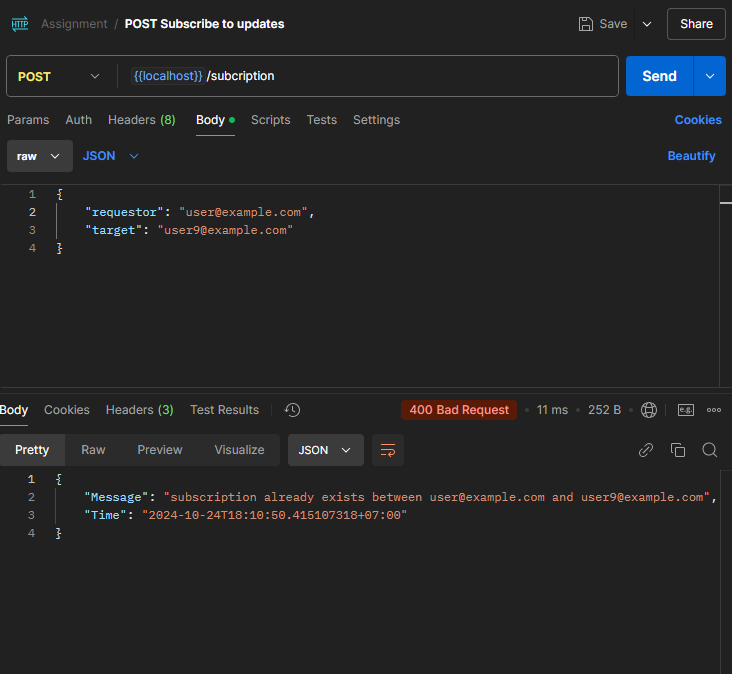


Figure . Errors\_Case\_1: Subscribe to updates from an email address

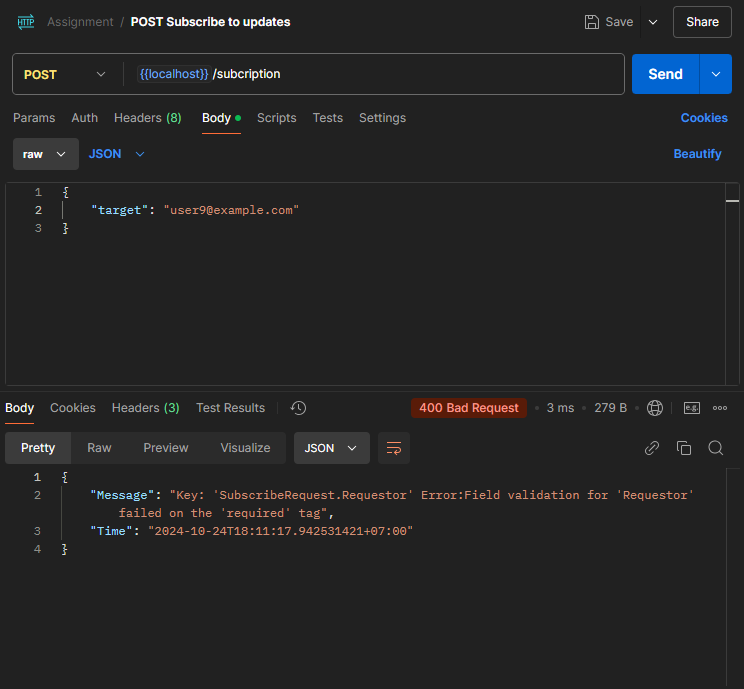


Figure . Errors\_Case\_2: Subscribe to updates from an email address

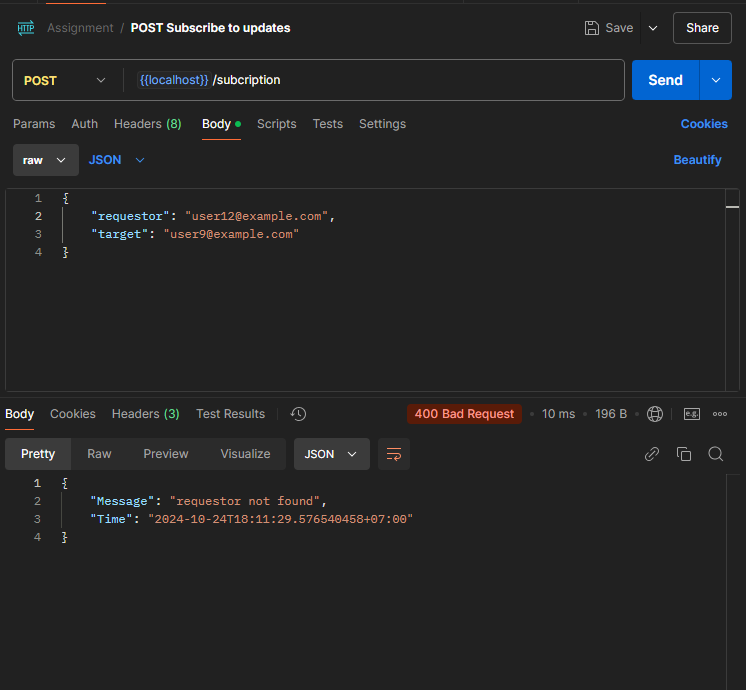


Figure . Errors\_Case\_3: Subscribe to updates from an email address

