* **Requirements for the use of written APIs**

All three tasks are included in the written solution. I will explain them in order.

1. **Task1**

* The project is based on NET6.
* There is no DbContext; it can run just by starting the project.
* There is Swagger as a UI.
* There is a controller with two actions to extract the data, online based on the Uri and offline based on the file attached to task number one (included in the project).
* For online mode, you need to copy the link of your browser of “Booking.com/everything” and send your request by Swagger.
* For offline mode, there is a settings file in the project which is appSettings, as you know, and by changing it, you can extract other files too.

1. **Task2**

* The project is based on NET6.
* The architecture I considered for this project is Clean Architecture Pattern.
* There is a DbContext, and you should change the “ConnectionString” in the appSettings file before running. There is a mechanism of auto migration after running the first time the Database is created.
* There is Swagger, and you can upload your “task 2 – hotelrates.json” file to the API.
* There is a controller with two actions, the first is for downloading the Excel file by calling the API, and another is for creating a job for sending a file at another time.
* There is a mechanism to automate the process and send files by Email, which Hosted Service implements, starting after the projects run. There is an Interval for checking Database for unsent records.
* There are some settings in the appSettings file (ReportSettings) that can help change the setting by your data, like the Interval and email properties.

1. **Task3**

* The project is based on NET6
* There is no DbContext; it can run just by starting the project.
* There is Swagger, and you can upload your “task 3 - hotelsrates.json” file to the API.
* The ArrivalDate is like “2016-03-15T00:00:00.000+01:00” because the type is DateTimeOffset
* **As you mentioned in the assignment file, the UnitTests are written and executable.**