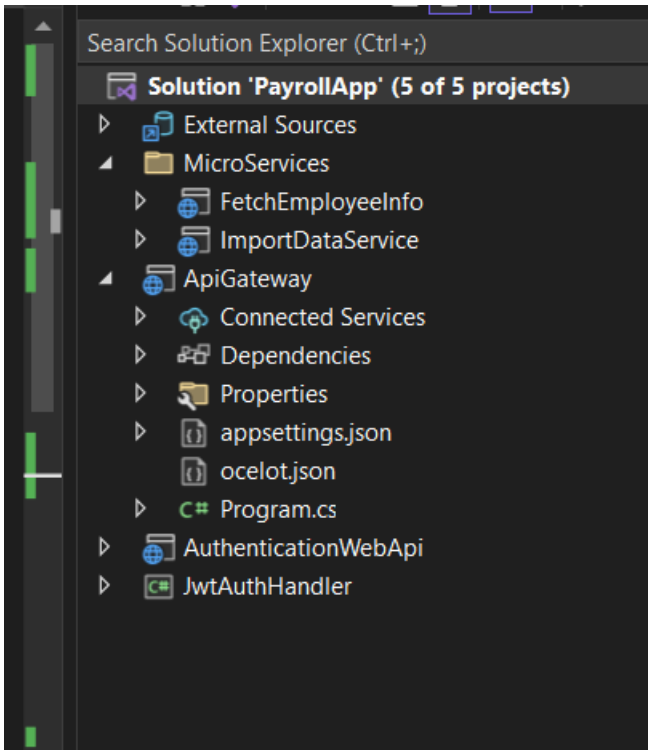
Explanation on the Development

Here in the solution, I used micro services architecture to manage different services to

- Develop (.Net Core/ .Net 6,7, SQL SERVER)
- Use Services in Cloud (Service Bus, Storages, Logic Apps)
- Scale up. (Can change the plan to scale up, down and pay as you go options)
- Deploy (Pipelines in azure, manual deployment for some environments like Prod)
- Release (GitHub, CICD pipelines)
- Manage (Key vaults, Pull request, Branches and Resource Groups, Environments)

