

Project: Clarity Ventures Email Application
Author: Richard Howlett

Instructions

Step 1 (Dependency Check):

Build project to ensure dependencies aren't damaged. If there is a namespace error the **EmailService.dll** file is present within the repo.

Step 2 (Local Database):

A database was used to record the emails being sent via my dummy email. A local database is required to set it up, add a database and update the **Connection String** in appsettings.json. The database was managed using **EntityFrameworkCore**, So to add a new migration simply use the "Add-Migration" command and name the initial setup. All models were loaded into **ApplicationDbContext.cs**.

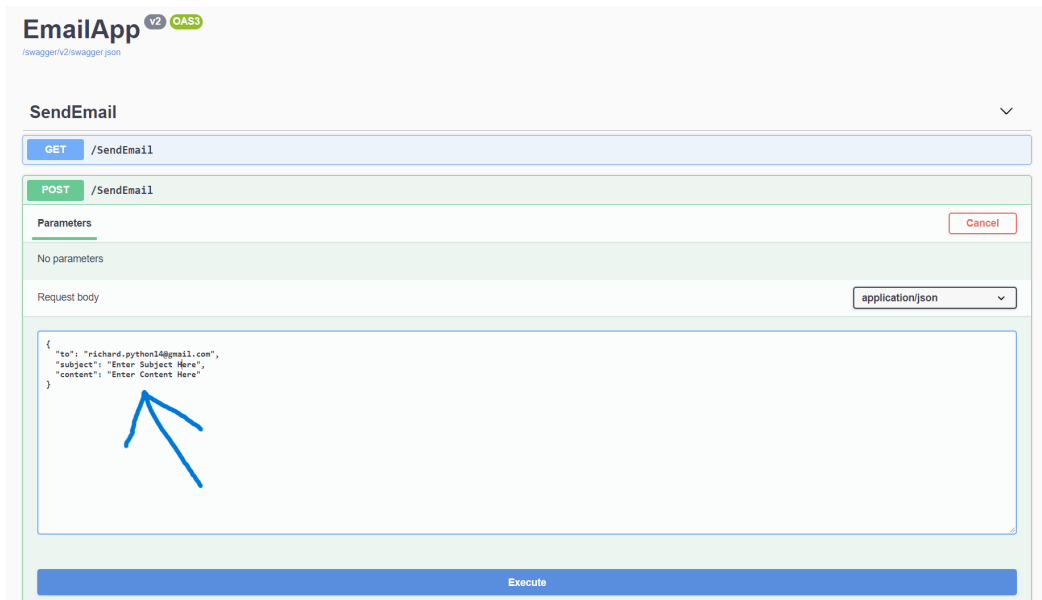
Step 3:

The email address is not using ssl so accept the prompt about the certificate download, else replace the email credentials in appsettings.json with another email of your choosing. If using no ssl please **disable your antivirus** as it might block the email attempts.

The email used was on Google's smtp, so despite having the identity verification disabled google may still prompt for verification. This can simply be overridden if you login using the credentials or provide a different email and adjust the information in appsettings.json.

Step 4:

Start the **EmailApp** once the API UI has loaded you can enter the email information by replacing the “string” portions of the request parameters in the **POST** method.



The screenshot displays the Swagger UI for an application named "EmailApp" (version v2, OAS3). The endpoint being viewed is "/SendEmail". Two methods are listed: GET and POST. The POST method is selected and highlighted in green. Below the method selection, there is a "Parameters" section which is currently empty, with a "Cancel" button to its right. Underneath, the "Request body" section is visible, showing a dropdown menu set to "application/json". The request body is a JSON object:

```
{  "to": "richard.python14@gmail.com",  "subject": "Enter Subject Here",  "content": "Enter Content Here"}
```

. A blue arrow points to the "to" field. At the bottom of the interface is a large blue "Execute" button.

Once you've entered the recipient email ("to"), the subject and the email message ("content"), click the execute button and wait for the response. If there is a response code of 500 there is a possibility that your antivirus is blocking the email please disable it while testing and the issue should be resolved.

A display of results is generated at the end of the **POST** method, but can still be accessed using the **GET** method.