Scheduler Exercise

This project is implemented with **Web Api Asp.Net Core 2.1.**

We assume this project is a part of Office Automation and we receive Tickets from clients.

Assumptions in Scenario:

We have 3 type of users; A) Higher priorities B) Ordinary Users C) Low Priorities

1- Users of Type A orders, should not take longer than 10 minute with SLA tier: `TenMinute`.
2- Users of Type B orders, should not take longer than 1 hour with SLA tier: `OneHour`.
3- Users of Type C orders, will be done if there is no job remaining with SLA tier: `BestEffort`.
4- If a user of type B sends an order sooner than type A, the priority is with Type A unless, the remaining time of type B sender is exceeding 60 minutes.

In repository layer I have hard coded 9 different users, in UserSeeds.cs file and imported by **Dependency Injection**.

In "appsetting.json" we have two public variables; ServerNumber and ProcessTime. Which can change while program is running.
ServerNumber: The maximum number of Servers which can work for us.
ProcessTime: The amount of time that takes for a doc to be processed.

For making visible changes and logging different message, I have used, "log4net" component; which is adjustable over "log4net.config". In the .config file I have set to store all the logs to a file called "**log.txt**".

All the order objects are stored in a list called "**processQue**". Because this List should be created once and may be used by different requests, I have implemented it by **Singleton.**

The number of servers which are working on our tasks are controlled by another singleton variable called `activeServersCount`.

The main process is done in a class library called Document Processor and it is called by a Web Api project.

In Class library we have two Tiers; Models and Services.

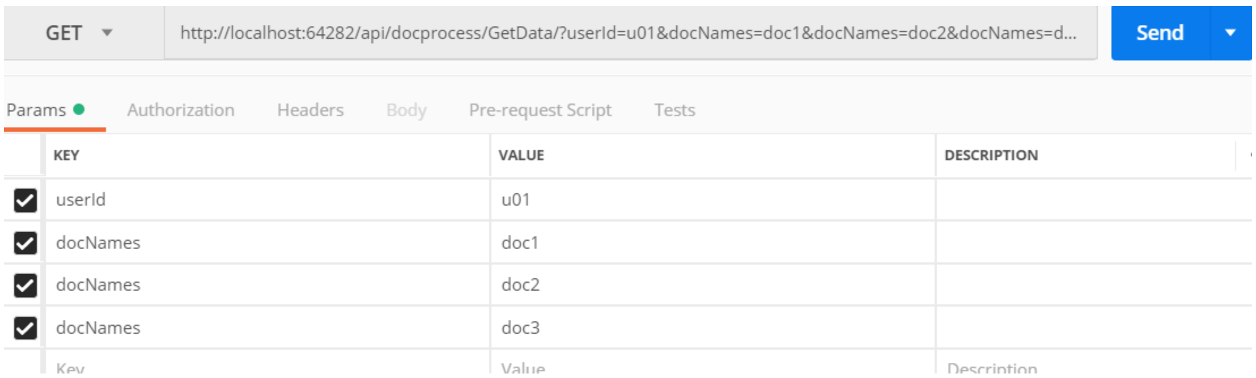
Most of the classes are created in Static mode, because of Asynchronous nature they were following, I preferred to not create new object per request of methods.

Functionality:

We receive orders through URL as a GET method. The suggested format would be:

http://localhost:64282/api/docprocess/GetData/?userId=u01&docNames=doc1&docNames=doc2&docNames=doc3

Or we can use API Development environments like **Fidler** or **Postman**.


GET
http://localhost:64282/api/docprocess/GetData/?userId=u01&docNames=doc1&docNames=doc2&docNames=d...
Send
Params
Authorization
Headers
Body
Pre-request Script
Tests

	KEY	VALUE	DESCRIPTION
✓	userId	u01	
✓	docNames	doc1	
✓	docNames	doc2	
✓	docNames	doc3	
	Key	Value	Description

Different users in repository:

u01, u02, u03	Type A with Ten Minute SLA
u04, u05, u06	Type B with One Hour SLA
u07, u08, u09	Type C with Best Effort SLA

The received data (username and list of Docs) will convert to a Client and a List<DocForPorcess>.

In DocForPorcess we have different properties such as

```
    public SLATiers SLALevel { get; set; }

  public DateTime OrderTime { get; set; }
```

Per task, we create a list of tasks with limitation of size of Server Allowed.

For example, if the Server Allowed is 3, we can run the tasks simultaneously.

The rest of the tasks should wait, until the first ones (the highest priority ones) are over.

Execution Sample:

After running program if we insert this url:

http://localhost:64282/api/docprocess/GetData/?userId=u01&docNames=doc1&docNames=doc2&docNames=doc3

Here is a sample of Log.txt for user u01 which is a High priority member who have sent 3 docsuments: doc1, doc2, doc3.

```
2018-11-08 09:06:22,091 [11] - Client u01, with access level TenMinute, has added 3
tasks.

2018-11-08 09:06:26,430 [12] - Start working on doc1

2018-11-08 09:06:26,430 [7] - Start working on doc3

2018-11-08 09:07:26,432 [7] - Finish working on doc3

2018-11-08 09:07:26,432 [12] - Finish working on doc1

2018-11-08 09:07:26,532 [11] - Required servers are 2 and we have 2 so we have to wait
for releasing servers.

2018-11-08 09:07:26,532 [11] - Items remaining in que are 1 and count of active CPUs is 2

2018-11-08 09:07:26,532 [12] - Start working on doc2

2018-11-08 09:08:26,533 [12] - Finish working on doc2
```