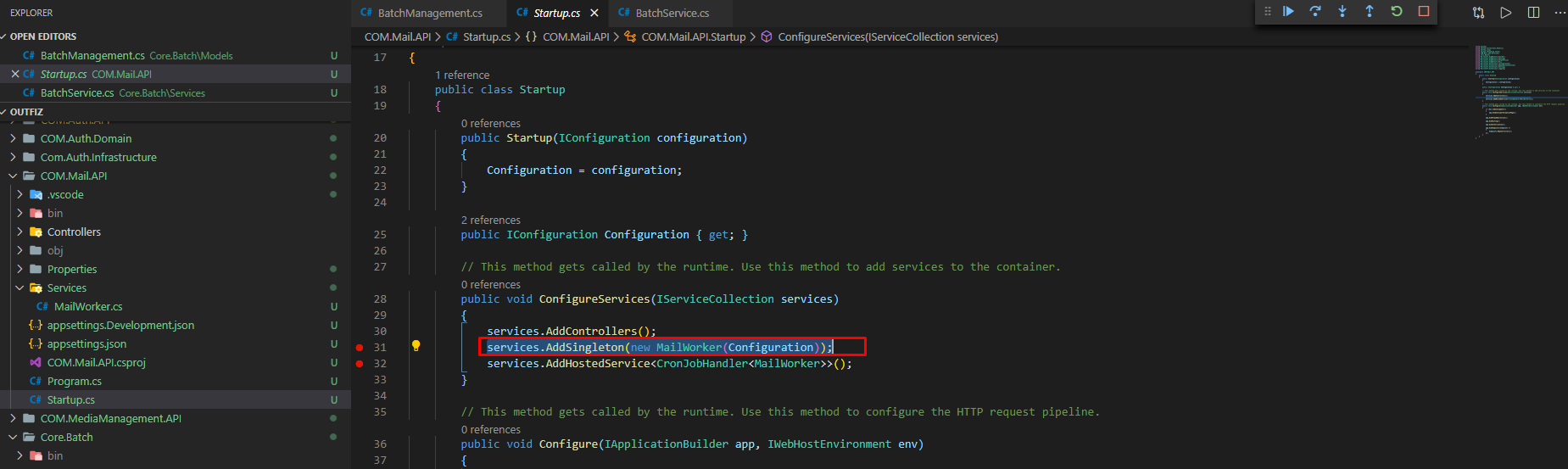


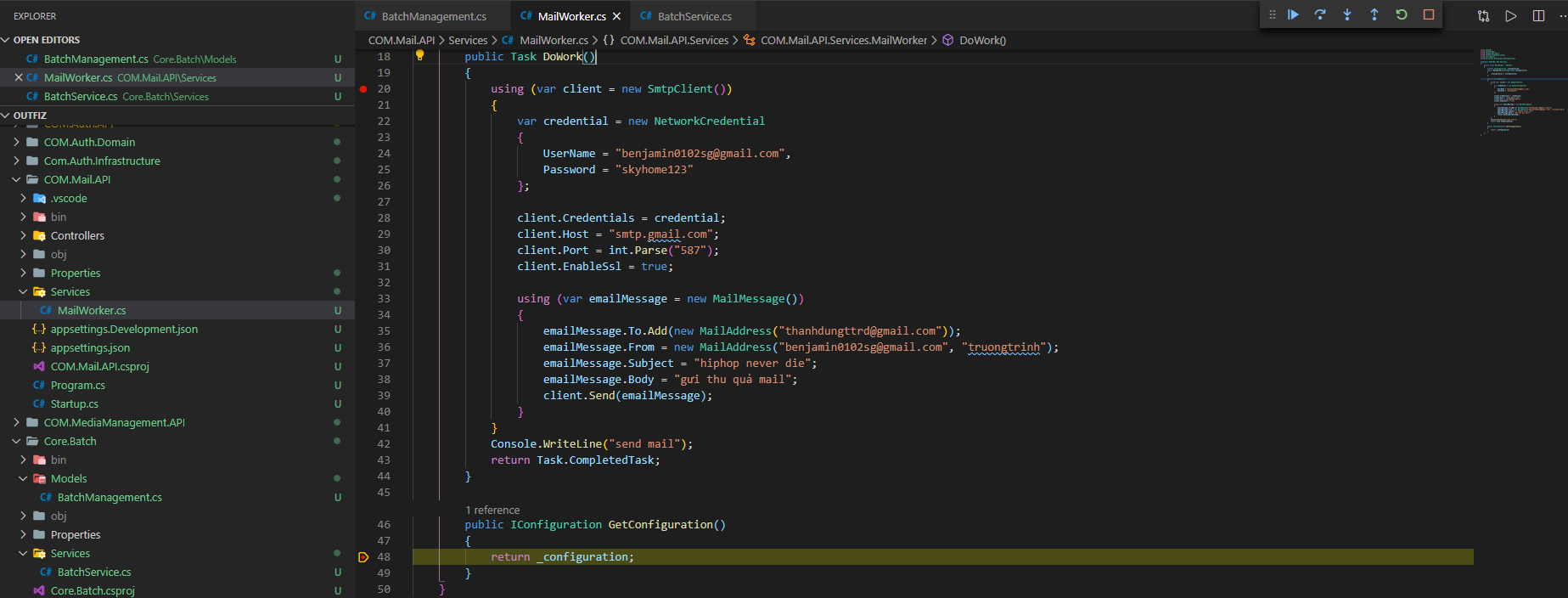
Class MailWorker đưa Iconfiguration vào bằng contructor ( đây không phải là tiêm, bì chưa register Iconfiguration tại startup)



