**Blog Tracker Application**

**GithubLink:** [BARANIDHARAN-S-Git/BlogTrackerApplication (github.com)](https://github.com/BARANIDHARAN-S-Git/BlogTrackerApplication)

1)Create a classLibrary DAL and create Database in Azure using Code First Approach.

2)Create Three Table AdminInfo,EmpInfo and BlogInfo.

3)create Context Class by inheriting DbContext Class.

4)Add one more class by inheriting DropCreateDatabaseIfModelChanges<contextclass> and default data into AdminInfo table By Overriding the SEEDmethod.

5)write Database Functionalities for the EmpInfo and BlogInfo.

6)Create a ASP.Net WebAPI .This application contains RESTFUL Services to consume functionalities from the DAL Class.

7)Create a controller and write action methods to issue a GET,POST,PUT AND DELETE request to perform all the functionalities written for EmpInfo and BlogInfo entities inside DAL Layer.

8)Using Postman support to test this Layer.

9)Create a ASP.NET MVC as a Front end Application.

10)create controller and the respective view with the HTML ,CSS,Bootsrap for designing.

11)To display all blog details on the homepage written by all employees.

12)Authenticate Admin Using Admin EmailId and Password.

13)Page will be shown for adding Employee details by the Admin.

14)Authenticate Employee Using Employee Email Id and Passcode.

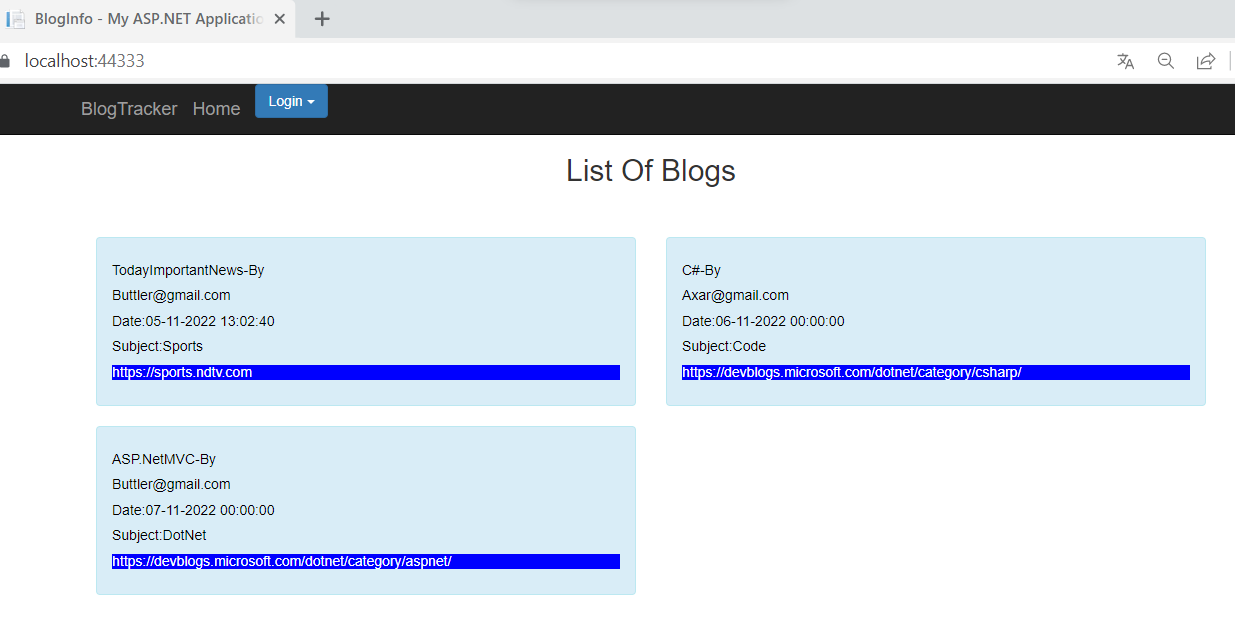
15)Page will be shown for adding Blog Details by the Employees.

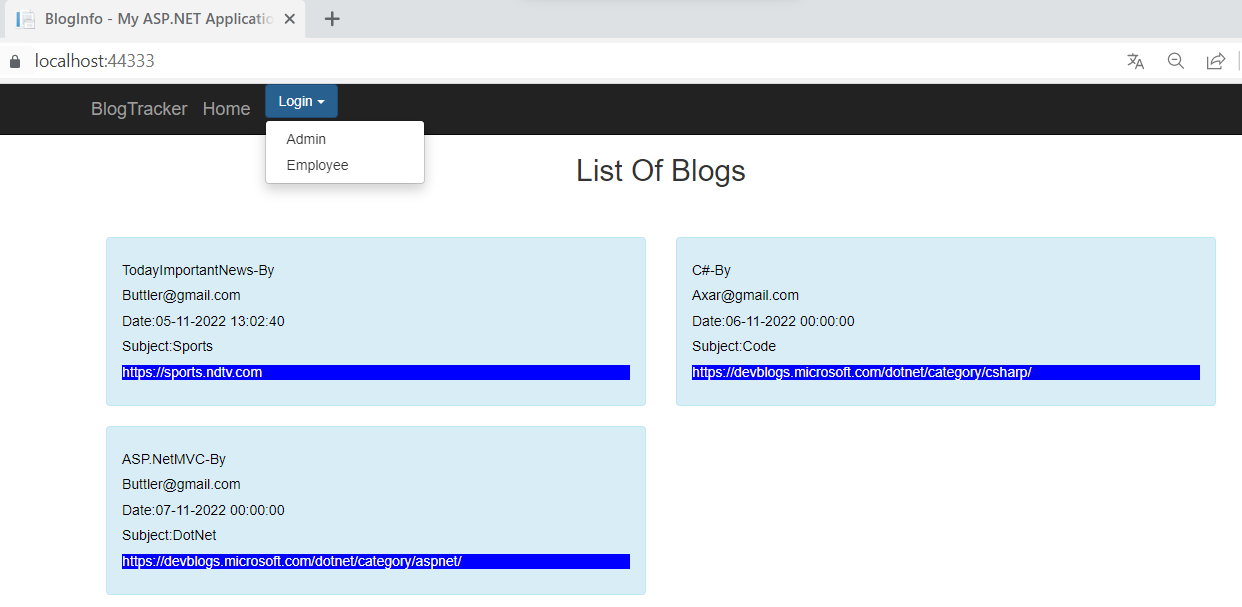
16)create DAL Test layer is a type of class Library to test DAL Layer Functionality .

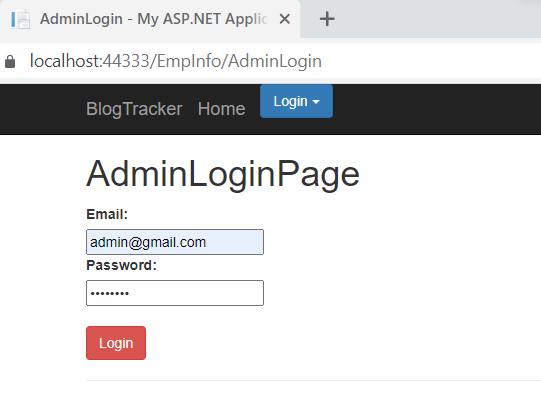
17)Deploy it in the Docker.

