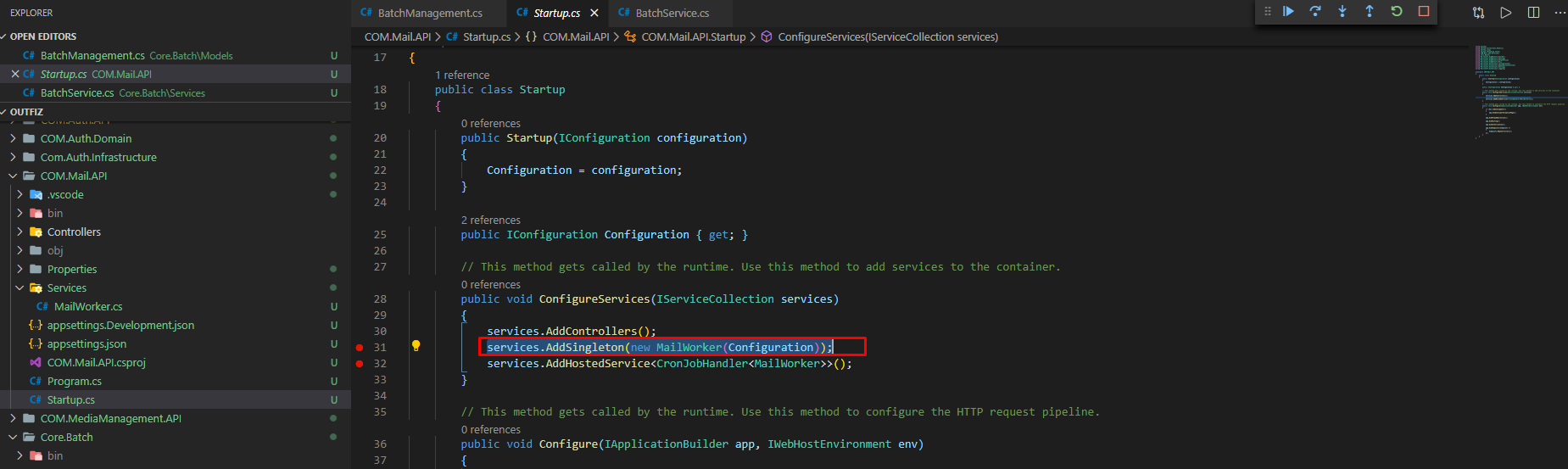


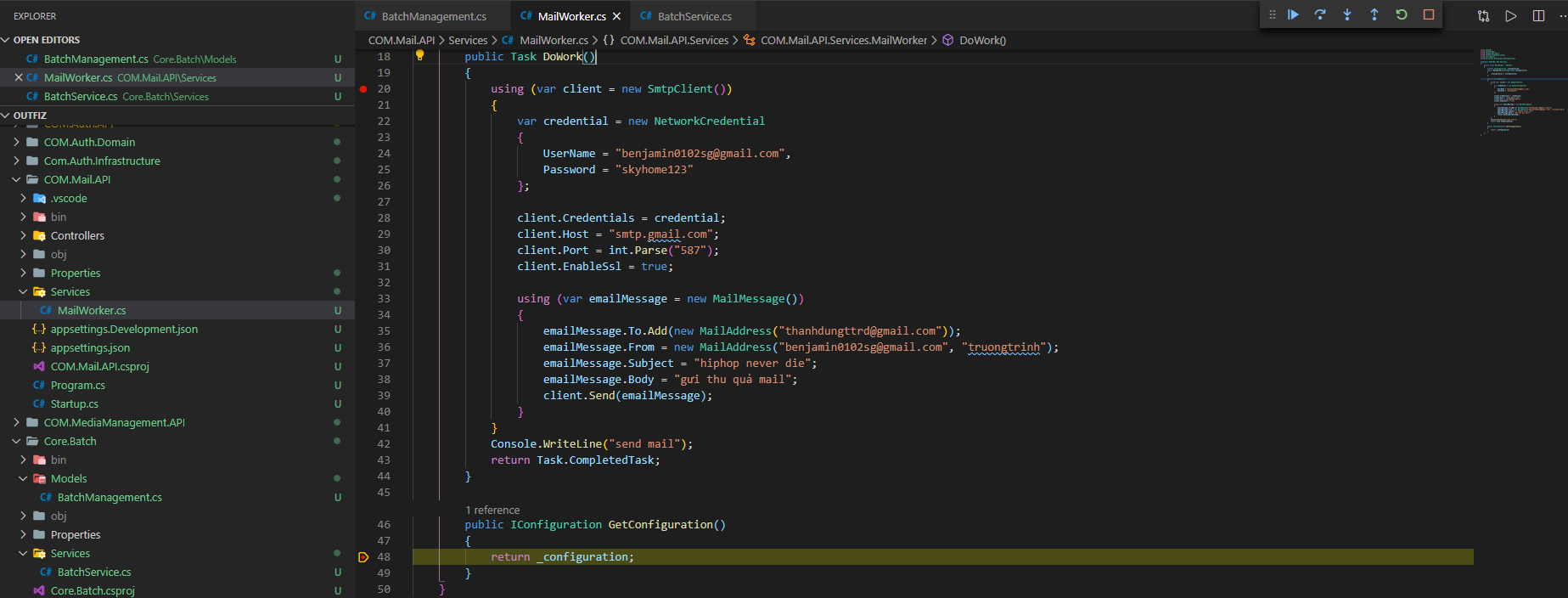
Class MailWorker đưa Iconfiguration vào bằng contructor ( đây không phải là tiêm, bì chưa register Iconfiguration tại startup)