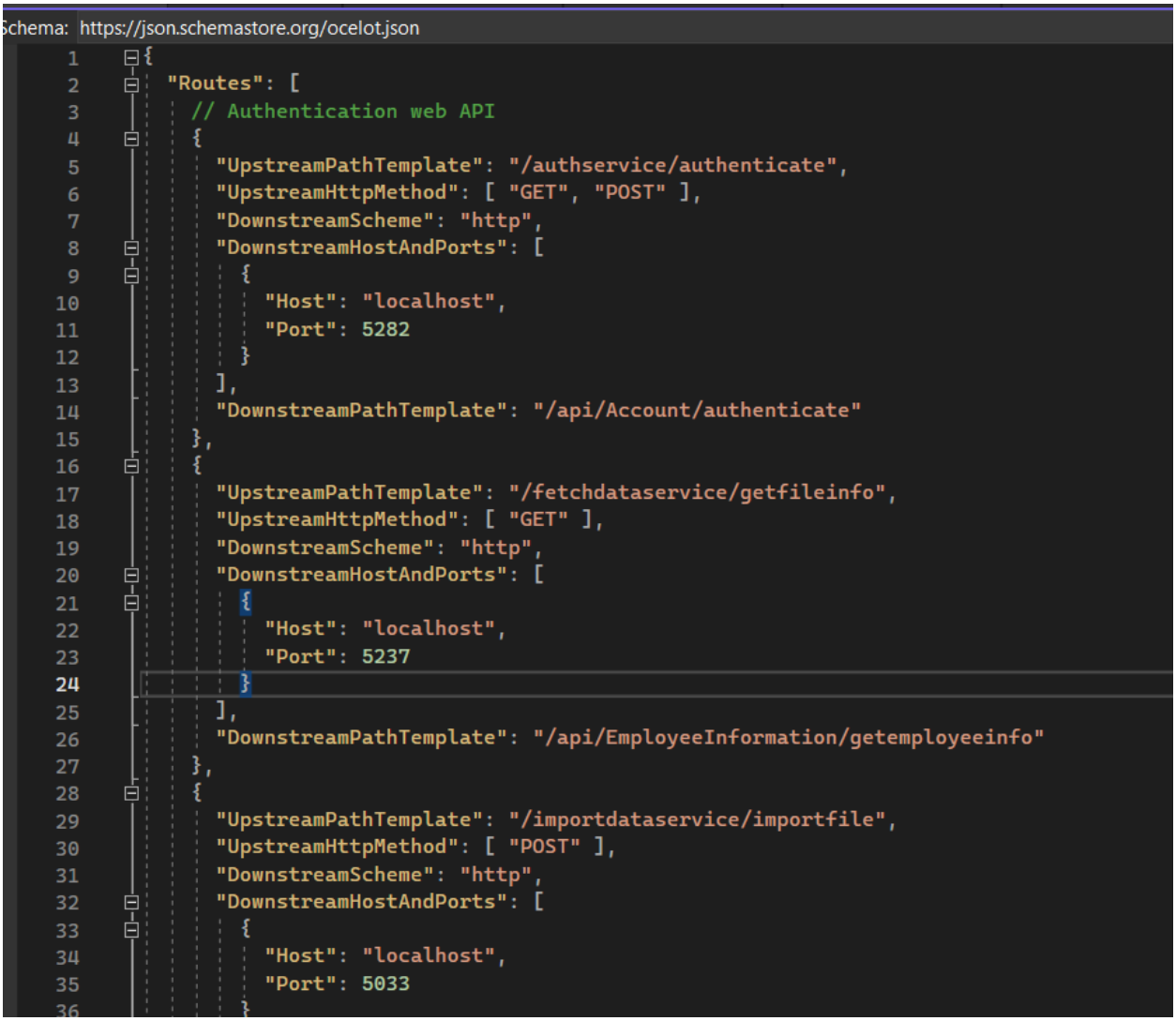
Services independently



FetchEmployeeInfo and ImportDataService are the main services used in the application. API gateway is the main single centralized route provider for the above services.

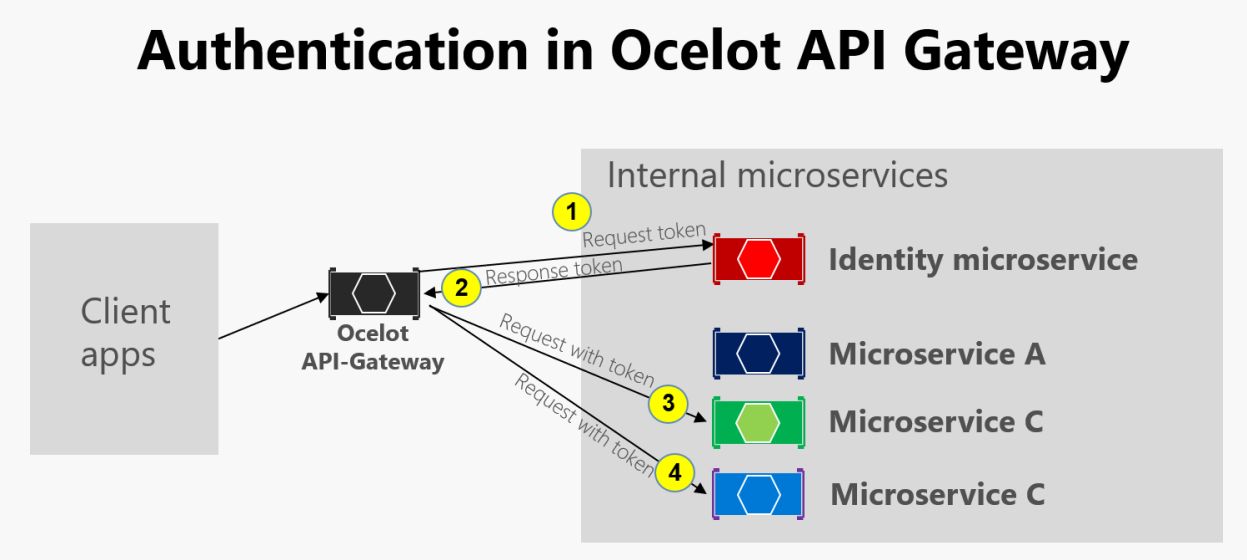
Other authenticate services and Handler app for support authentication and authorization over the app.