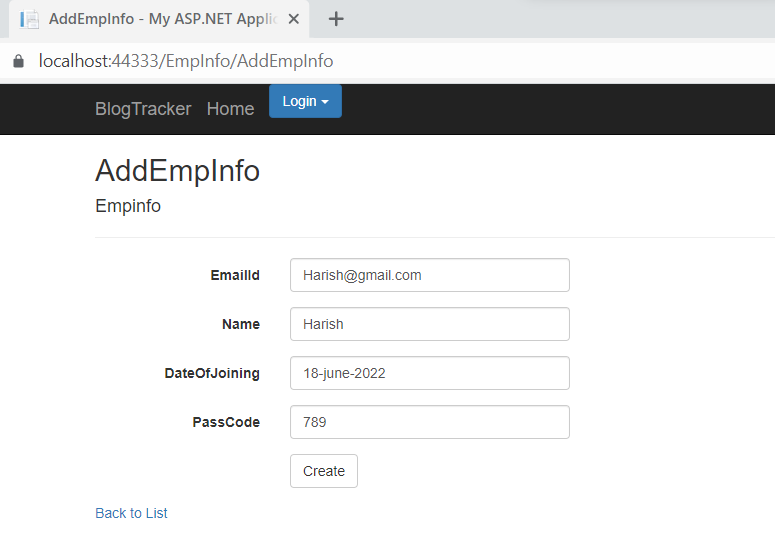
18)Run the Project and see the Output in the Browser.

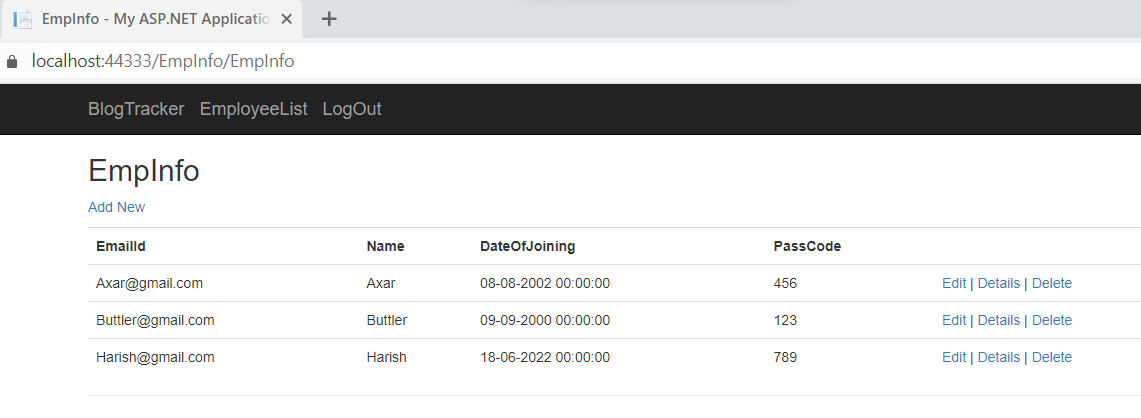
OUTPUT:

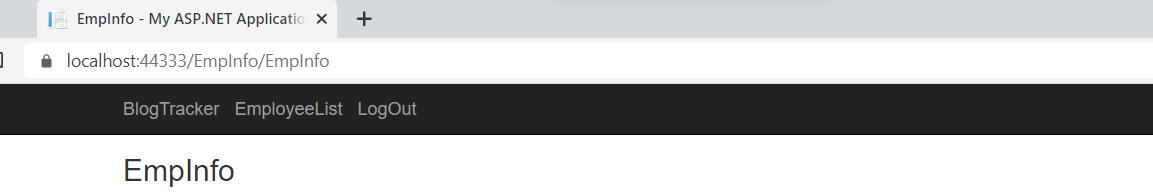


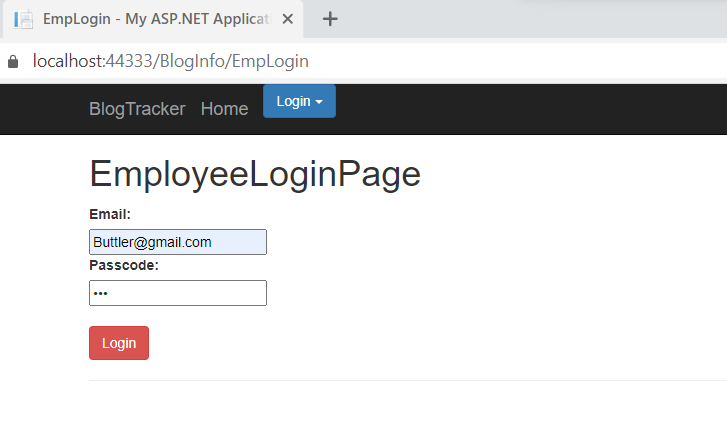


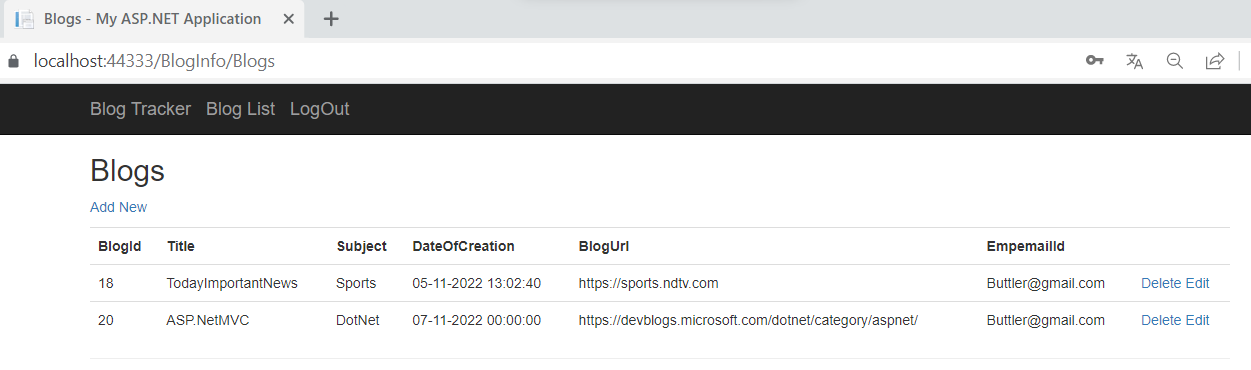