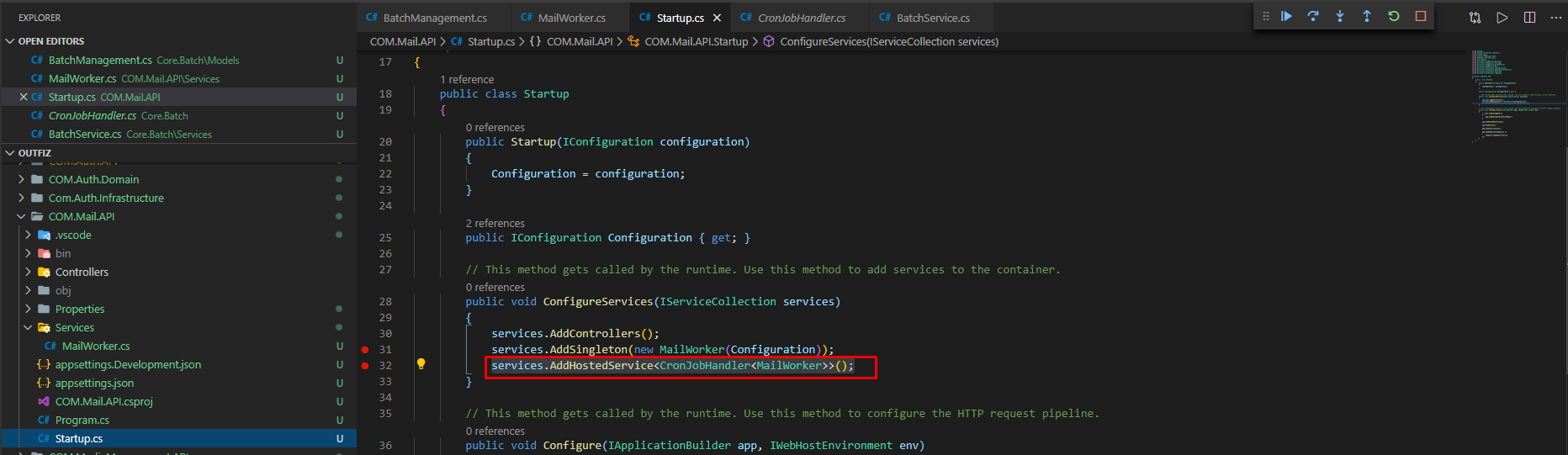
AddSingleton object MailWoker và đưa Configuration vào bằng contructor

Khi đó object MailWorker trên toàn bộ app là object lúc add vào này, và có chứa Configuration

