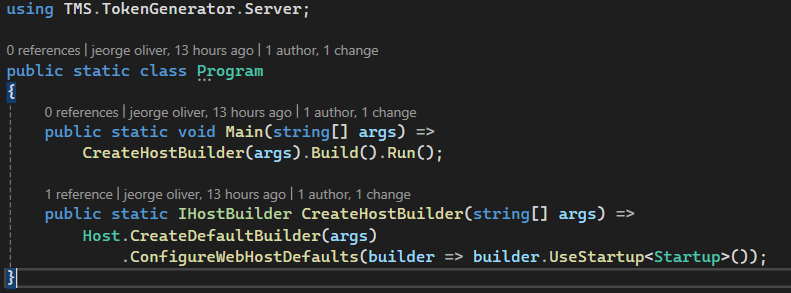
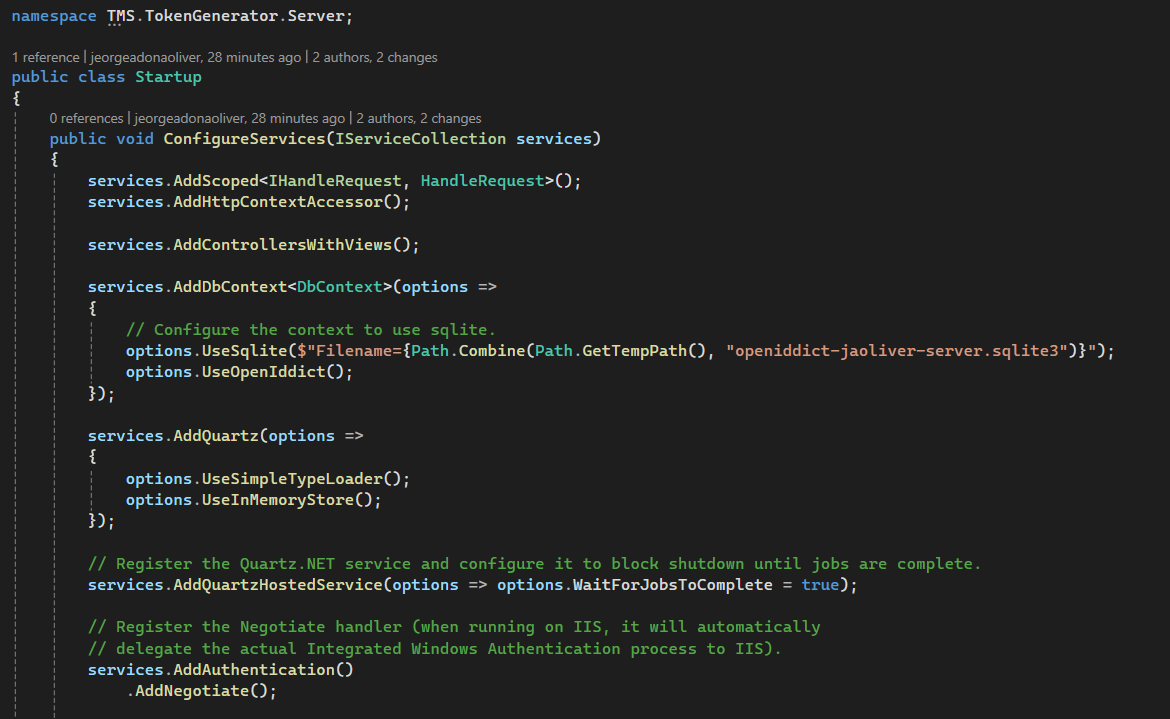
**TOKEN MANAGEMEMT SYSTEM DOCUMENTATION  
  
TMS.Api.Server**

**Program.cs** – this is the entry point of the system where we create **Host Server** for the application(Developers can put another configuration on this class). For this system, **Startup.cs** is the starting configuration.