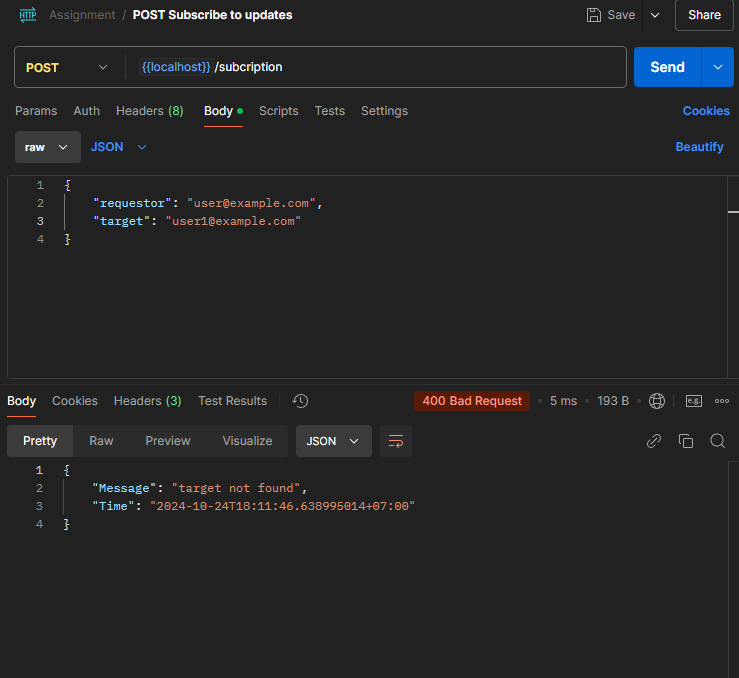


Figure . Errors\_Case\_4: Subscribe to updates from an email address

* 1. Block updates from an email address

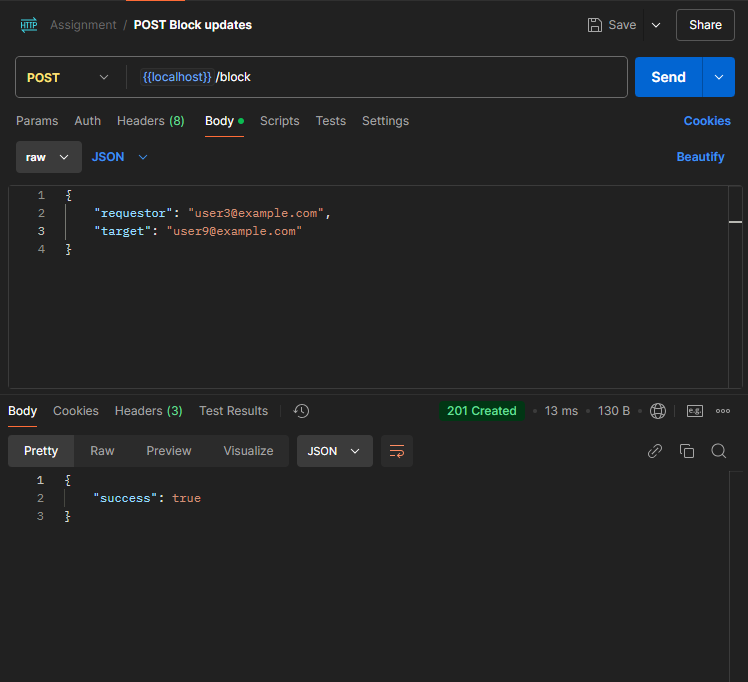


Figure . Success\_Case\_1: Block updates from an email address

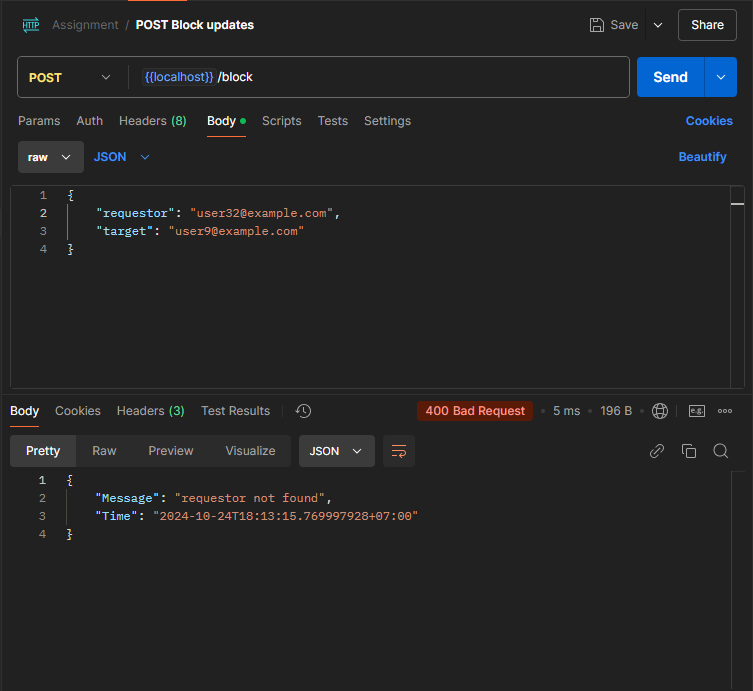


Figure . Error\_Case\_1: Block updates from an email address