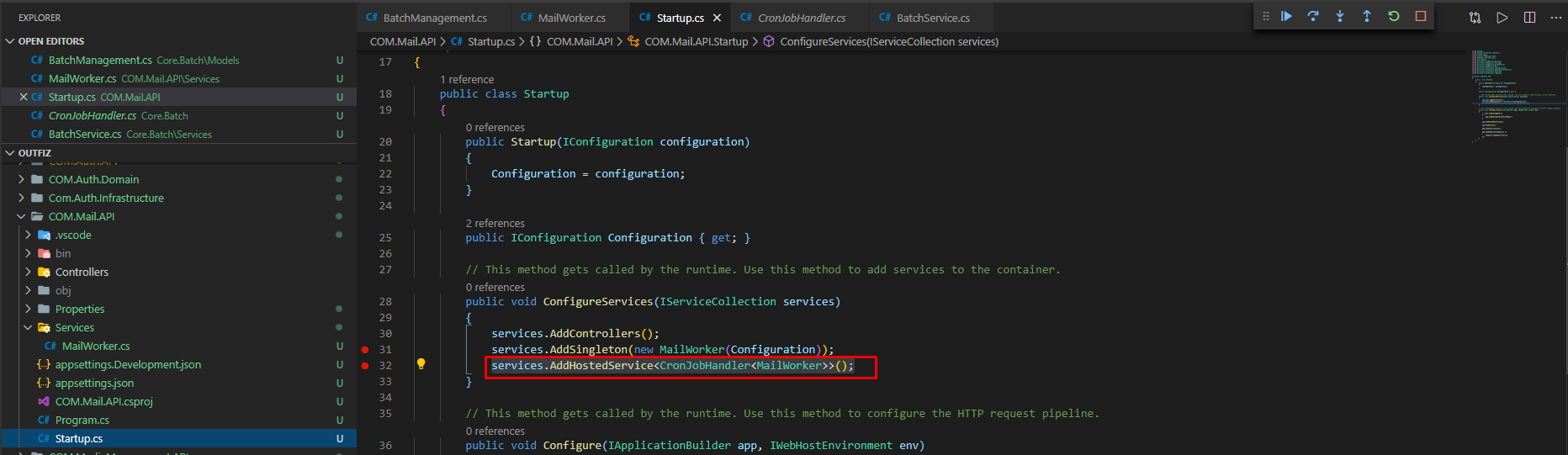
AddSingleton object MailWoker và đưa Configuration vào bằng contructor