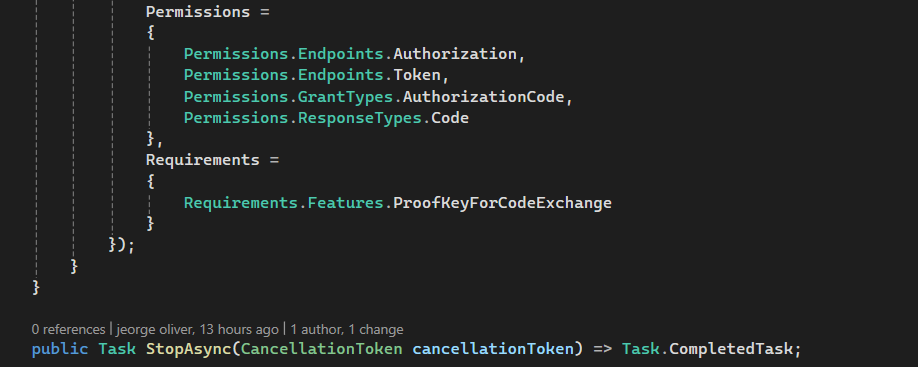
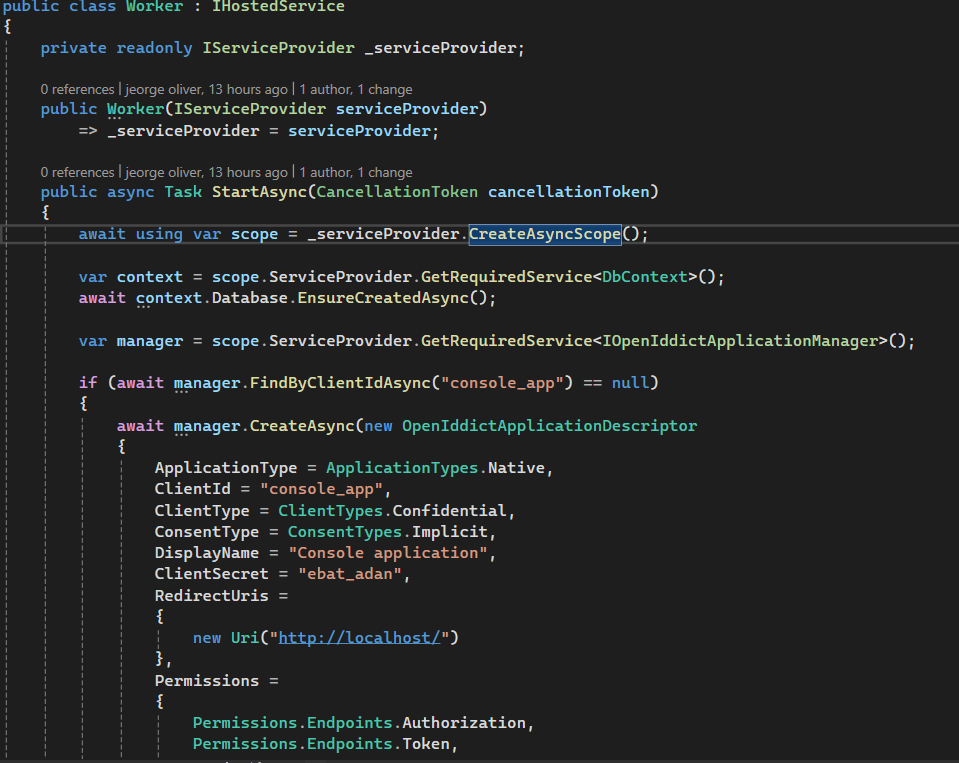


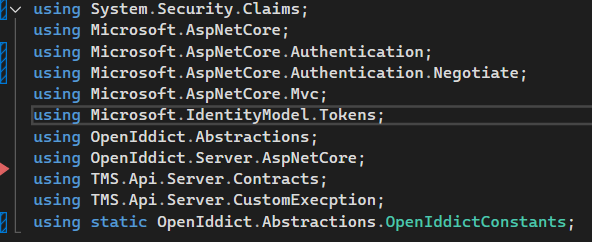
**Startup.cs –** this is where the configuration of the application and policy will take of. Additional details are included as comments In the file for reference and additional modification.