//

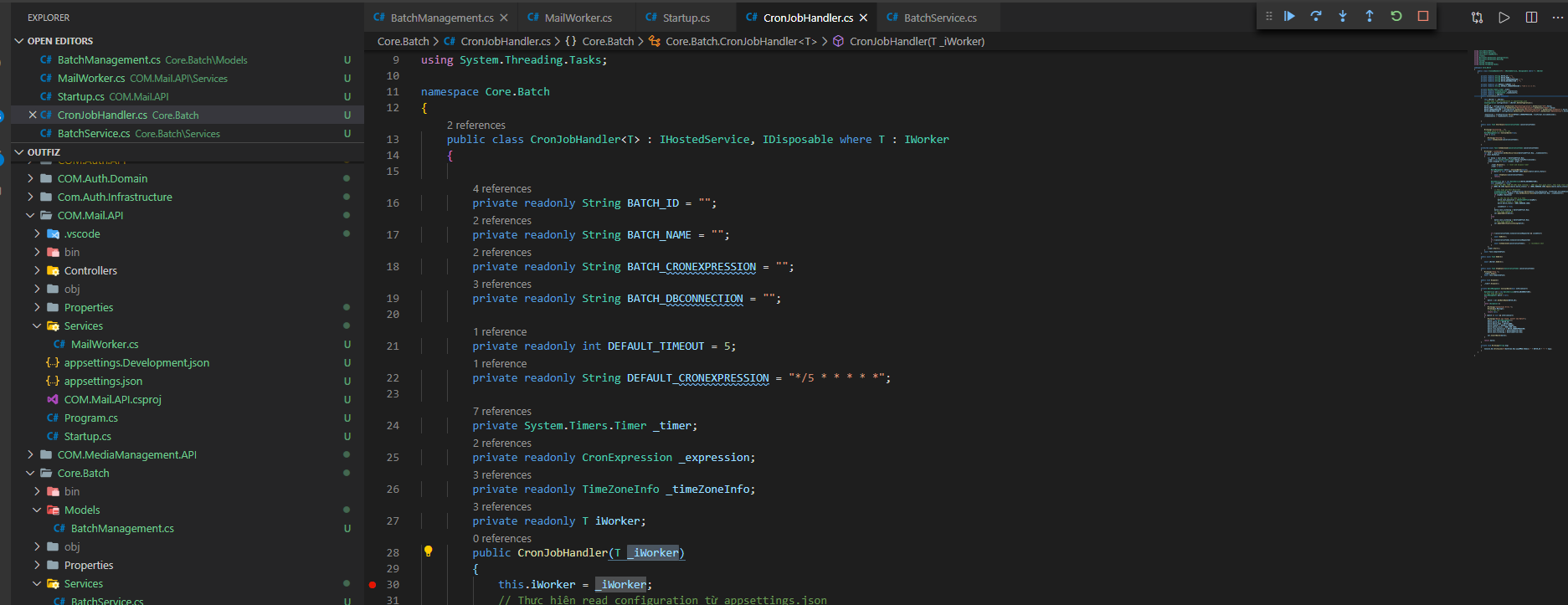


Trên MailWorker có class GetConfiguration từ class MailWorker có thể lấy configuration = Mailworker.Getconfiguration()



addHostedService khi đó contructor của CronJobHandler được khởi chạy với generic được truyền vào là MailWorker

lúc này vì đã addSinglerton object MailWoker ở trên, nên object MailWorker được tạo ra trên cả app chính là object được add ở trên



CronJobHandler có generic T là kiểu IWorker

Kế thừa từ IhostedService và sử dung Idisposable

BATCH\_ID là id của batch ( sẽ được ghi vào bảng batch\_management)

BATCH\_NAME là tên của batch

BATCH\_CRONEXPRESSION: mô tả thời điểm chạy batch

BATCH\_DBCONNECTION: DBStringConnection

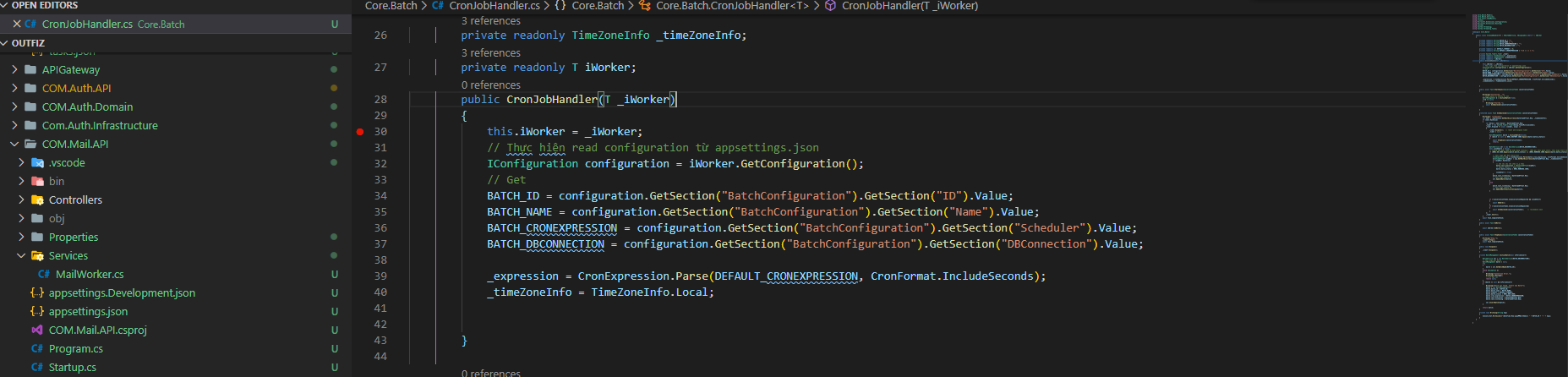
DEFAULT\_CRONEXPRESSION: giá trị mặc định mô tả thời điểm chạy batch