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

//

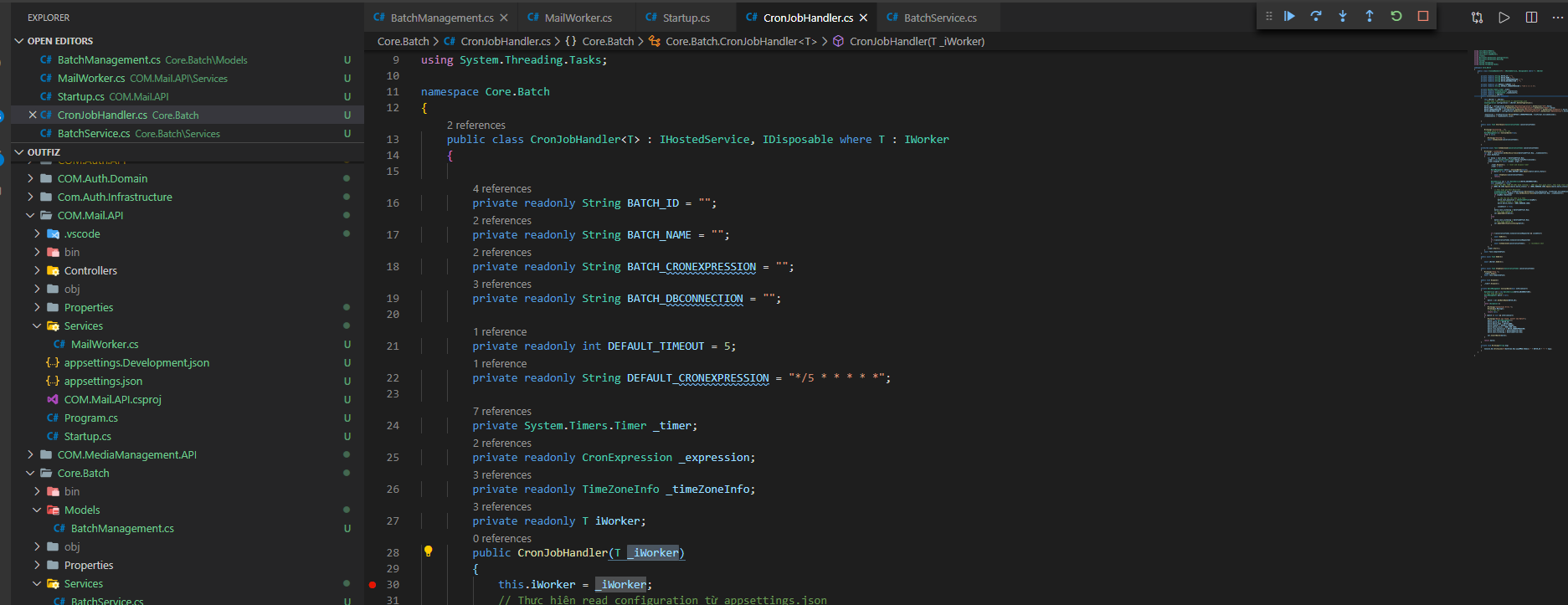


Trên MailWorker có class GetConfiguration từ class MailWorker có thể lấy configuration = Mailworker.Getconfiguration()