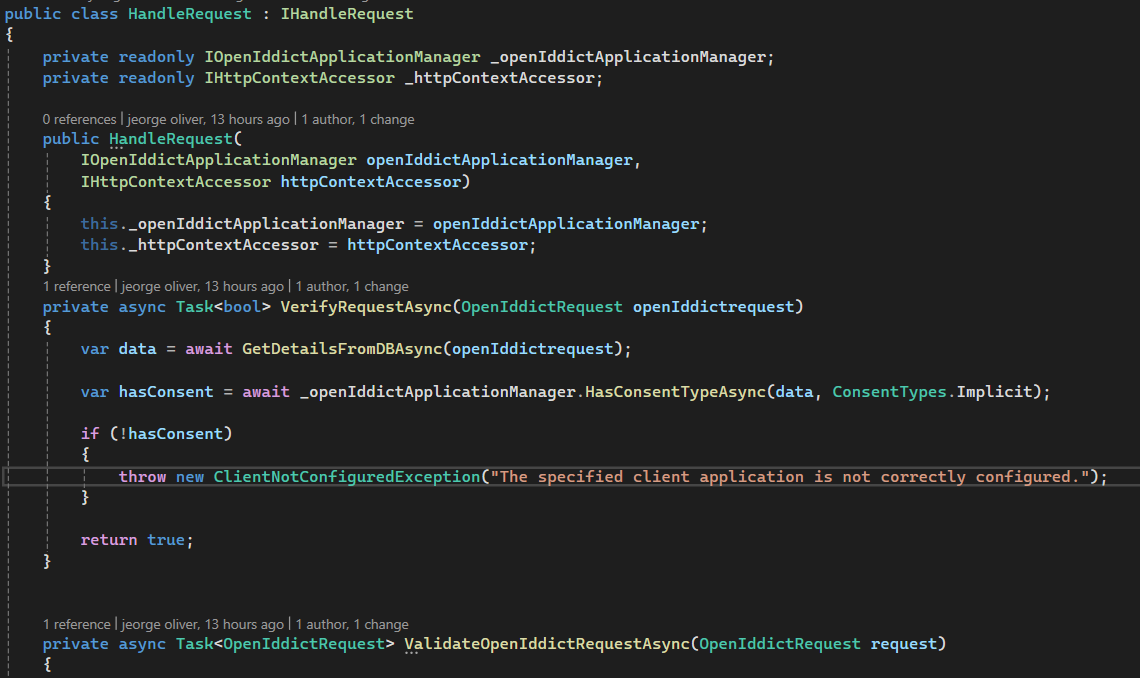
NOTES:

**OPenIdDict** – Oauth 2 open source library for that handle authentication   
 MS SQL LITE – for the database since I don’t have the actual database

**Quartz** – for the task scheduling, Loader (included on the sample implementation. Can modify depending on the implementation or requirements)

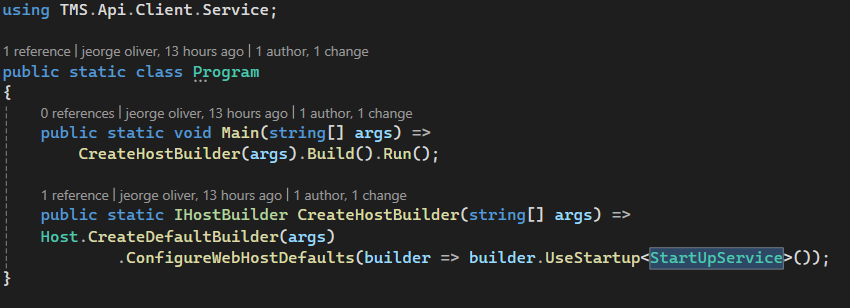
**Worker**.**cs –** Assigning Identification, Permission and Requirements of your App. May change depending on your requirements