Ocelot for API configuration



As seen here Authenticate, Data import and Data retrieve APIs are managed here and used single Jwt authentication to manage security.

For hosting these we can use **Azure App Service** to host each service and manage.

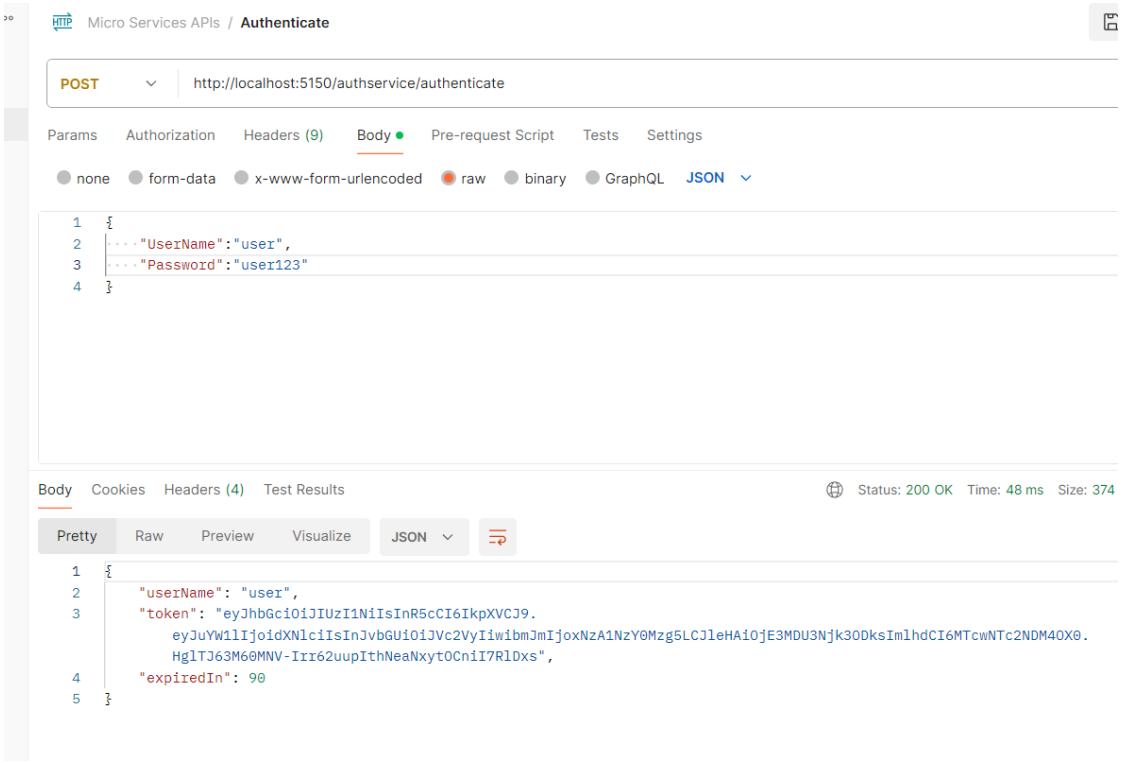


Here is the overall idea how APIs communicate between components.

And here how we used ORM(Entity Framework Core) to get and insert information with database