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

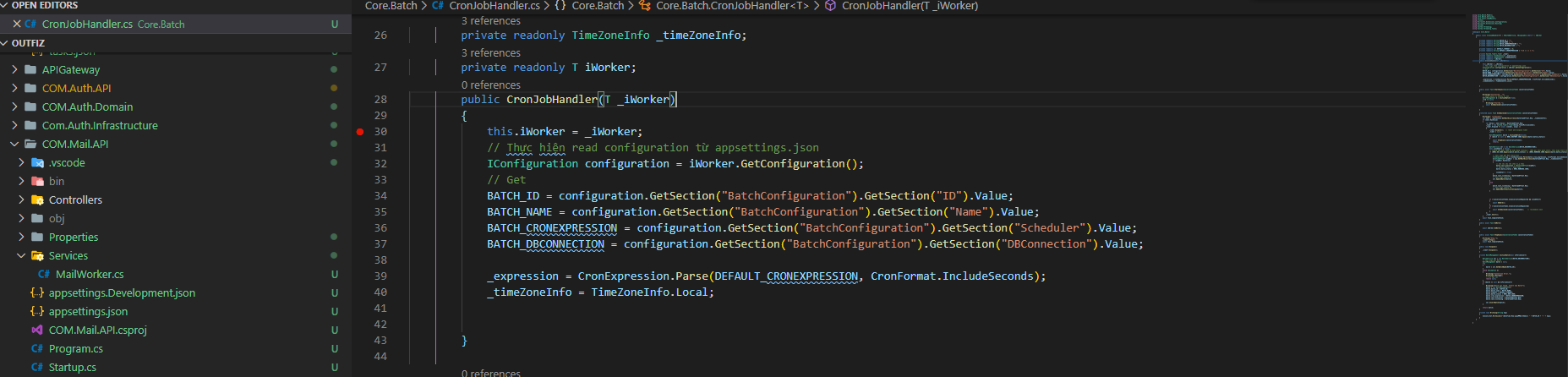
DEFAULT\_CRONEXPRESSION: giá trị mặc định mô tả thời điểm chạy batch

\_timer:

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

\_timeZoneInfo: lấy giá trị timezone

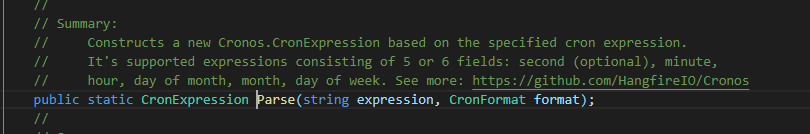
iWorker: generic thừa kế từ IWoker

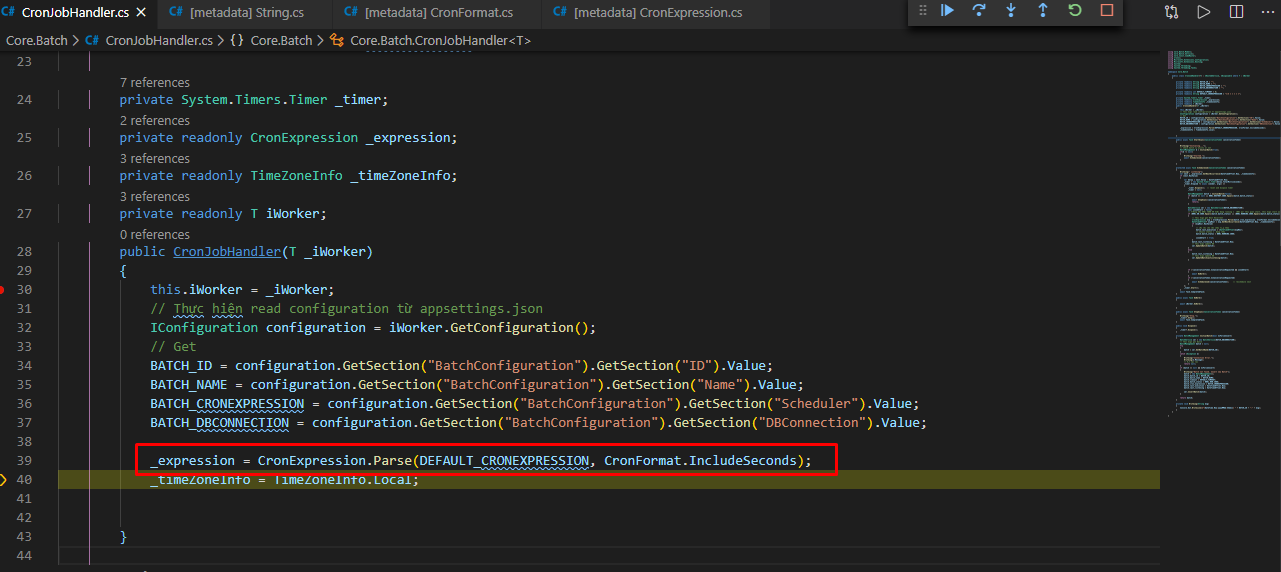


Khi services.AddHostedService<CronJobHandler<MailWorker>>(); thì contructor của CronJobHandler trên classlib core.bath được khởi chạy, với generic truyền vào