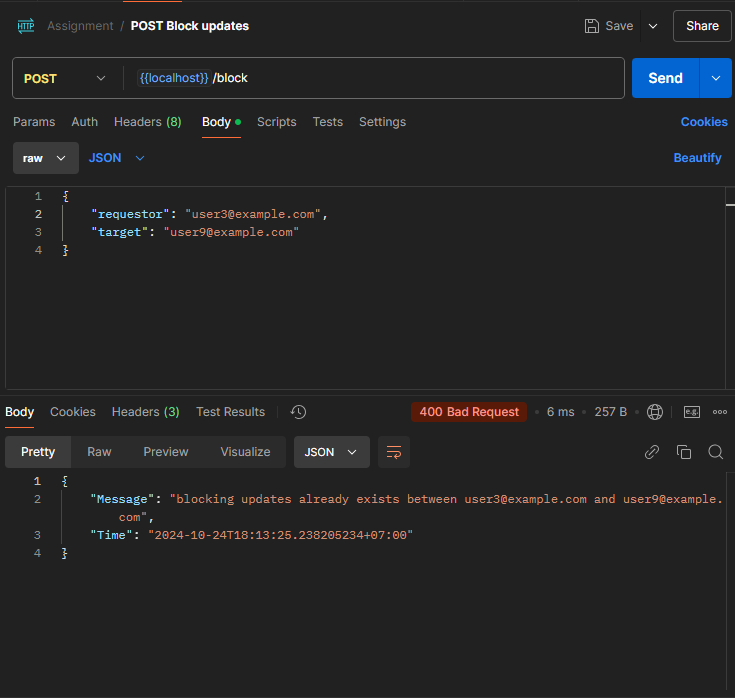


Figure . Error\_Case\_2: Block updates from an email address

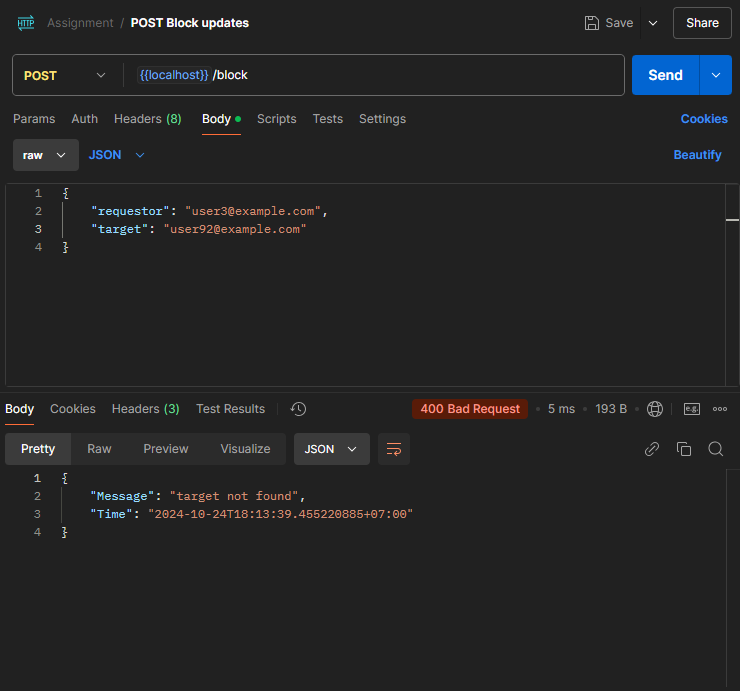


Figure . Error\_Case\_3: Block updates from an email address

* 1. Retrieve all updatable email addresses

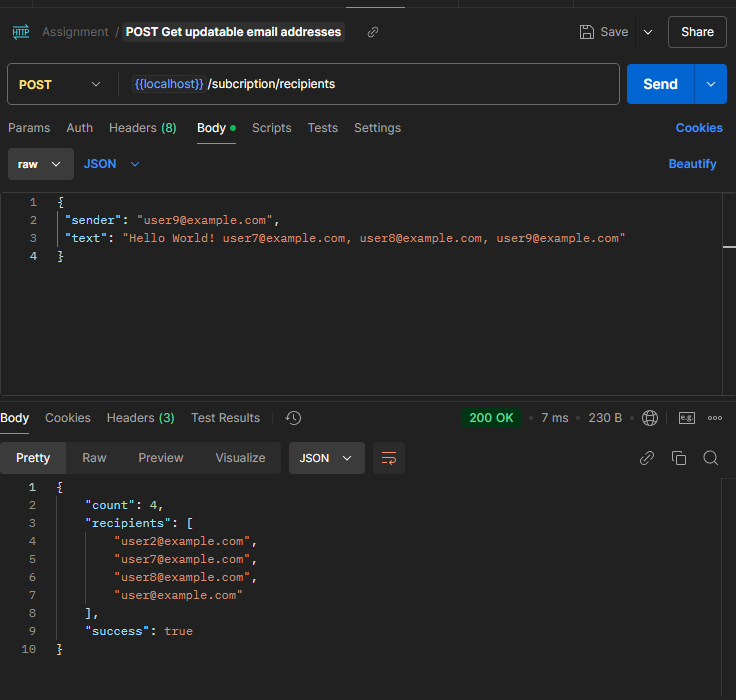


Figure . Success\_Case\_1: Retrieve all updatable email addresses