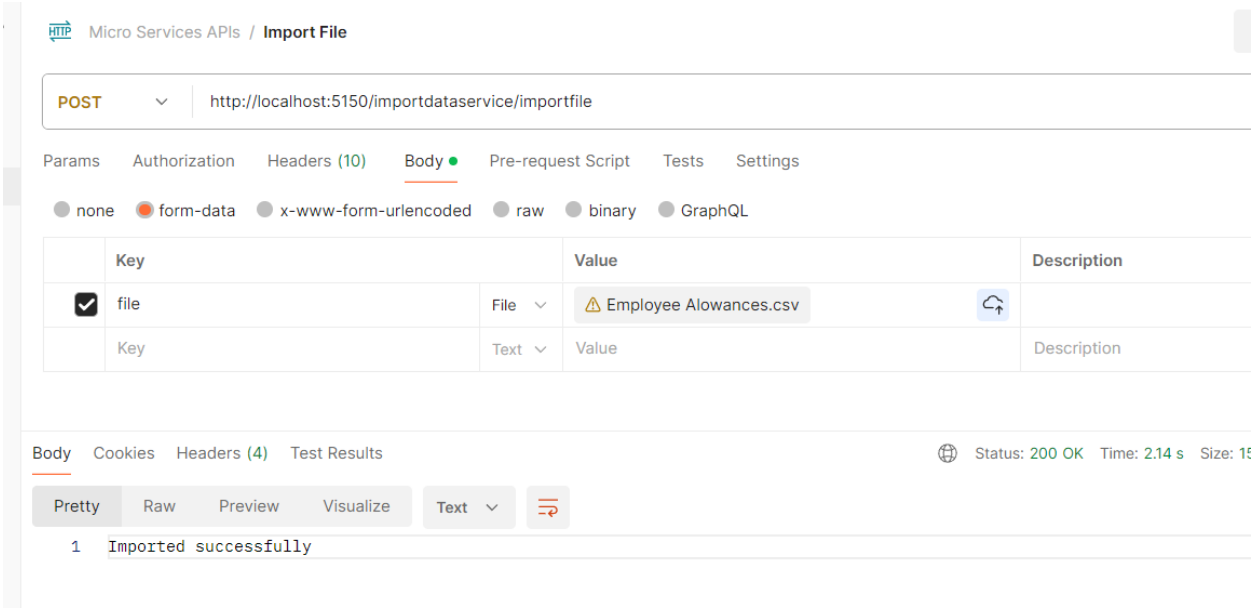
```
namespace ImportDataService.Models
{
    [Table("EmployeeInfo")]
    3 references
    public class EmployeeInfo
    {
        [Key]
        [Column("EmployeeInfoID")]
        0 references
        public int EmployeeInfoID { get; set; }

        [Column("EmployeeID")]
        1 reference
        public int EmployeeID { get; set; }

        [Column("DepartmentID")]
        1 reference
        public int DepartmentID { get; set; }

        [Column("Date")]
        1 reference
        public DateTime Date { get; set; }

        [Column("Amount")]
        1 reference
        public decimal Amount { get; set; }

        [Column("Status")]
        1 reference
        public string Status { get; set; }
    }
}
```

And used Postman to test the APIs. Solution will contain the postman collection to check again.

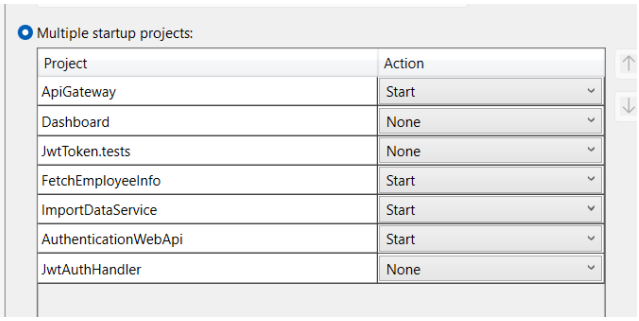


Need to add this token as Bearer <token> in the Authorization header in the other requests,

If the file is correct format and correct type, data will be inserted, and API will return success message

