# Job Scheduler

## Description:

Implement a code component that enables different modules in the system to run tasks at various times.

Examples of such needs:

* Reporting module runs a daily report
* Maintenance module runs a daily backup

## Scenario

A module in the system needs to run a daily operation at a certain time.

The module registers the specific timing and task with the Scheduler, and the Scheduler executes the job at the specified time.

Example: Maintenance module registers a job to run a Backup every day at 01:00 AM

## Requirements:

1. Receives a Job which details:
   1. When to run (for simplicity: support daily execution at a given time)
   2. What to run (the task to be executed)
   3. Number of occurrences (allow limiting the number of times a job runs)
2. Run the job when it reaches the execution time
3. The jobs registered in the Scheduler should be persistent and survive a restart.
4. Implement a feature that allows users to view or retrieve the current state of the scheduler, including all registered jobs and their details
5. Use the built-in Dependency Injection (DI) container of Microsoft to register and inject the scheduler component
6. Implement a usage example through Unit Test

## **Implementation** Guidelines

1. Pass cancellation tokens wherever possible throughout the implementation.
2. Scheduler job handlers should use dependency injection. For example, use ILogger instead of Console.WriteLine in the handler.
3. The scheduler should support creating jobs with both a limited number of occurrences and unlimited occurrences.
4. When retrieving job details, include relevant historical information about previous job executions.
5. The scheduler can be implemented as a Web API or a console application.
6. For persistence, choose between
   1. Using a database with any provider supported by Entity Framework Core.
   2. Implementing persistence using the file system.

If using the file system, use JSON files and write the output using indentation.

Please submit your solution as a Git repository with a [README.md](http://readme.md/) file explaining how to run and test your implementation