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

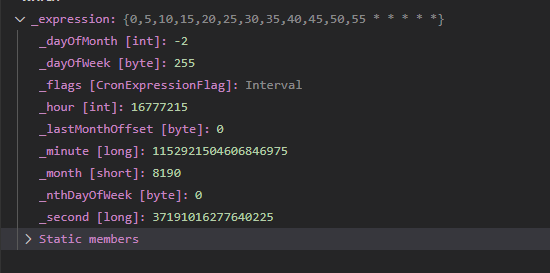
Khi đó dùng GetSection có thể get được các section trong file appsetting.json trên APIApp dung classlib này

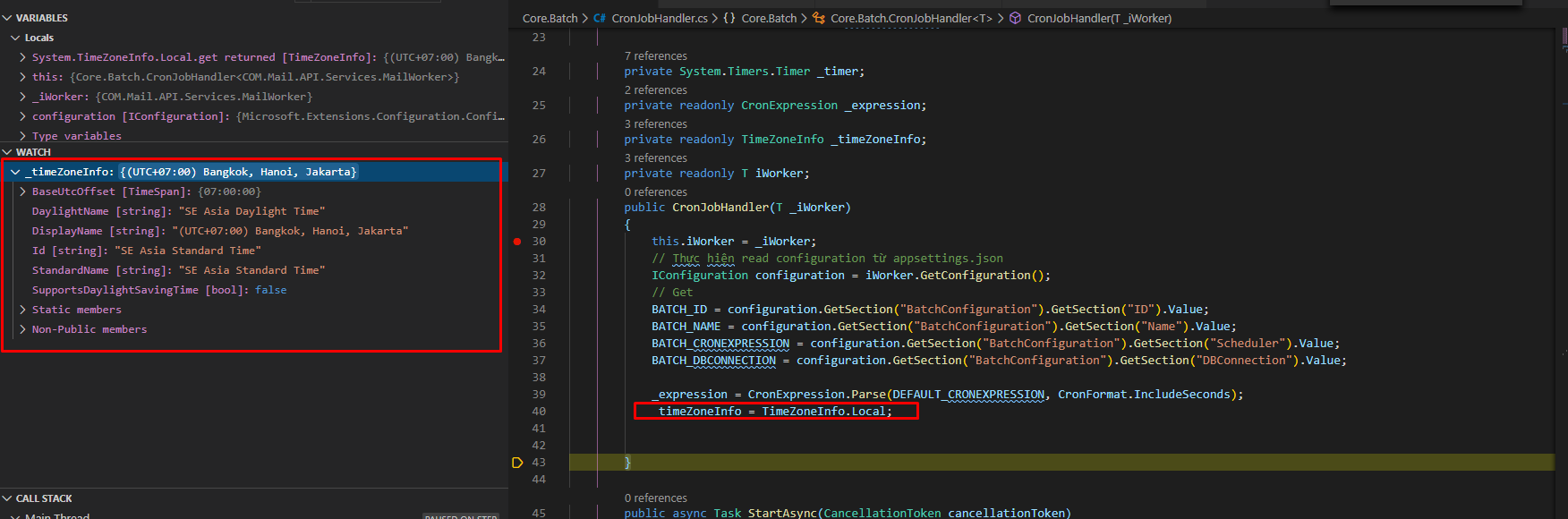