\_timer:

\_expression: Giá trị để config thời gian theo giay giờ phút

\_timeZoneInfo: lấy giá trị timezone

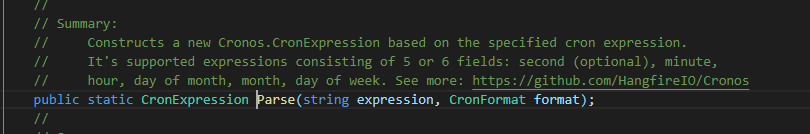
iWorker: generic thừa kế từ IWoker

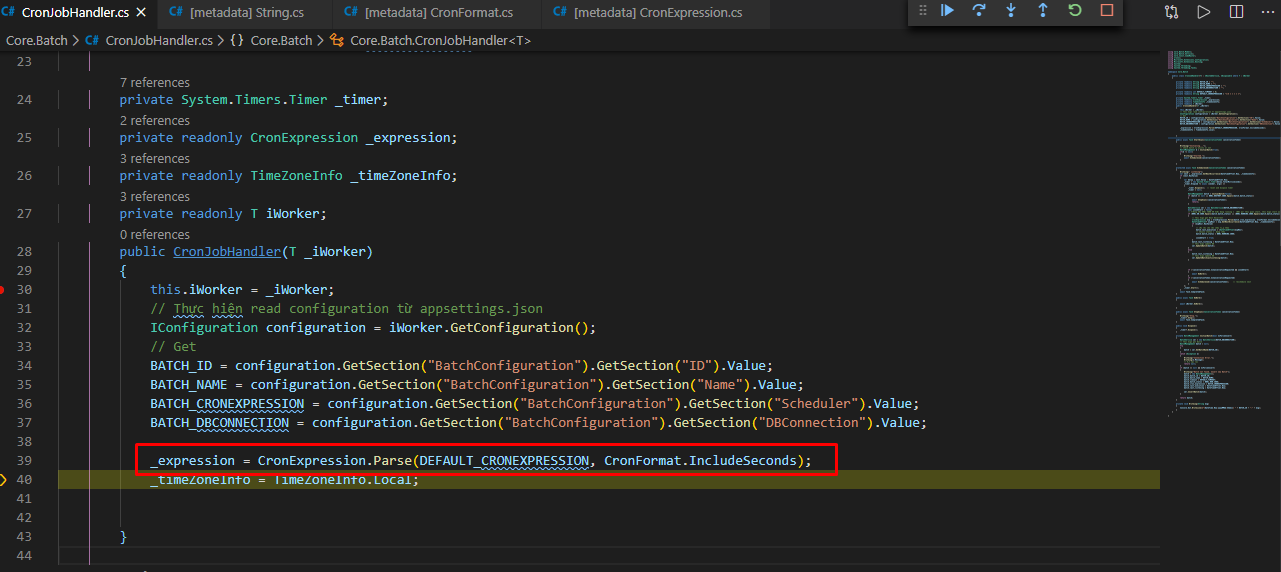


Khi services.AddHostedService<CronJobHandler<MailWorker>>(); thì contructor của CronJobHandler trên classlib core.bath được khởi chạy, với generic truyền vào

Ở contructor truyền vào T \_Iworker, thì Iworker ở đây chính là object MailWorker đã được addsingleton ở trên. Iworker được khai báo biến private ở trên để sử dụng trong class này

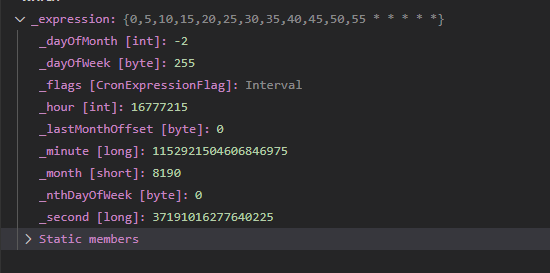
Khi đó dùng GetSection có thể get được các section trong file appsetting.json trên APIApp dung classlib này