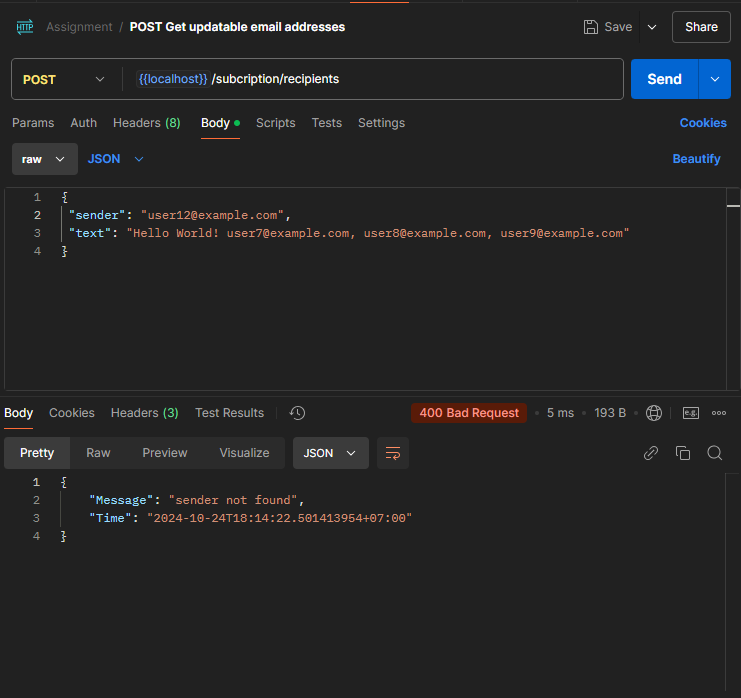


Figure .Error\_Case\_1: Retrieve all updatable email addresses

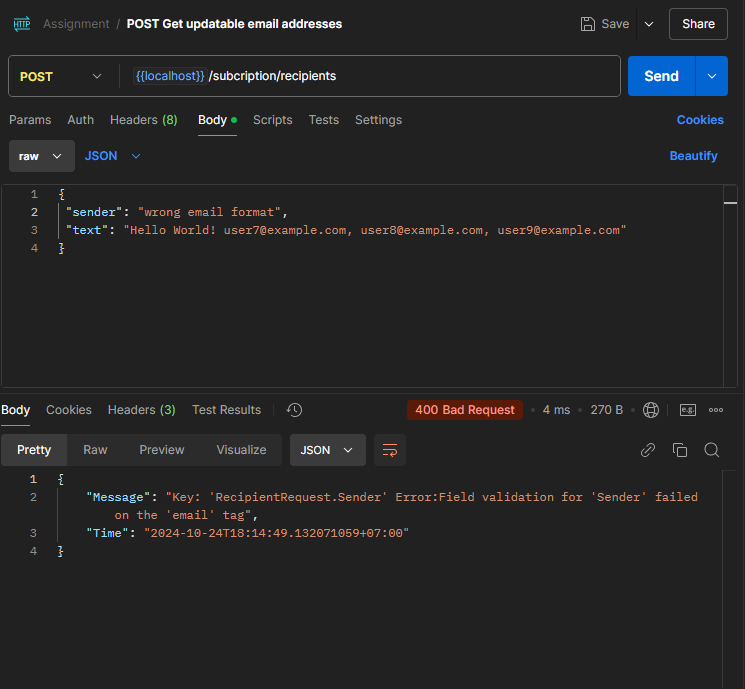


Figure .Error\_Case\_2: Retrieve all updatable email addresses

\*\* Unit Test

* Handler layer

=== RUN TestCreateFriendHandler

=== RUN TestCreateFriendHandler/Valid\_request

=== RUN TestCreateFriendHandler/Invalid\_request\_-\_empty\_friends

2024/10/25 14:44:39 ERROR: 400 -- Invalid request payload

=== RUN TestCreateFriendHandler/Invalid\_request\_-\_same\_friends

2024/10/25 14:44:39 ERROR: 400 -- Invalid request payload

--- PASS: TestCreateFriendHandler (0.00s)

--- PASS: TestCreateFriendHandler/Valid\_request (0.00s)