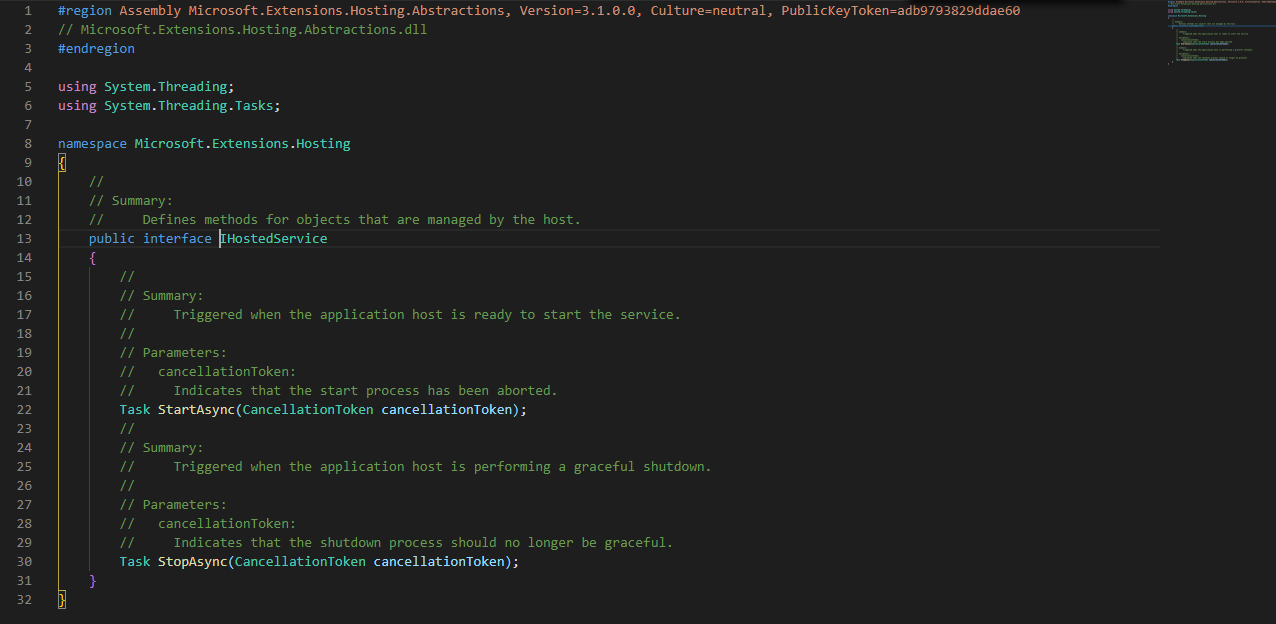
Format ở đây đưa thêm second vào

\_expression

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



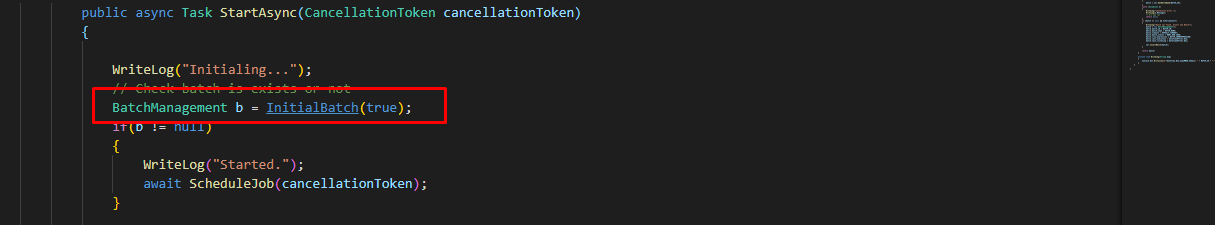
timeZoneInfo có kiểu timeZoneInfo chứa các giá trị của timezone