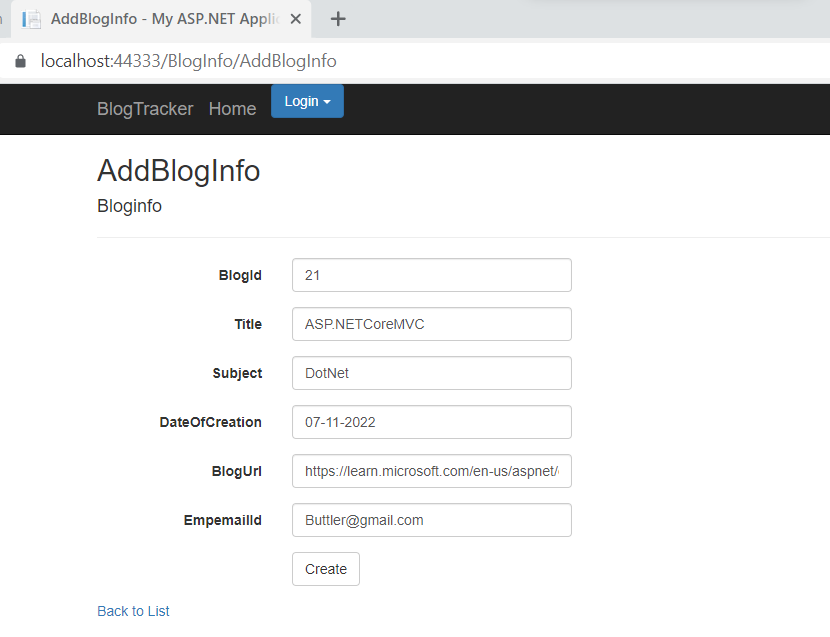


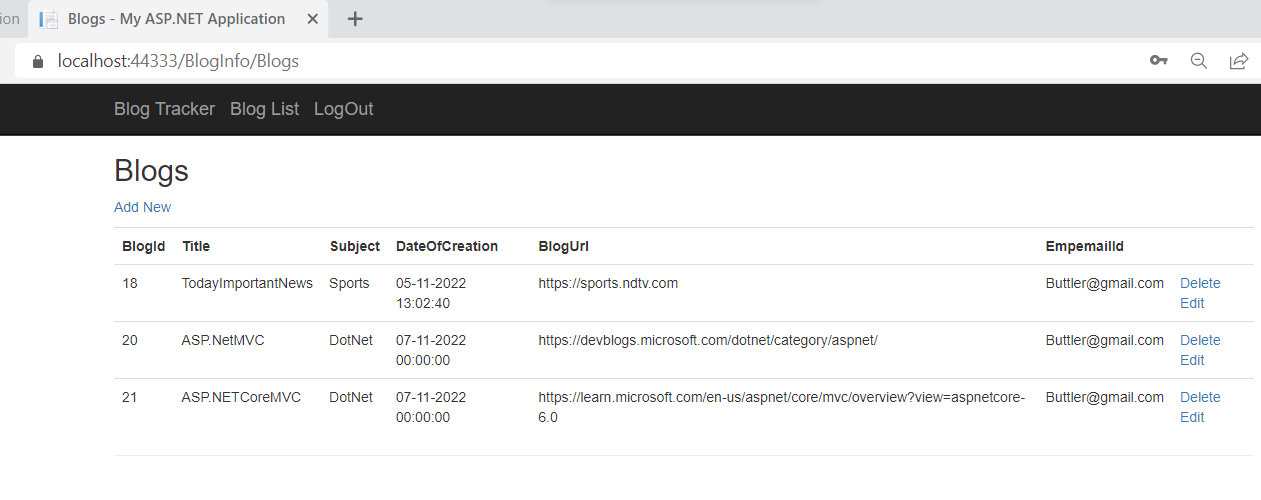


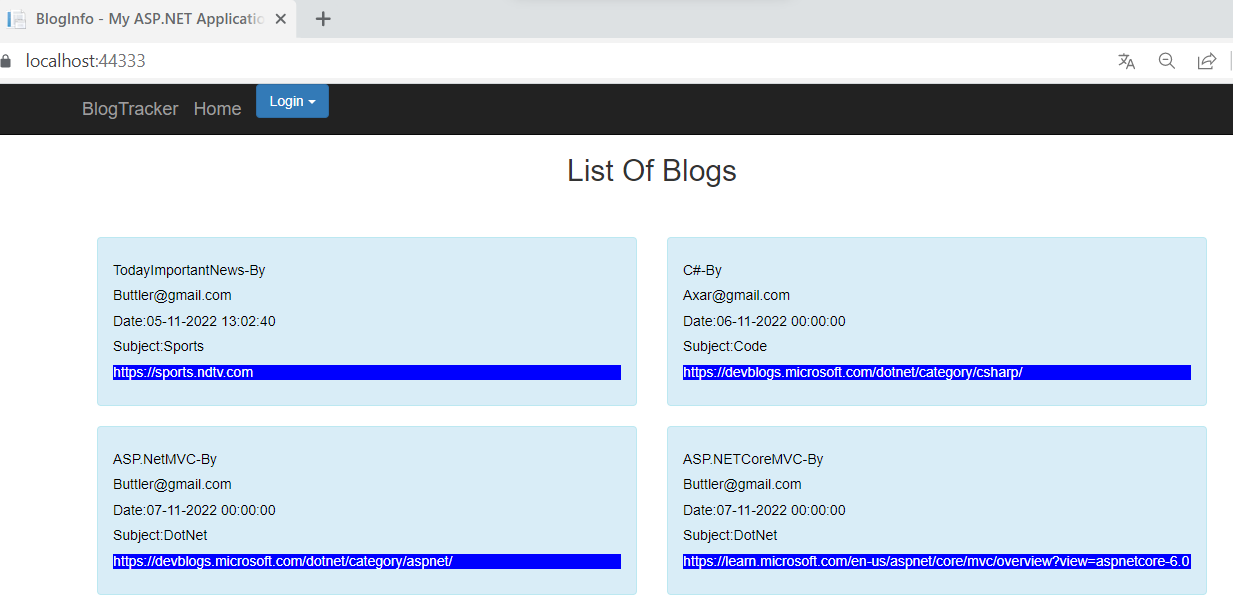




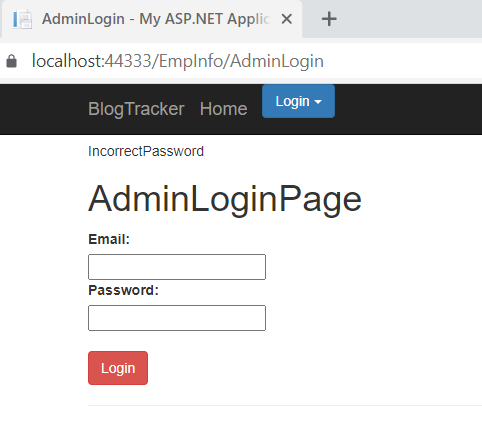