--- PASS: TestCreateFriendHandler/Invalid\_request\_-\_empty\_friends (0.00s)

--- PASS: TestCreateFriendHandler/Invalid\_request\_-\_same\_friends (0.00s)

=== RUN TestGetFriendListByEmailHandler

=== RUN TestGetFriendListByEmailHandler/Valid\_request

=== RUN TestGetFriendListByEmailHandler/Invalid\_request\_-\_empty\_email

2024/10/25 14:44:39 ERROR: 400 -- Key: 'EmailRequest.Email' Error:Field validation for 'Email' failed on the 'required' tag

--- PASS: TestGetFriendListByEmailHandler (0.00s)

--- PASS: TestGetFriendListByEmailHandler/Valid\_request (0.00s)

--- PASS: TestGetFriendListByEmailHandler/Invalid\_request\_-\_empty\_email (0.00s)

=== RUN TestGetCommonListHandler

=== RUN TestGetCommonListHandler/Valid\_request

=== RUN TestGetCommonListHandler/Invalid\_request\_-\_same\_friends

2024/10/25 14:44:39 ERROR: 400 -- Invalid request payload

--- PASS: TestGetCommonListHandler (0.00s)

--- PASS: TestGetCommonListHandler/Valid\_request (0.00s)

--- PASS: TestGetCommonListHandler/Invalid\_request\_-\_same\_friends (0.00s)

=== RUN TestSubscribeHandler

=== RUN TestSubscribeHandler/Valid\_subscription\_request

=== RUN TestSubscribeHandler/Invalid\_subscription\_-\_same\_requestor\_and\_target

2024/10/25 14:44:39 ERROR: 400 -- Invalid request payload

=== RUN TestSubscribeHandler/Invalid\_JSON\_payload

2024/10/25 14:44:39 ERROR: 400 -- Invalid request payload

--- PASS: TestSubscribeHandler (0.00s)

--- PASS: TestSubscribeHandler/Valid\_subscription\_request (0.00s)

--- PASS: TestSubscribeHandler/Invalid\_subscription\_-\_same\_requestor\_and\_target (0.00s)

--- PASS: TestSubscribeHandler/Invalid\_JSON\_payload (0.00s)

=== RUN TestBlockUpdatesHandler

=== RUN TestBlockUpdatesHandler/Valid\_block\_request

=== RUN TestBlockUpdatesHandler/Invalid\_block\_-\_same\_requestor\_and\_target

2024/10/25 14:44:39 ERROR: 400 -- Invalid request payload

=== RUN TestBlockUpdatesHandler/Invalid\_JSON\_payload

2024/10/25 14:44:39 ERROR: 400 -- Invalid request payload

--- PASS: TestBlockUpdatesHandler (0.00s)

--- PASS: TestBlockUpdatesHandler/Valid\_block\_request (0.00s)

--- PASS: TestBlockUpdatesHandler/Invalid\_block\_-\_same\_requestor\_and\_target (0.00s)

--- PASS: TestBlockUpdatesHandler/Invalid\_JSON\_payload (0.00s)

=== RUN TestGetUpdatableEmailAddressesHandler

=== RUN TestGetUpdatableEmailAddressesHandler/Valid\_request\_for\_updatable\_emails

=== RUN TestGetUpdatableEmailAddressesHandler/Invalid\_request\_-\_missing\_sender\_and\_text

2024/10/25 14:44:39 ERROR: 400 -- Key: 'RecipientRequest.Sender' Error:Field validation for 'Sender' failed on the 'required' tag

Key: 'RecipientRequest.Text' Error:Field validation for 'Text' failed on the 'required' tag

--- PASS: TestGetUpdatableEmailAddressesHandler (0.00s)

--- PASS: TestGetUpdatableEmailAddressesHandler/Valid\_request\_for\_updatable\_emails (0.00s)

--- PASS: TestGetUpdatableEmailAddressesHandler/Invalid\_request\_-\_missing\_sender\_and\_text (0.00s)