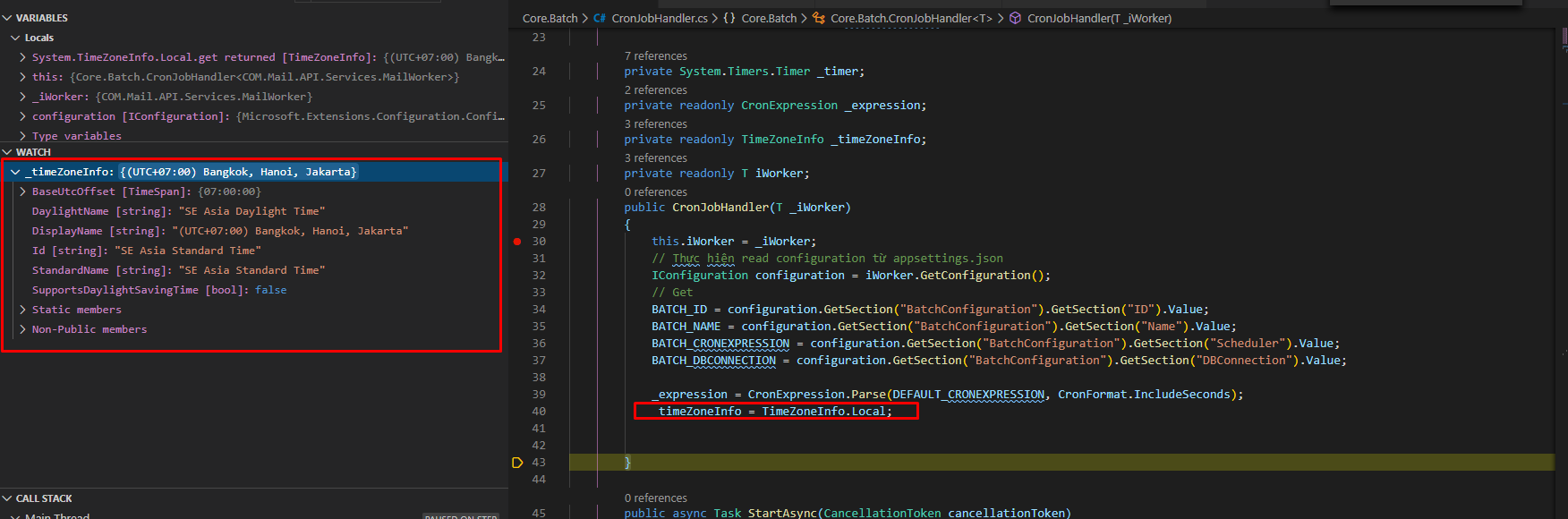




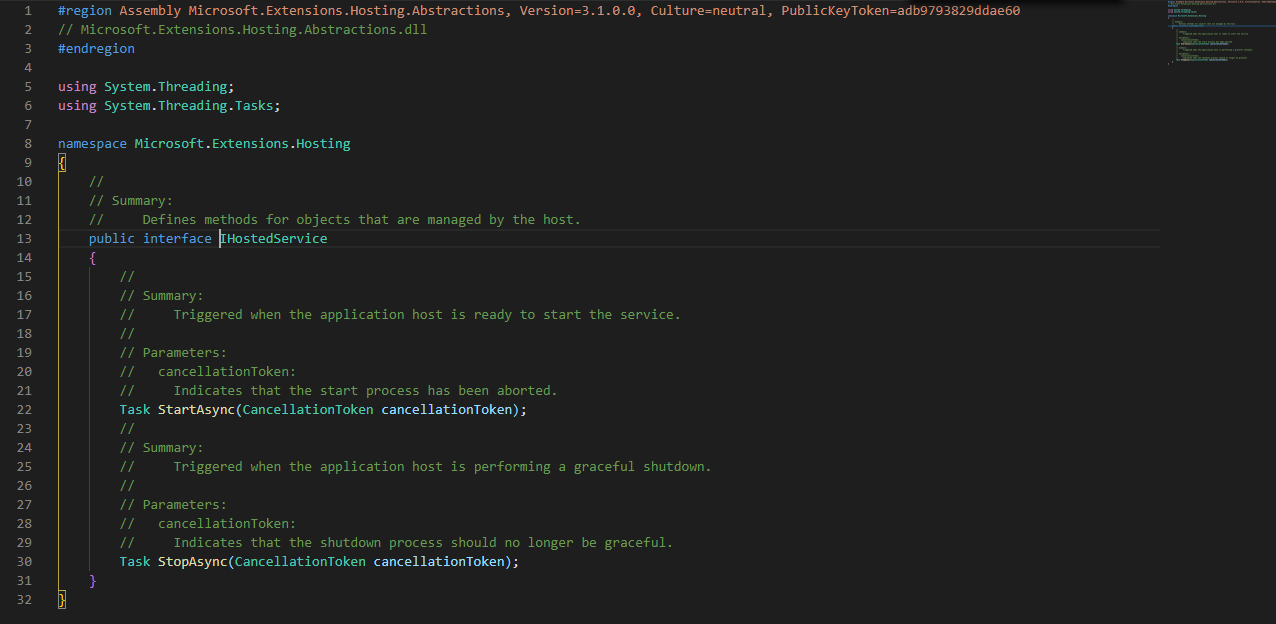
Format ở đây đưa thêm second vào

\_expression

\_expression có kiểu CronExpression chứa các thông số config cho ngày tháng