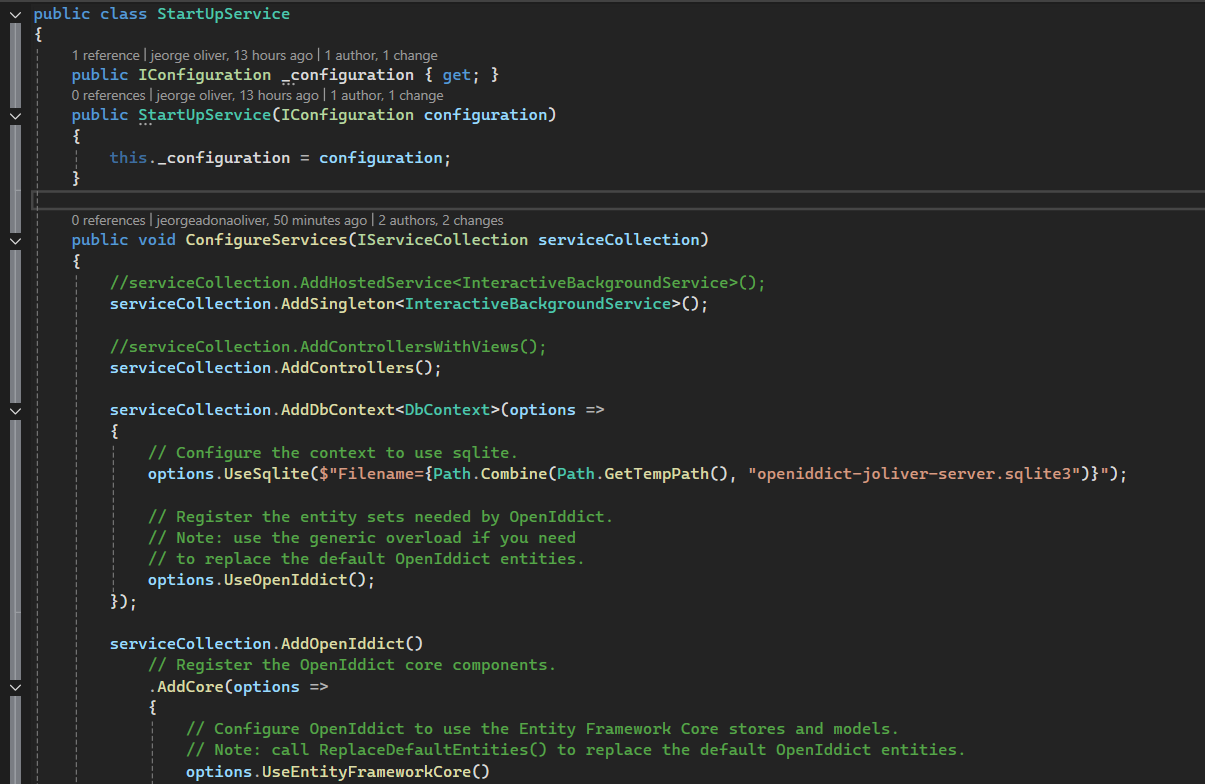
**AuthorizationController.cs** – Implementing Authorize endpoint(“~/connect/authorize”). Some of the methods here are built-in function of different libraries.  
 