=== RUN TestCreateUserHandler

=== RUN TestCreateUserHandler/Valid\_user\_creation\_request

=== RUN TestCreateUserHandler/Invalid\_request\_-\_JSON\_decoding\_error

2024/10/25 14:44:39 ERROR: 400 -- Key: 'CreateUser.Email' Error:Field validation for 'Email' failed on the 'required' tag

=== RUN TestCreateUserHandler/User\_creation\_failure

--- PASS: TestCreateUserHandler (0.00s)

--- PASS: TestCreateUserHandler/Valid\_user\_creation\_request (0.00s)

--- PASS: TestCreateUserHandler/Invalid\_request\_-\_JSON\_decoding\_error (0.00s)

--- PASS: TestCreateUserHandler/User\_creation\_failure (0.00s)

PASS

ok github.com/koeylp/friends-management/internal/handlers 0.006s

* Service layer

=== RUN TestCreateFriend

2024/10/25 14:45:15 ERROR: 400 -- friendship already exists between requestor@example.com and target@example.com

2024/10/25 14:45:15 ERROR: 400 -- blocking updates exists between requestor@example.com and target@example.com

2024/10/25 14:45:15 ERROR: 400 -- user not found with email requestor@example.com

2024/10/25 14:45:15 ERROR: 400 -- user not found with email target@example.com

--- PASS: TestCreateFriend (0.00s)

=== RUN TestGetFriendListByEmail\_Success

--- PASS: TestGetFriendListByEmail\_Success (0.00s)

=== RUN TestGetFriendListByEmail\_NoFriendsFound

--- PASS: TestGetFriendListByEmail\_NoFriendsFound (0.00s)

=== RUN TestGetFriendListByEmail\_ErrorFetchingFriends

--- PASS: TestGetFriendListByEmail\_ErrorFetchingFriends (0.00s)

=== RUN TestRelationshipService\_GetCommonList

--- PASS: TestRelationshipService\_GetCommonList (0.00s)

=== RUN TestSubcribe\_Success

--- PASS: TestSubcribe\_Success (0.00s)

=== RUN TestBlockUpdates

2024/10/25 14:45:15 ERROR: 400 -- blocking updates already exists between requestor@example.com and target@example.com

--- PASS: TestBlockUpdates (0.00s)

=== RUN TestGetUpdatableEmailAddresses

2024/10/25 14:45:15 ERROR: 400 -- sender not found

--- PASS: TestGetUpdatableEmailAddresses (0.00s)

=== RUN TestCreateUser\_Success

--- PASS: TestCreateUser\_Success (0.00s)

=== RUN TestCreateUser\_Failure

--- PASS: TestCreateUser\_Failure (0.00s)

=== RUN TestGetUserByEmail

--- PASS: TestGetUserByEmail (0.00s)

=== RUN TestGetUserByEmail\_NotFound

--- PASS: TestGetUserByEmail\_NotFound (0.00s)

PASS

ok github.com/koeylp/friends-management/internal/services 0.006s

* Repository layer

=== RUN TestCreateFriend

--- PASS: TestCreateFriend (0.00s)

=== RUN TestCheckFriendshipExists

--- PASS: TestCheckFriendshipExists (0.00s)

=== RUN TestGetFriends

--- PASS: TestGetFriends (0.00s)

=== RUN TestGetCommonFriends

--- PASS: TestGetCommonFriends (0.00s)

=== RUN TestSubscribe

--- PASS: TestSubscribe (0.00s)

=== RUN TestCheckSubscriptionExists

--- PASS: TestCheckSubscriptionExists (0.00s)

=== RUN TestCheckBlockExists

--- PASS: TestCheckBlockExists (0.00s)

=== RUN TestBlock

--- PASS: TestBlock (0.00s)

=== RUN TestGetUpdatableEmailAddresses

--- PASS: TestGetUpdatableEmailAddresses (0.00s)

=== RUN TestCreateUser

--- PASS: TestCreateUser (0.00s)

=== RUN TestGetUserByEmail

--- PASS: TestGetUserByEmail (0.00s)

PASS

ok github.com/koeylp/friends-management/internal/repositories (cached)