Upon checking out the project from :https://github.com/githubjvy/SamplesAndDemo.git

Aside from the basic requirement. I added my own touch for the solution with changes on the schema of response, structure of endpoints, pagination on get users. I also added an endpoint to reactivate and deactivate a user and also a queryParam called "show_all" in get users endpoint to return including the deactivated accounts.

Import the 3 projects onto the studio :
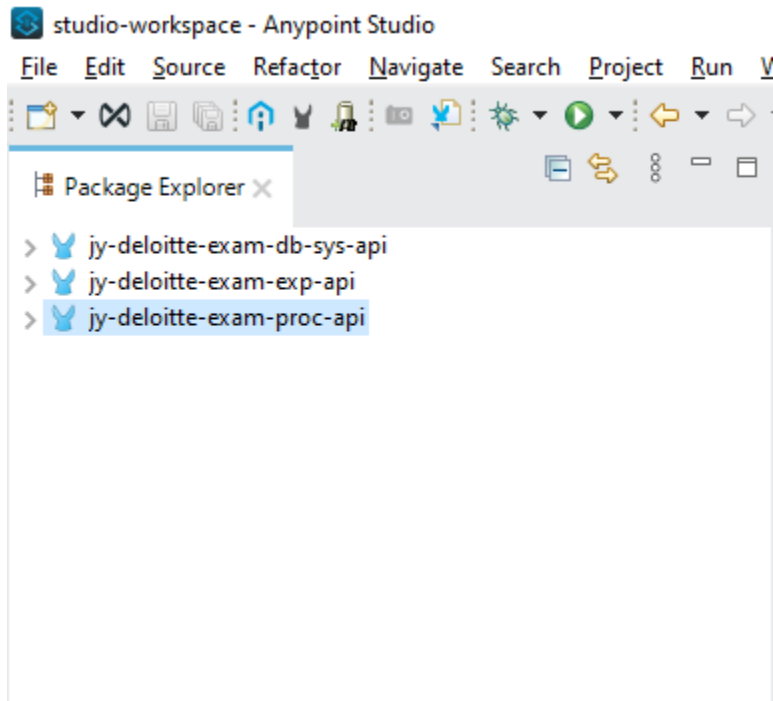


You need to run the 3 projects at the same time to simulate the API led approach that I have done for the solution. To that, create new Run Configuration and check all the projects
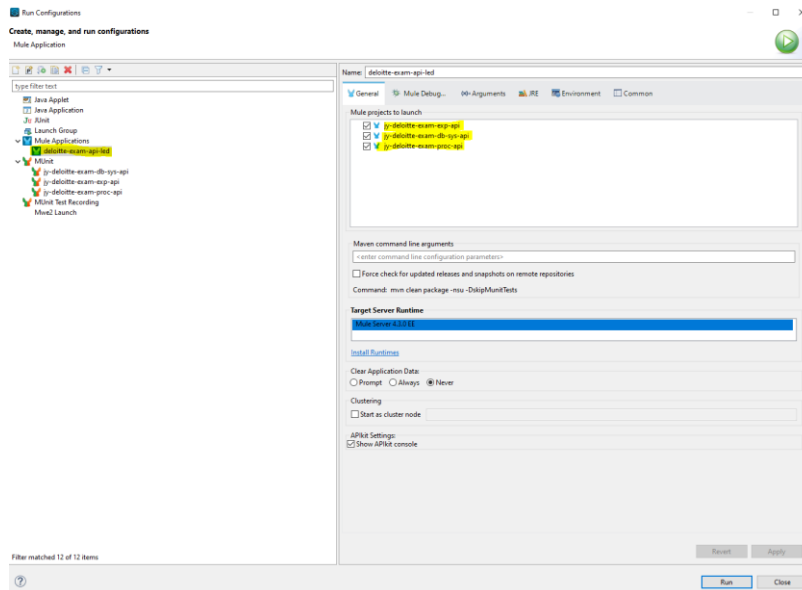
Ensure that you provide the environment variables properly because I have encrypted(Blowfish,CBC) the config file and the made the properties file dynamically available for different environment
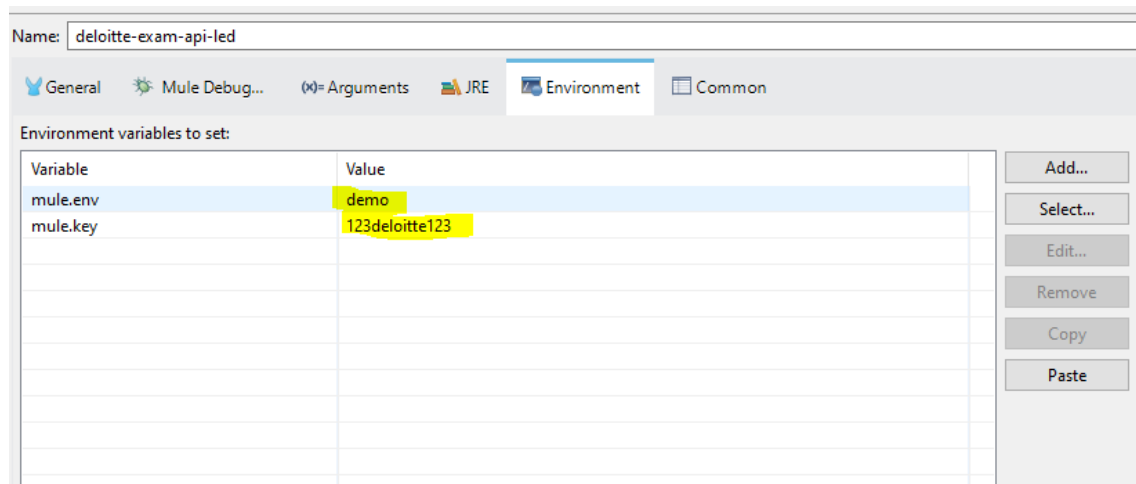