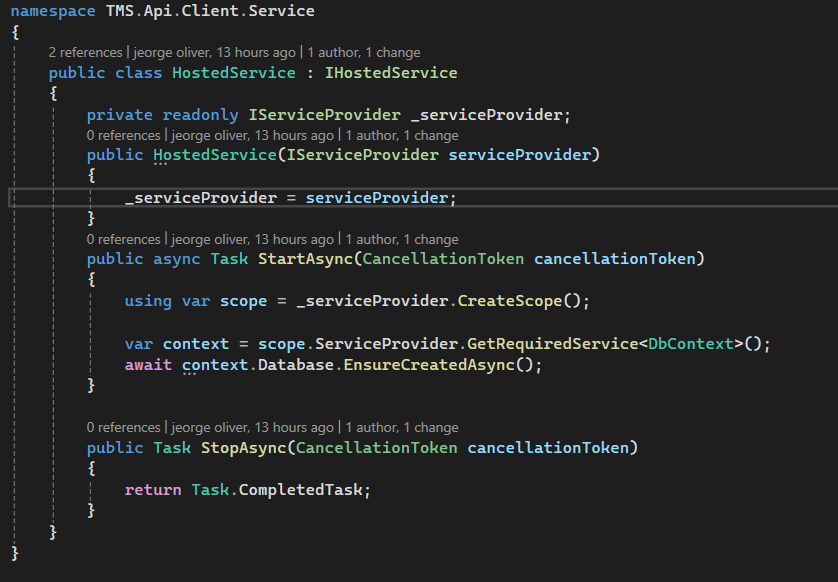
**HandleRequest.cs –** Implementation of the contract **IHandleRequest.cs**. This class will handle request from the Client server. With several private methods for readability and single responsibility. Use Dependency Injection to call this class.  
  


**TMS.Api.Client**

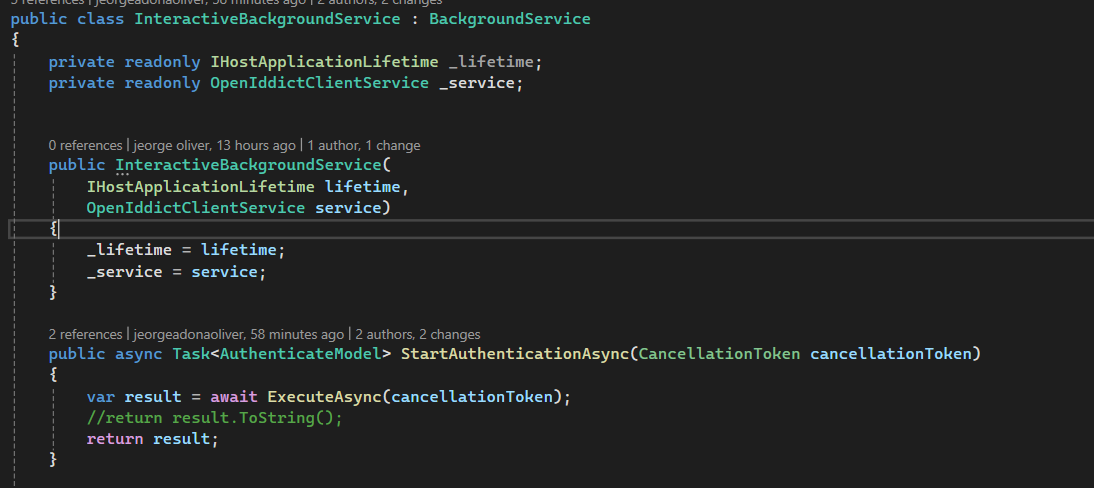
**Program.cs** – this is the entry point of the system where we create **Host Server** for the application(Developers can put another configuration on this class). For this system, **StartUpService.cs** is the starting configuration.



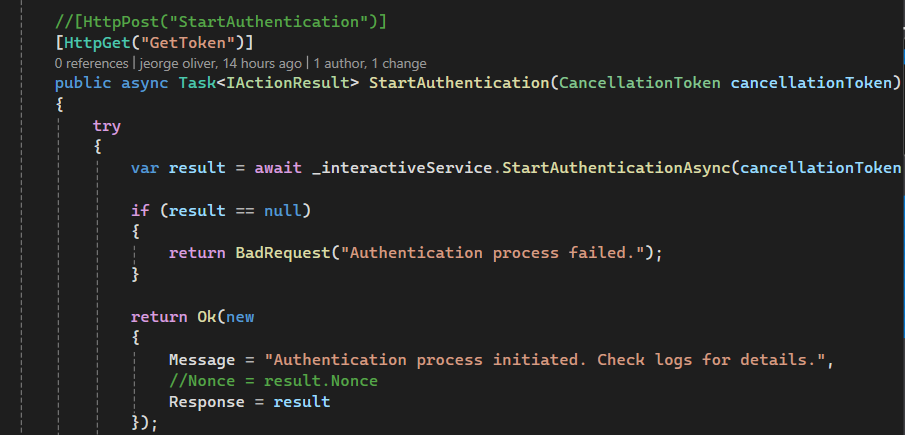
**StartUpService.cs –** this is where the configuration of the application and policy will take of. Additional details are included as comments In the file for reference and additional modification. ****