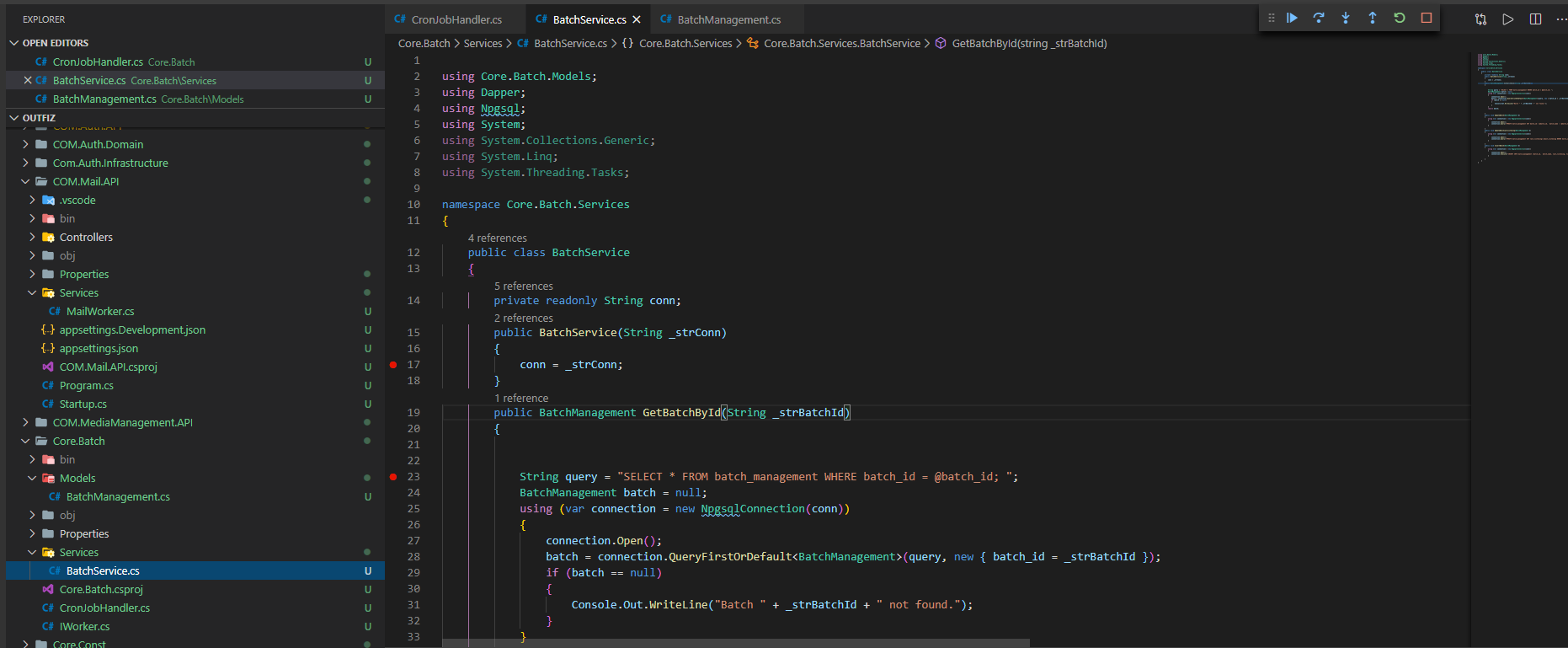
Trong IhostedService

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

StopAsync chạy khi app chạy xong và shutdown

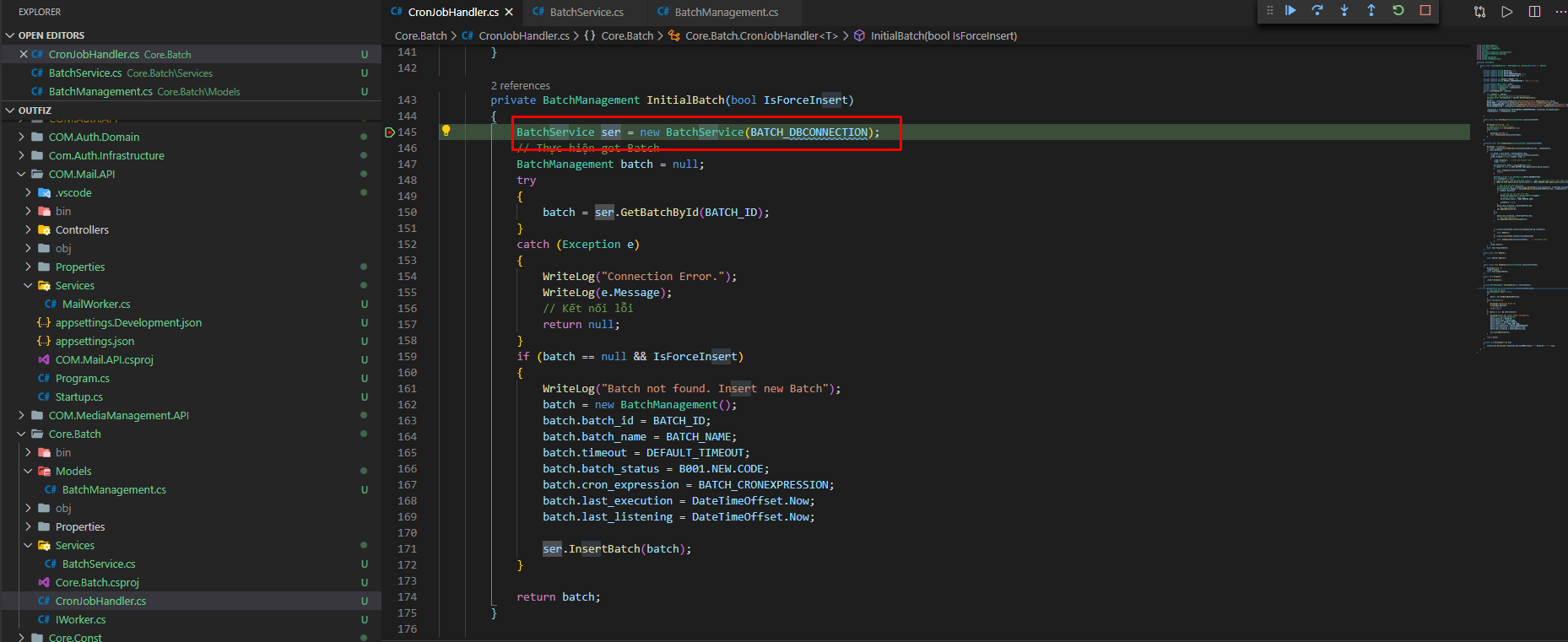


Gọi hàm InitialBatch(true)



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)