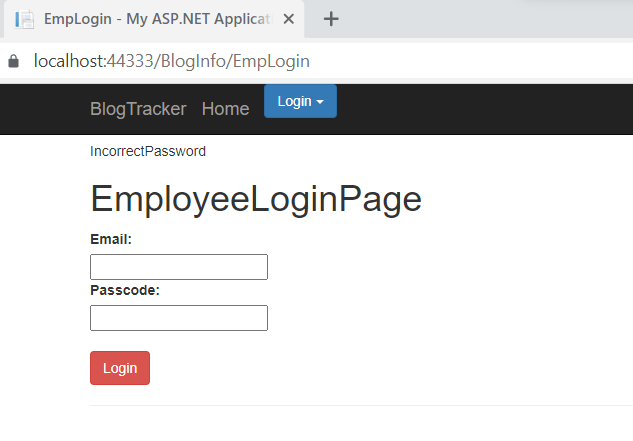




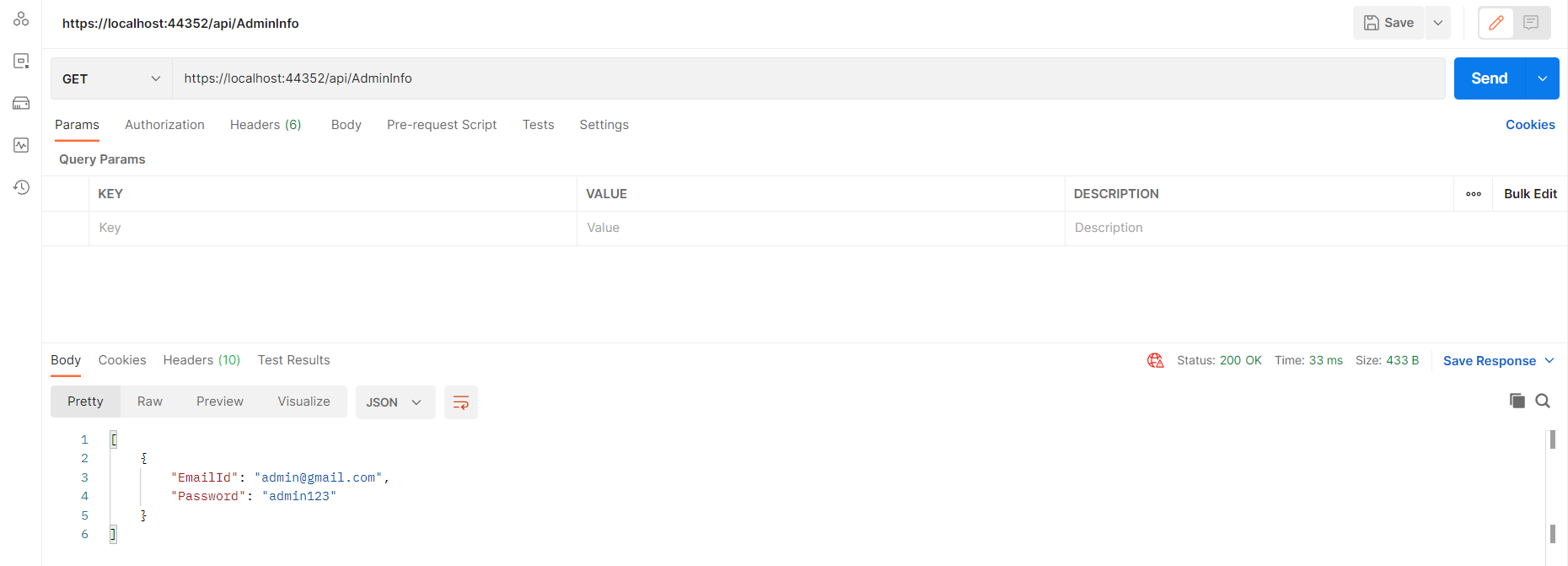
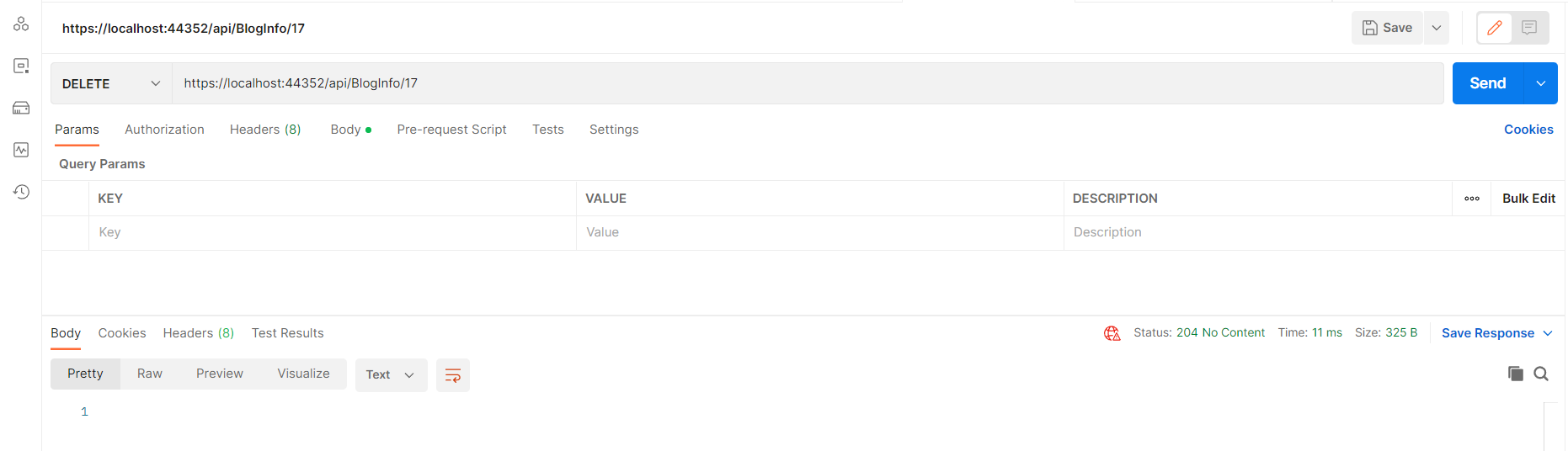