**HostedService.cs –** hosting/creating instance of the DBContext(using EF SQLITE)  
  


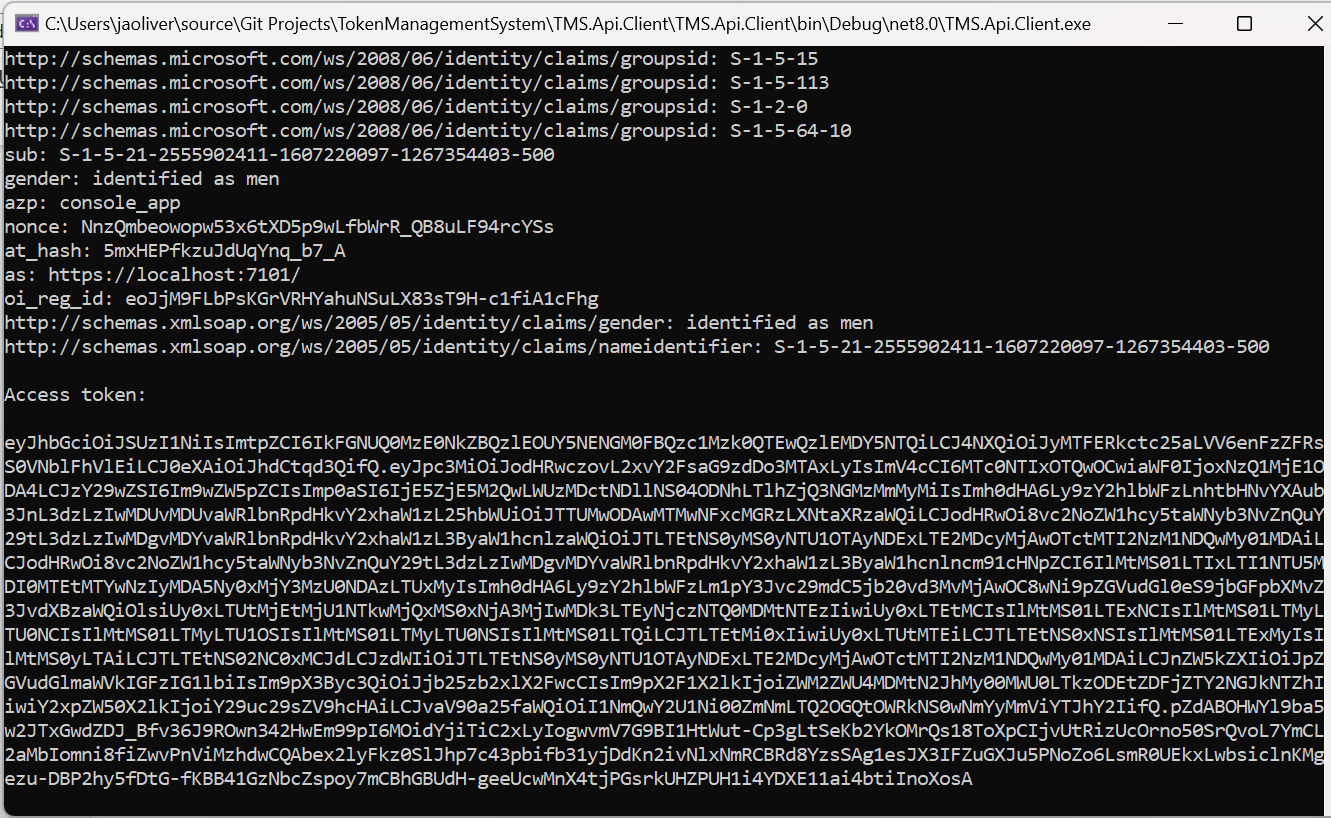
**InteractiveBackgroundService.cs –** this class inherit BackgroundService(Base Class) that will automatically execute the service even without calling the controller.