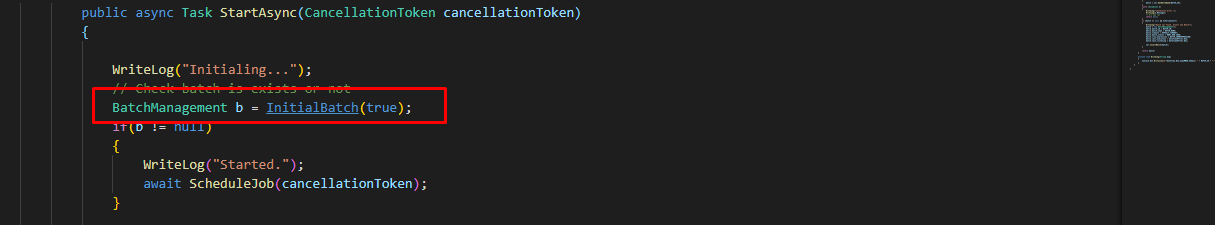
timeZoneInfo có kiểu timeZoneInfo chứa các giá trị của timezone



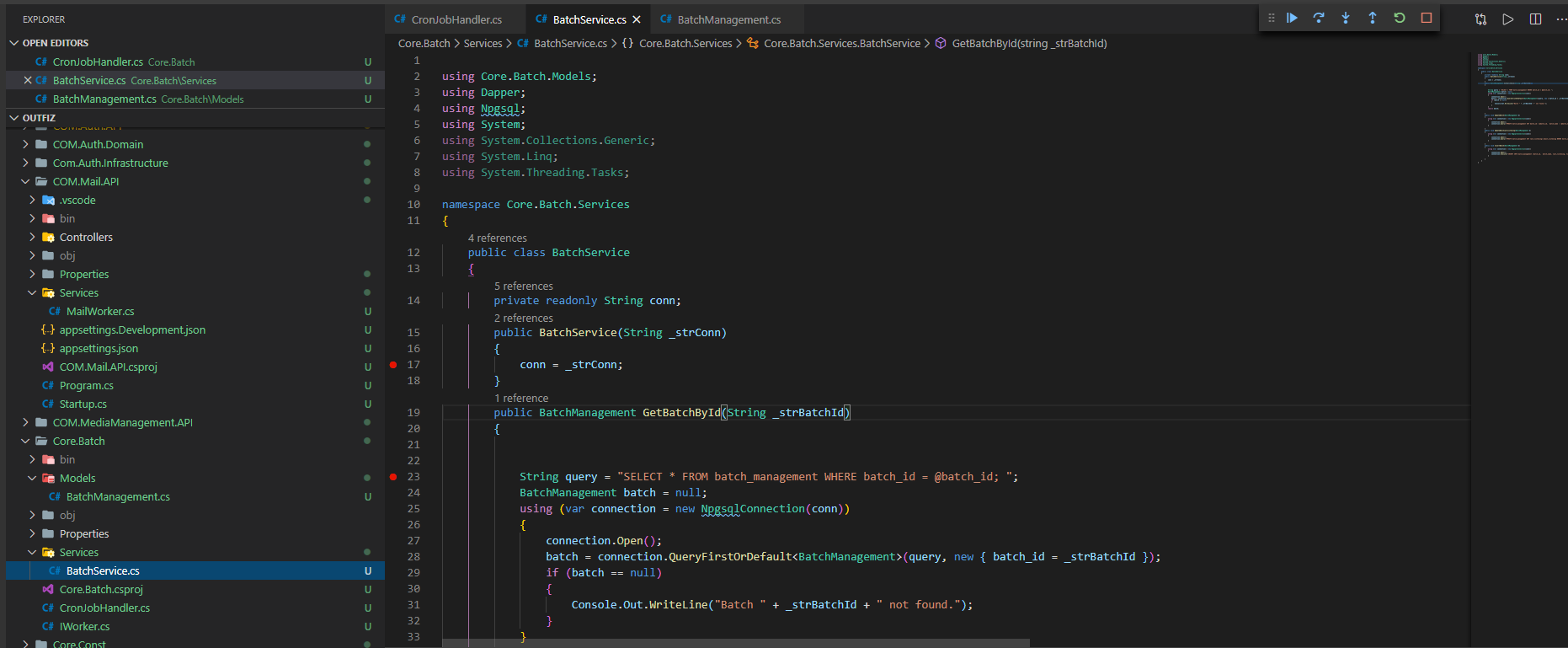
Trong IhostedService

Hàm StartAsync được kích hoạt khi add host và service ( chạy sau contructor)