I initially coded this with a docker instance of an MS SQL database and a stored procedure that I wrote to simultaneously create Account and UserDetails entry at the same time and to avoid race condition. I decided to write an SP so that it would rollback if one of the INSERT queries from either UserDetails or Account failed.

If you wish to run an instance of the DB, please ensure to create the tables and the stored procedure. you may recreate them by executing the scripts on the next page.

```sql
use deloitte_db;

CREATE PROCEDURE dbo.[create_user]
(
      @fullName varchar(255)
    , @birthday date
    , @gender char(1)
    , @username varchar(20)
    , @active bit
    , @date_registered date
)
AS
BEGIN
    BEGIN TRANSACTION;
    SAVE TRANSACTION MySavePoint;
        SET NOCOUNT ON

    BEGIN TRY
        INSERT INTO [dbo].[UserDetails]
            (
             [fullName]
            ,[birthday]
            ,[gender]
            )
        VALUES (
             @fullName
            ,@birthday
            ,@gender
            );

        INSERT INTO [dbo].[Accounts]
            (
            [userId]
            ,[username]
            ,[active]
            ,[date_registered]
            )
        VALUES (
           (SELECT TOP(1) userId FROM UserDetails ORDER BY userId DESC)
            ,@username
            ,@active
            ,@date_registered
            );

        COMMIT TRANSACTION
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
        BEGIN
            ROLLBACK TRANSACTION MySavePoint; -- rollback to MySavePoint
            RETURN '0'
        END
    END CATCH

    RETURN '1'
END;

CREATE TABLE UserDetails (
        userId int IDENTITY(1001,1) NOT NULL,
        fullName varchar(255) NOT NULL,
        birthday date NOT NULL,
        gender char(1) NOT NULL,
        PRIMARY KEY (userId)
);



CREATE TABLE Accounts (
    accountId int IDENTITY(101,1) NOT NULL,
    userId int NOT NULL,
    username varchar(20) UNIQUE NOT NULL,
    active bit NOT NULL,
    date_registered date NOT NULL,
    PRIMARY KEY (accountId),
    FOREIGN KEY (userId) REFERENCES UserDetails(userId)
);
```

For the actual running of application, I set the experience api port to 8081. Here are the list of endpoints

All endpoint requires the Authorization header with value of **j32io33ise4k2qq1**

Create a user :

POST : http://localhost:8081/api/v1/users/user


Create Multiple Users (Async) :

POST : http://localhost:8081/api/v1/users


Get User by Username :

GET :  http://localhost:8081/api/v1/users/{username}


Get Users :

GET : http://localhost:8081/api/v1/users

Note : Provide the Security-Key header to proceed. Any 16 length string (example : abcdefg123456789)



Activate a user :

PATCH : http://localhost:8081/api/v1/users/{username}/activate

Deactivate a user :

PATCH : http://localhost:8081/api/v1/users/{username}/deactivate