****

**AuthenticationController.cs –** this endpoint (“/api/GetToken”) will be the endpoint of the TMS.Api.Application.



This is the sample result of the **TMS.Api.Client**

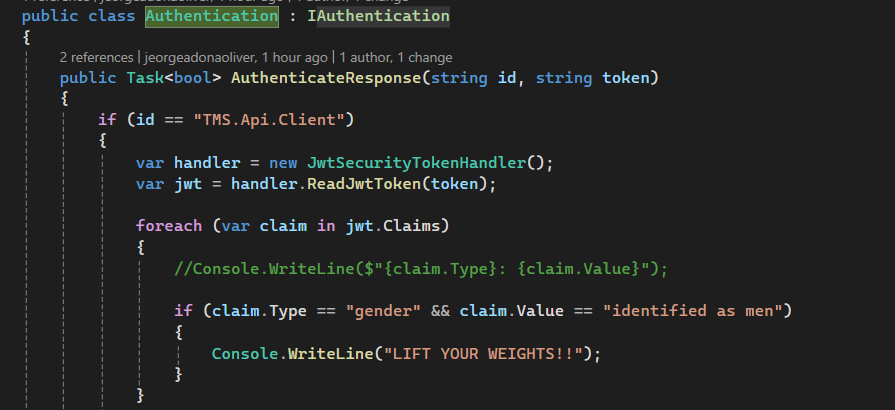
****

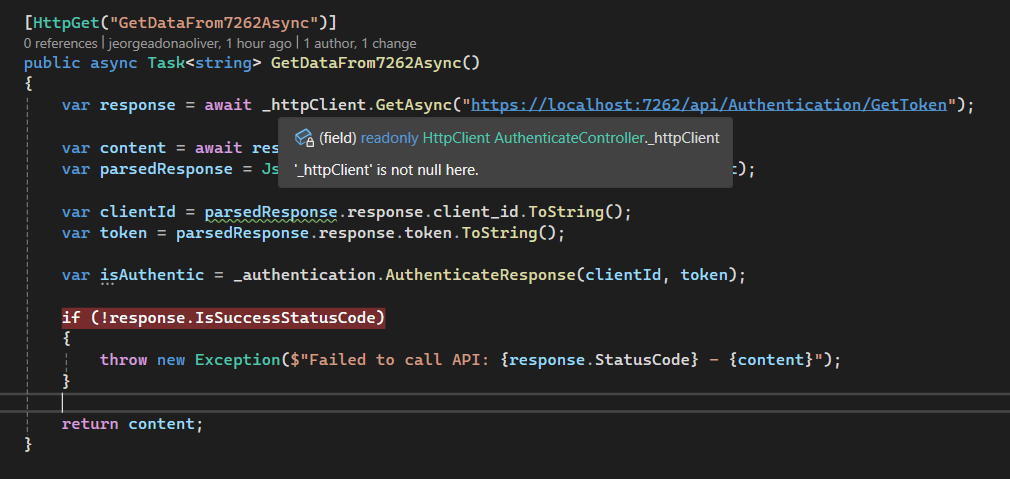
**TMS.Api.Application**

**Program.cs** – this is the entry point of the system where we confiure the system.

****

**Authentication.cs –** This class is the implementation of **IAuthentication.cs** to separate controllers to actual business logic of the app. Use Dependency Injection to call the class.

****

**AuthenticationController.cs –** this class will request(TMS.Api.Client) token to validate.   
  
****

This is the sample result of the **TMS.Api.Application**