StopAsync chạy khi app chạy xong và shutdown

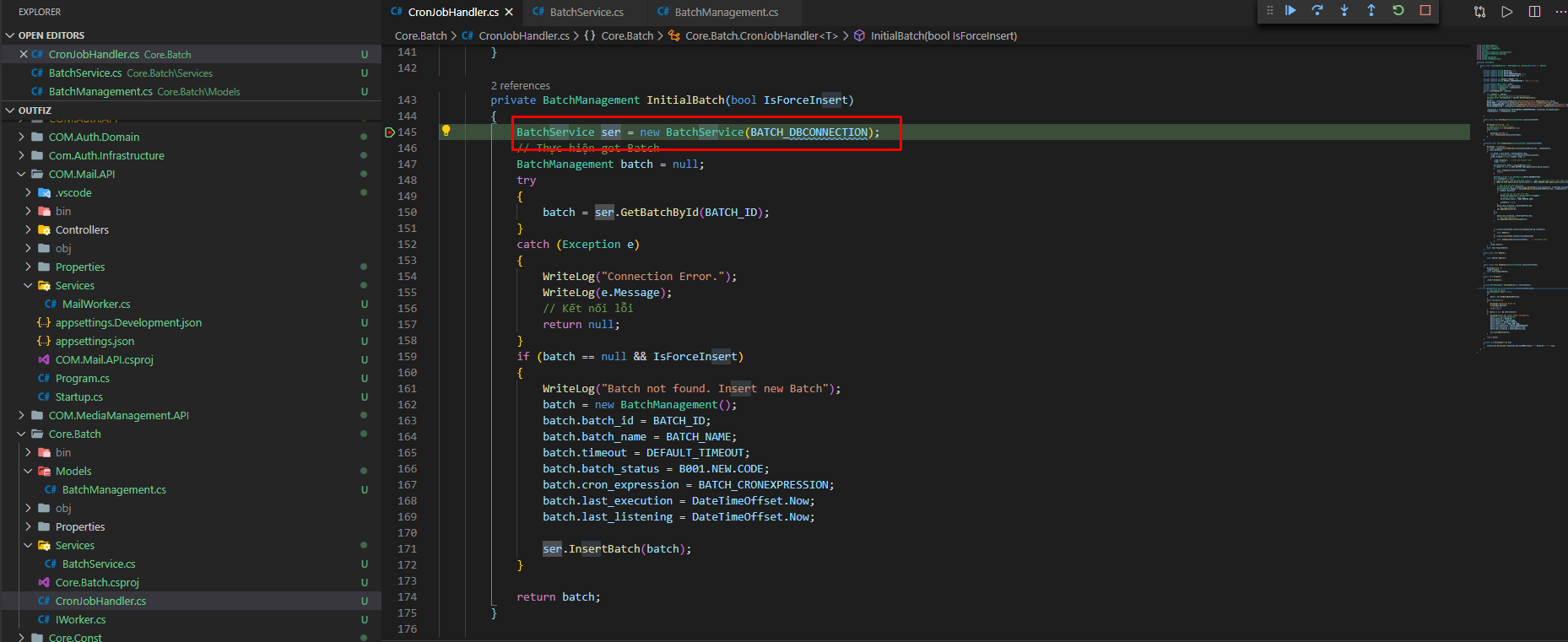


Gọi hàm InitialBatch(true)



Các hàm thao tác batch với db được viết trong batchService

Và thao tác với server sử dụng dapper (tham khảo cách viết trong source)