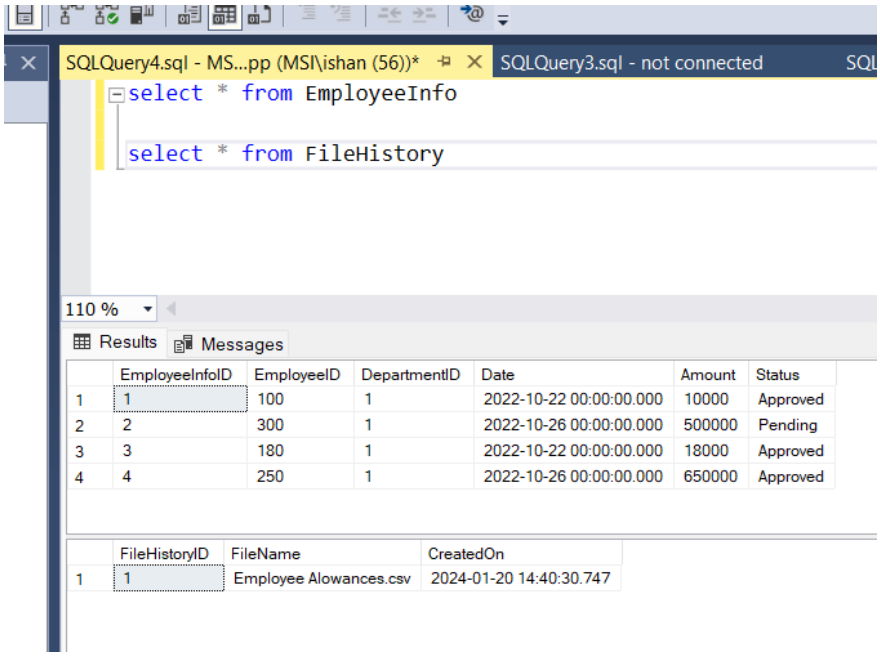


Tried to add Unit tests for some units. I had an issue injecting relevant services in the Unit test solution. Added the solution I tried.

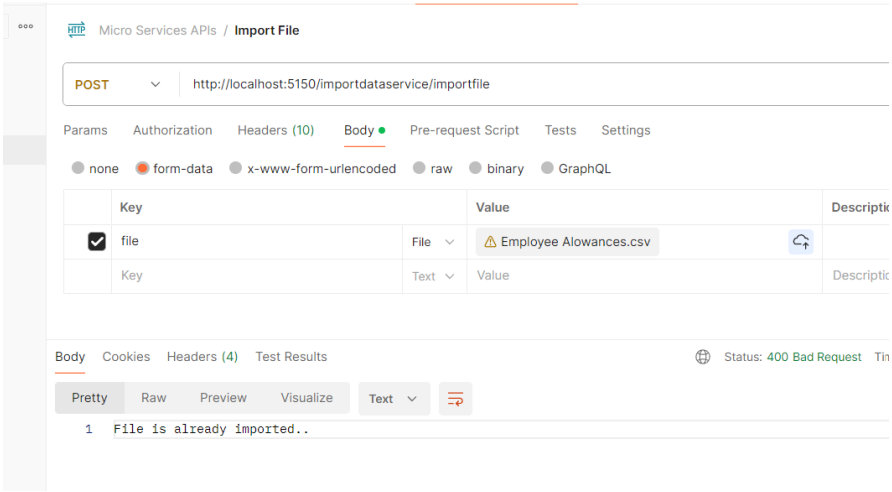


In the startup project we have to select multiple projects to up relevant services

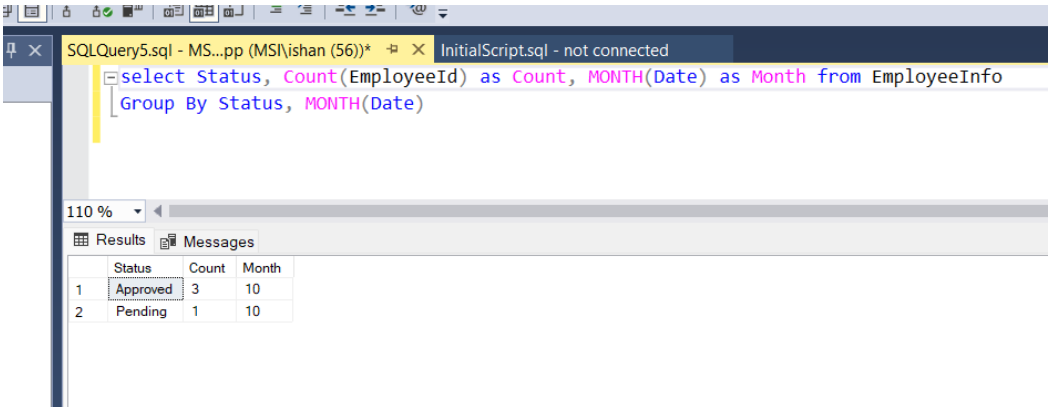
These are the table structure used to store data in MSSQL database.



If we try multiple with same file, it won't import data. Assume that each file contains unique records and that's the assumption and logic here to check. This can be change according to the logic. Below example is how if we get response if we try multiple times



We can create script like this to get 'Approved' and 'Pending' Counts for Specific month and if we have more Fields and more complex queries will be able to write.



Also This can be included In a View or Stored Procedure to store the logic and get data by providing parameters