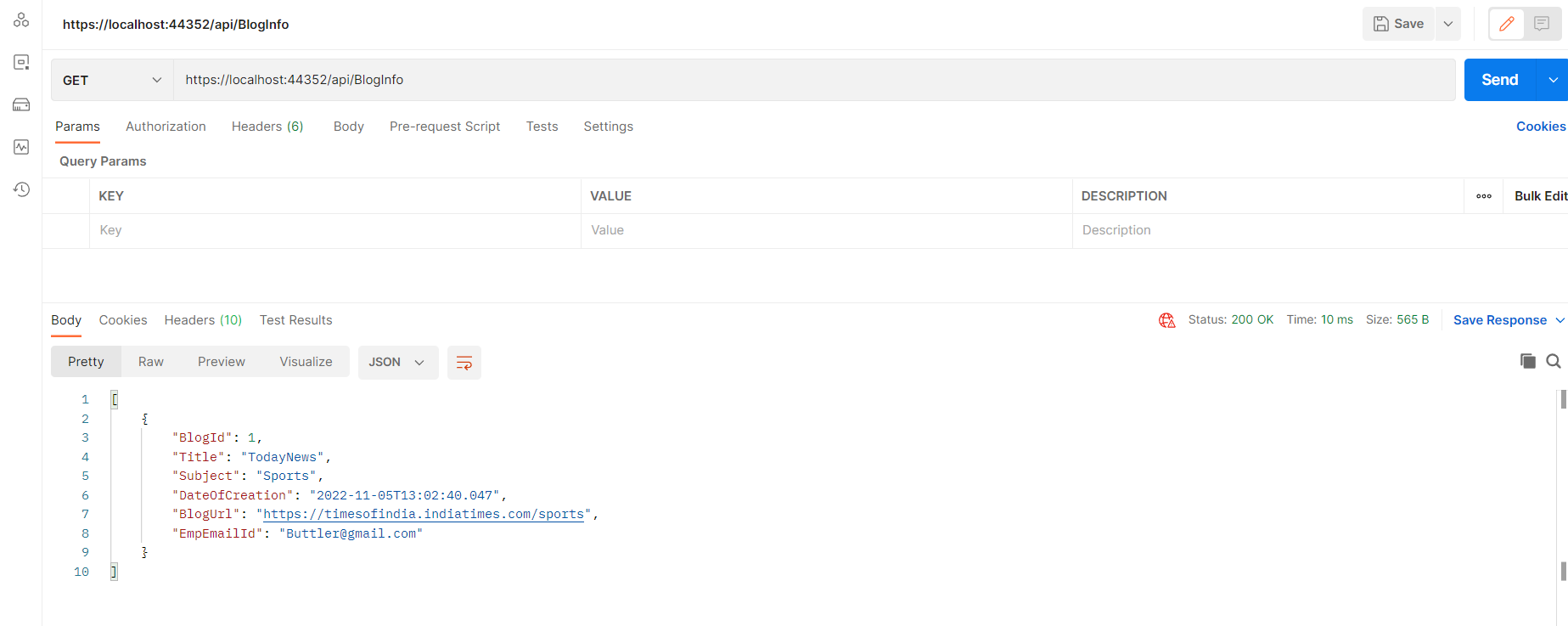
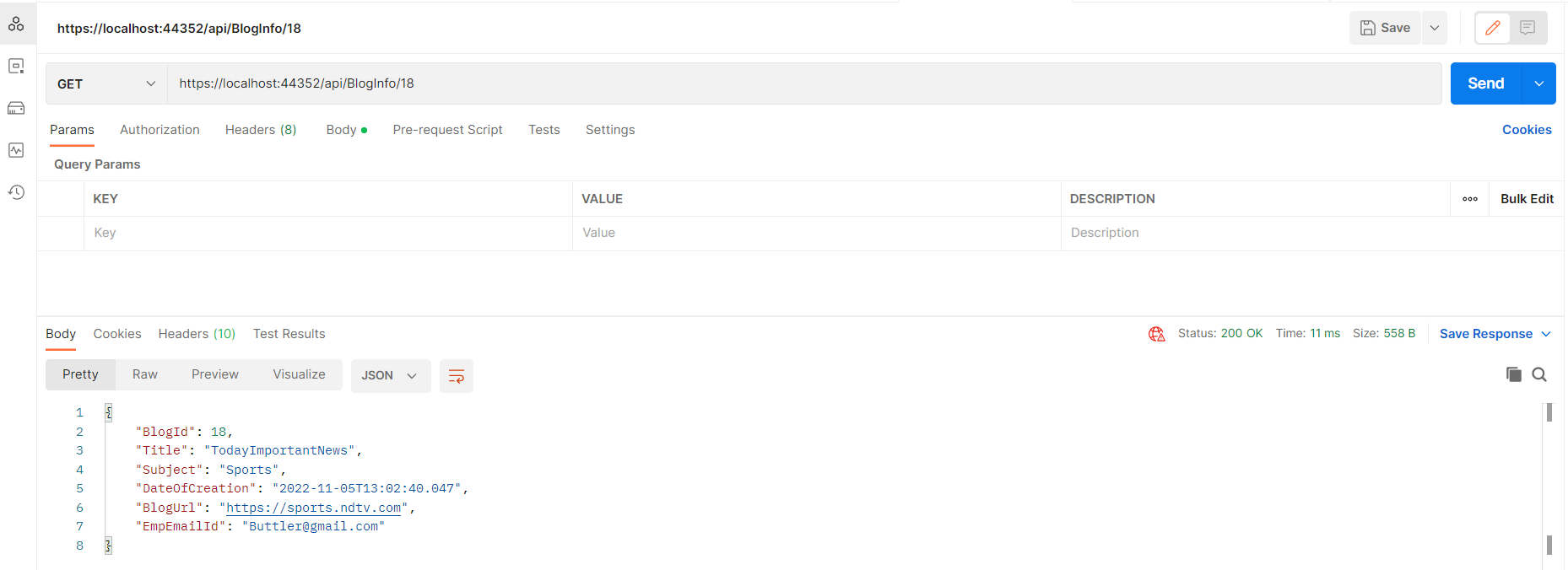
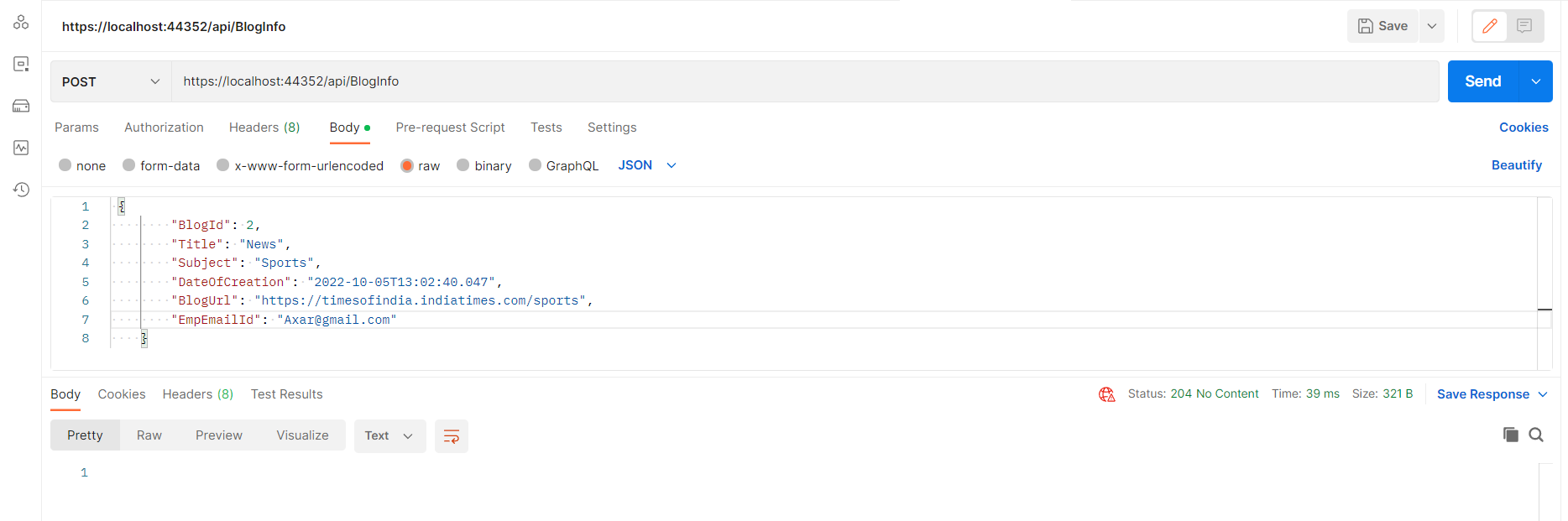
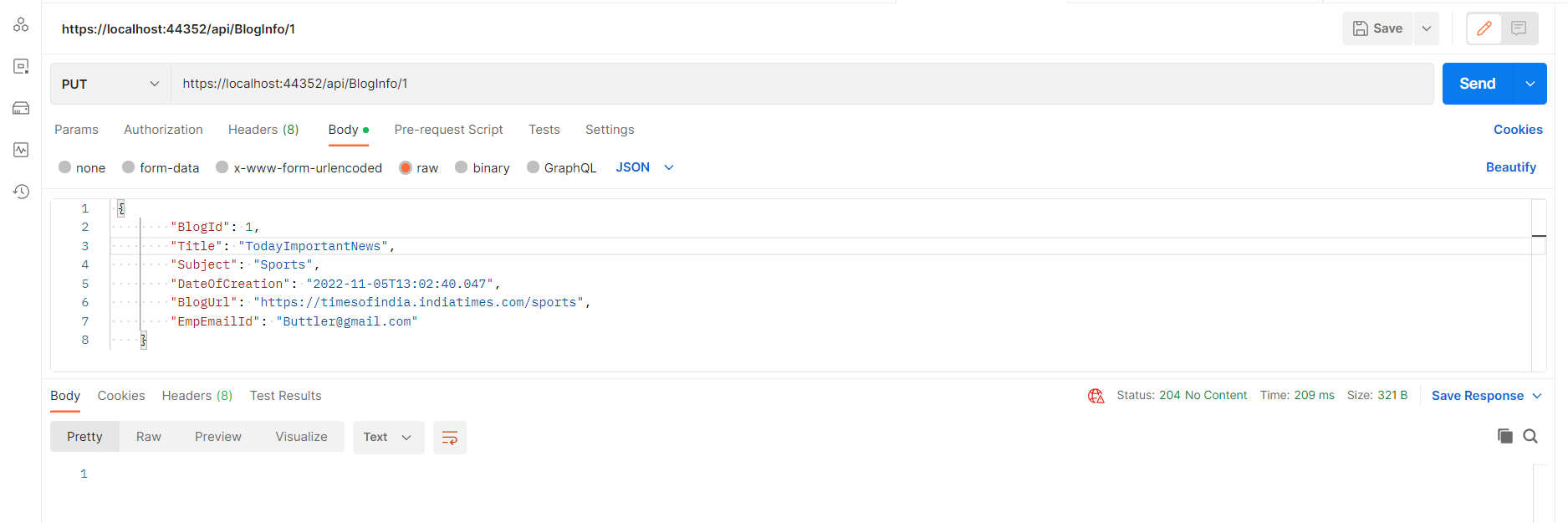