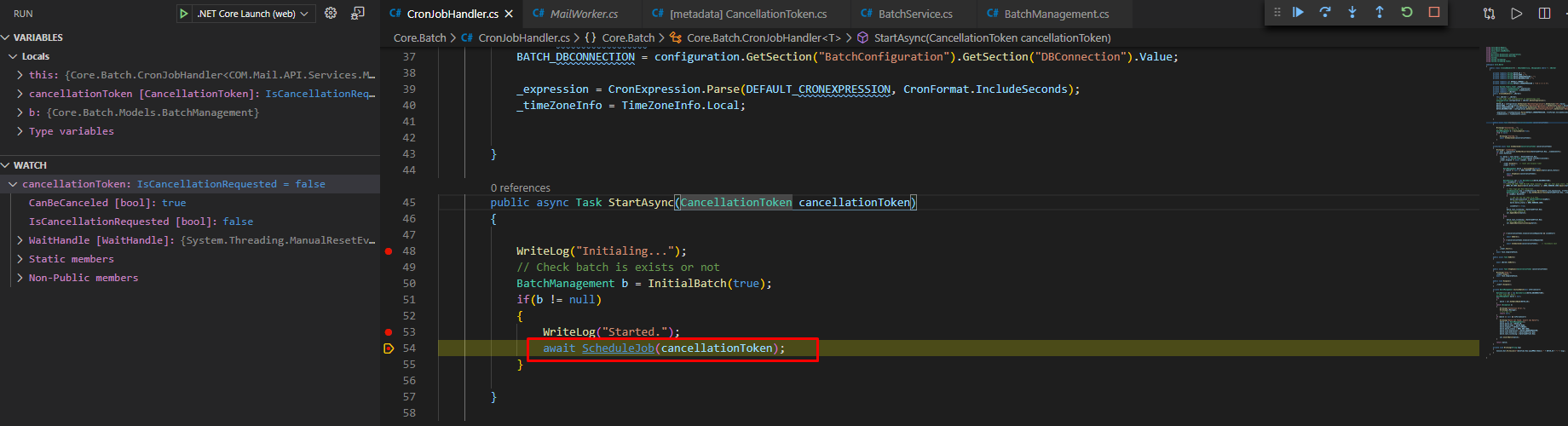


Tạo object BathService để thao tác batch với server -> các thao tác db thực hiện với 1 đối tượng thì được implement chung trong 1 file service, các action với db này là cơ bản, không chứa nghiệp vụ

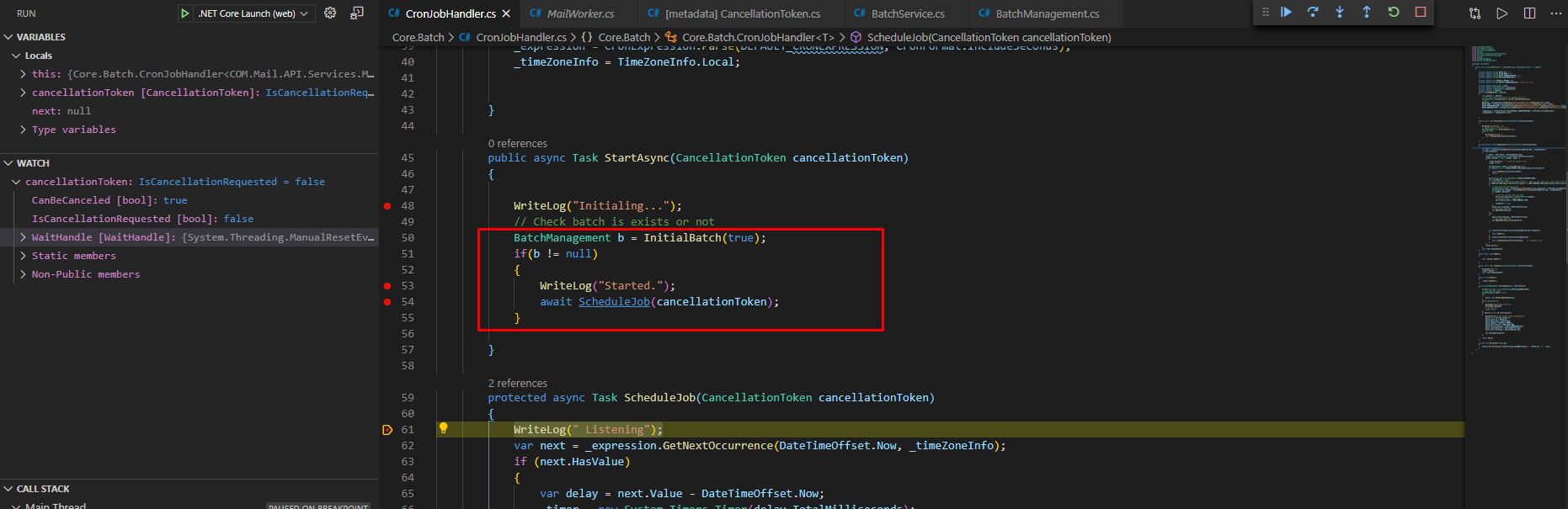
Try catch chổ getbatch để test connection

Khi batch lấy ra null và cho phép tạo batch mới thì insert new batch

Trả về batch đó



Biến cancellationToken.js được tạo từ class CancellationToken



Nếu b!= null thì thực hiện job