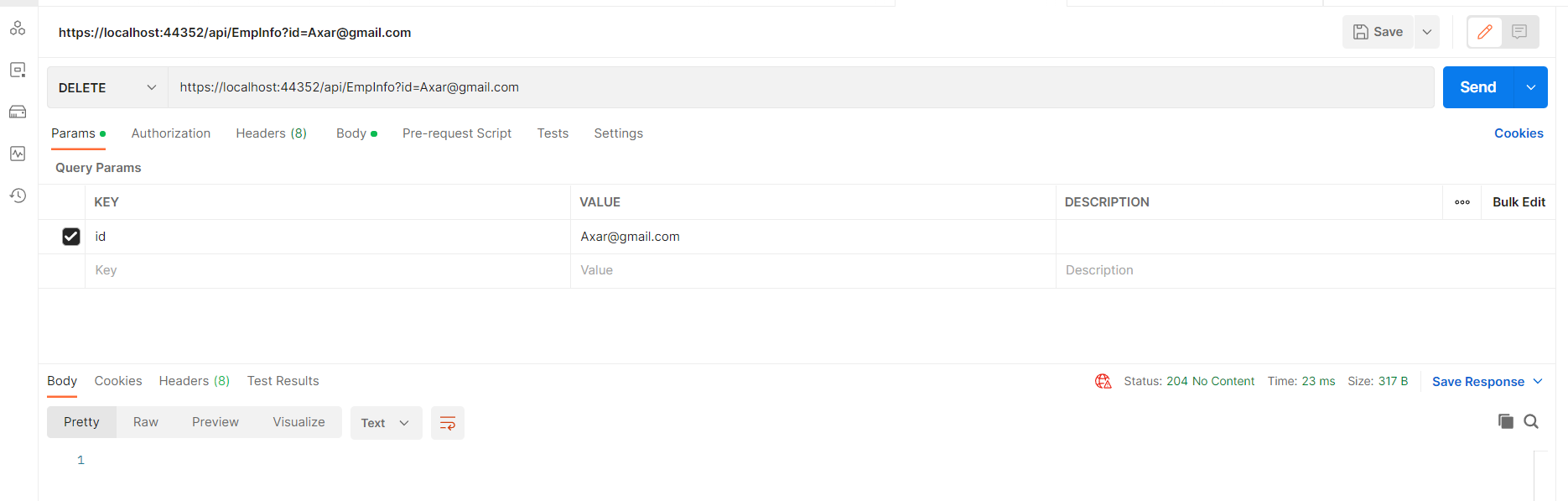
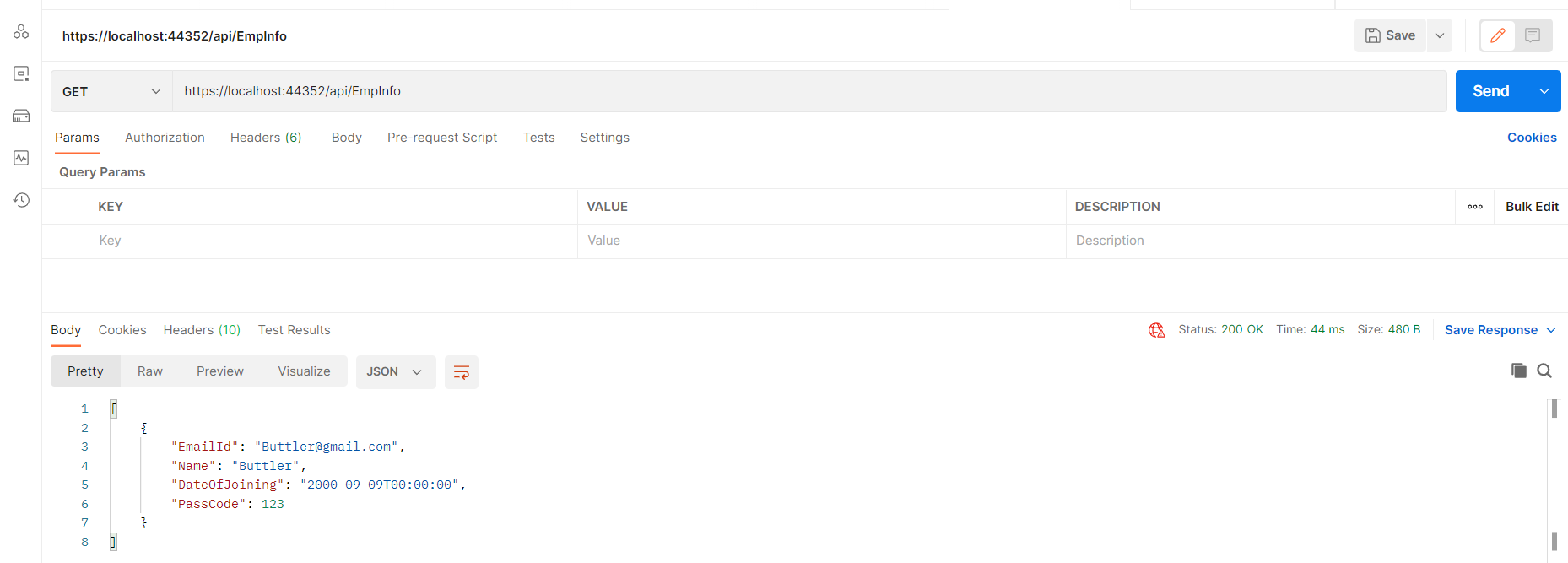
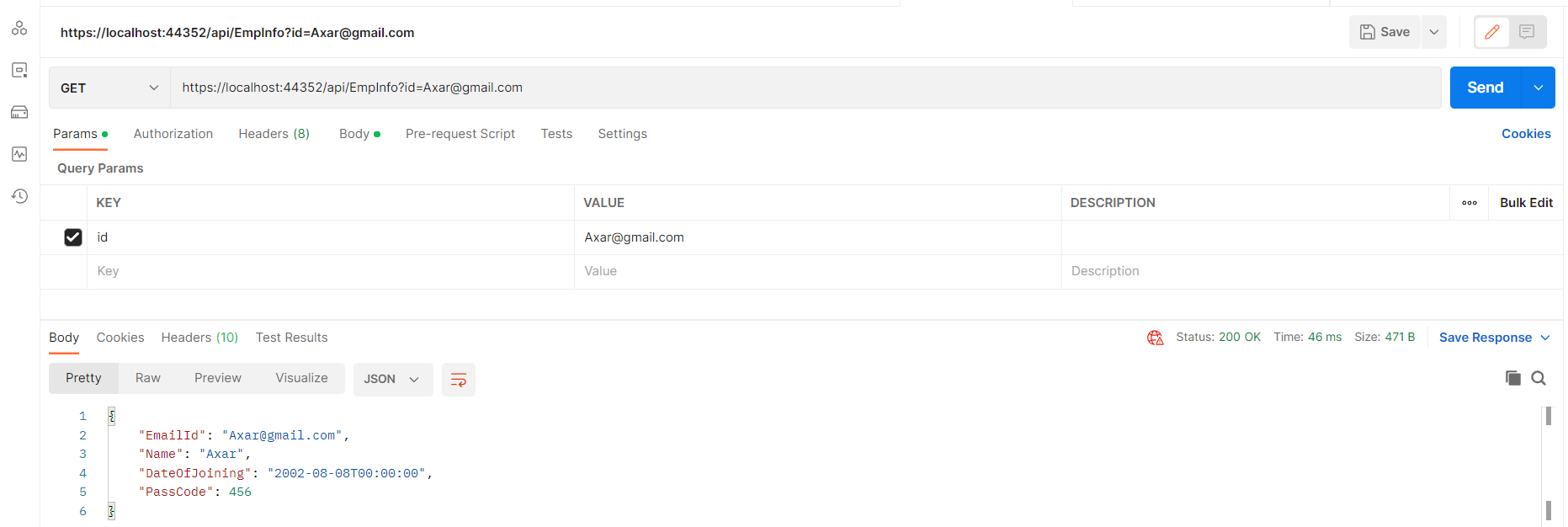
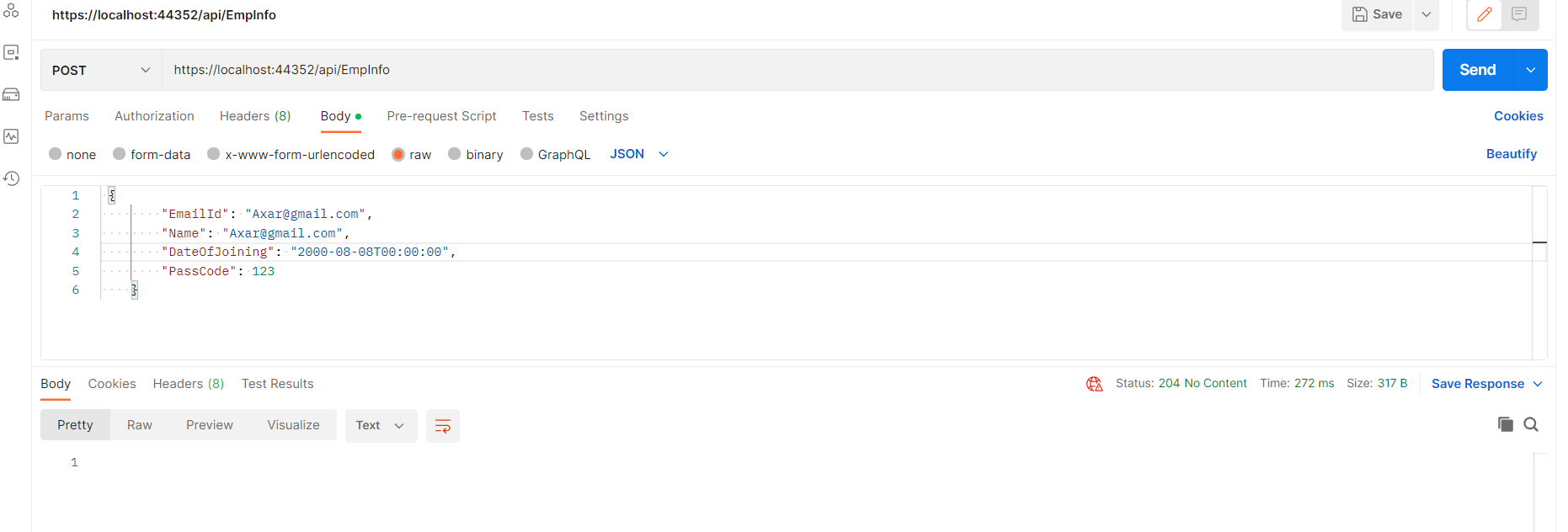
INCORRECTPASSWORD:

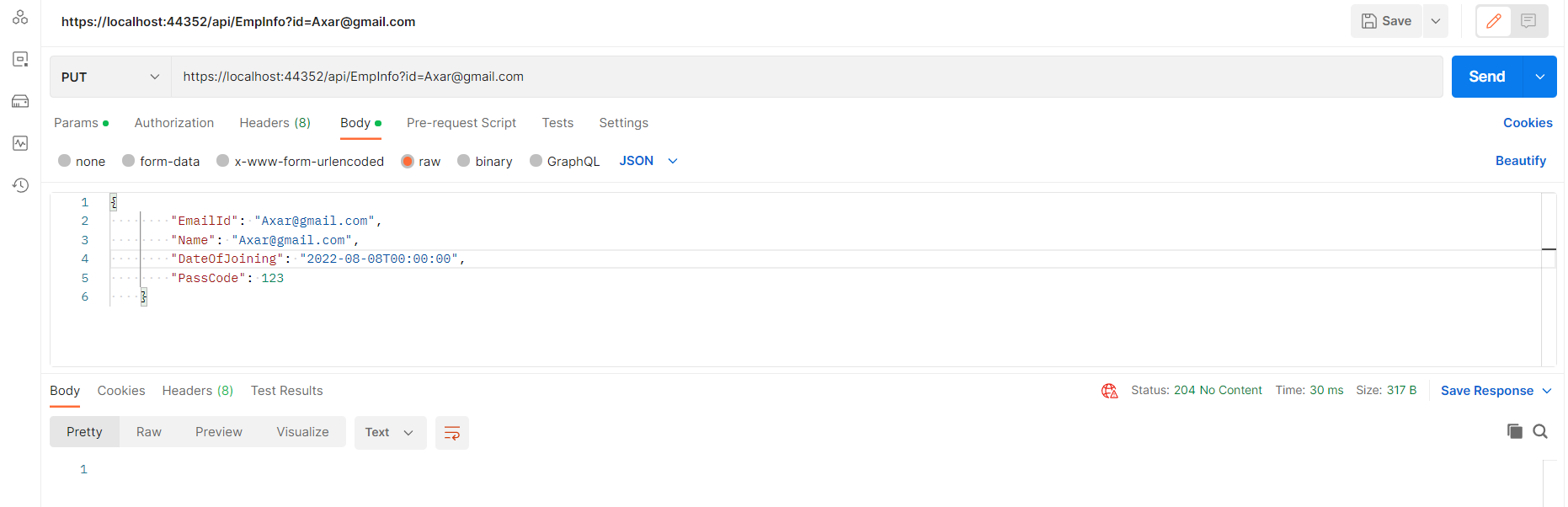




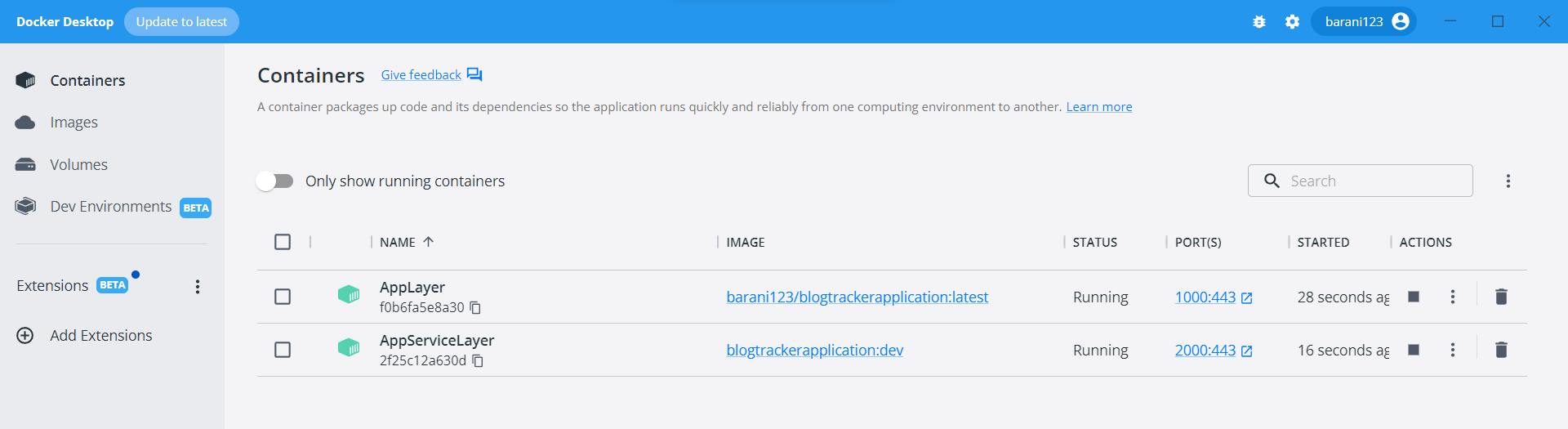
TOTESTWEBAPI:



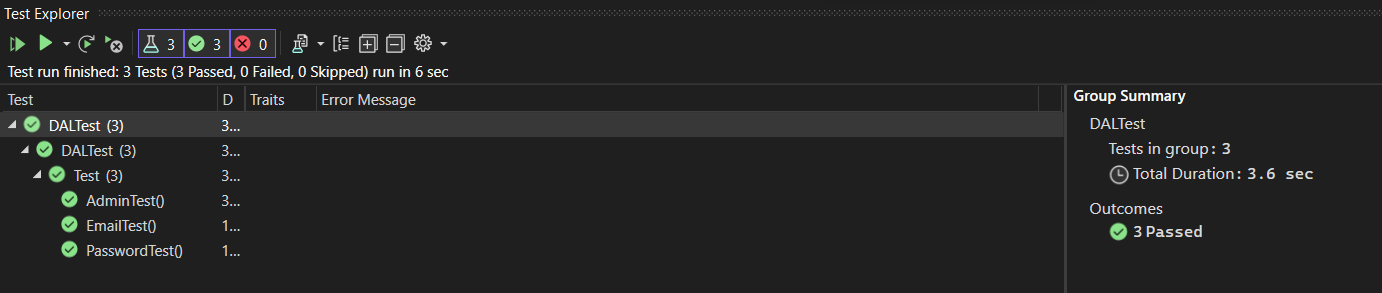




DOCKER:



TEST:



SSMS:  