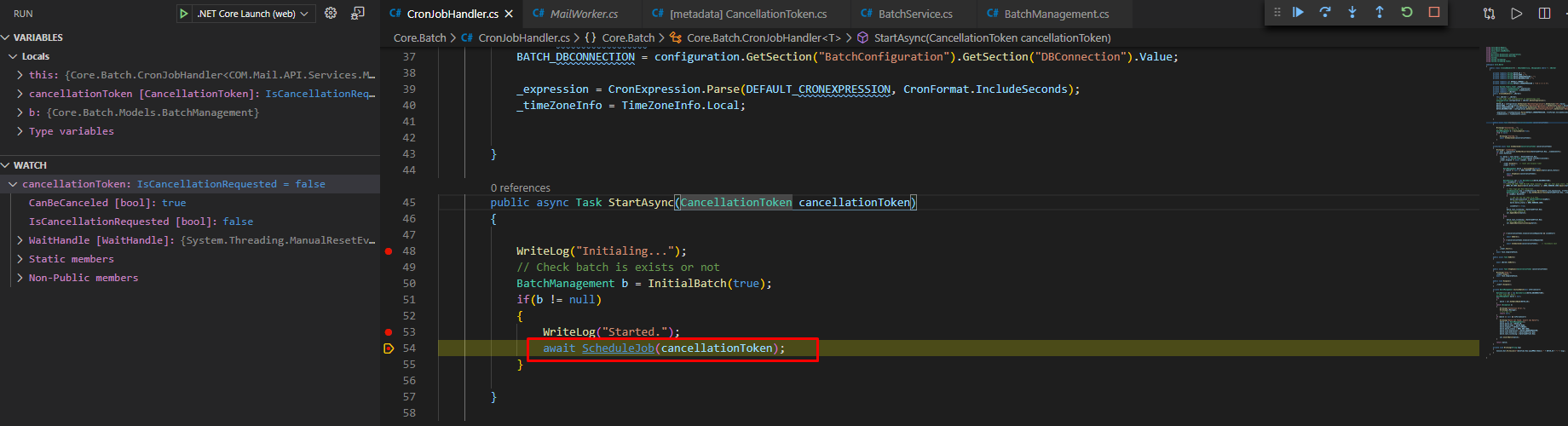


Tạo object BathService để thao tác batch với server -> các thao tác db thực hiện với 1 đối tượng thì được implement chung trong 1 file service, các action với db này là cơ bản, không chứa nghiệp vụ

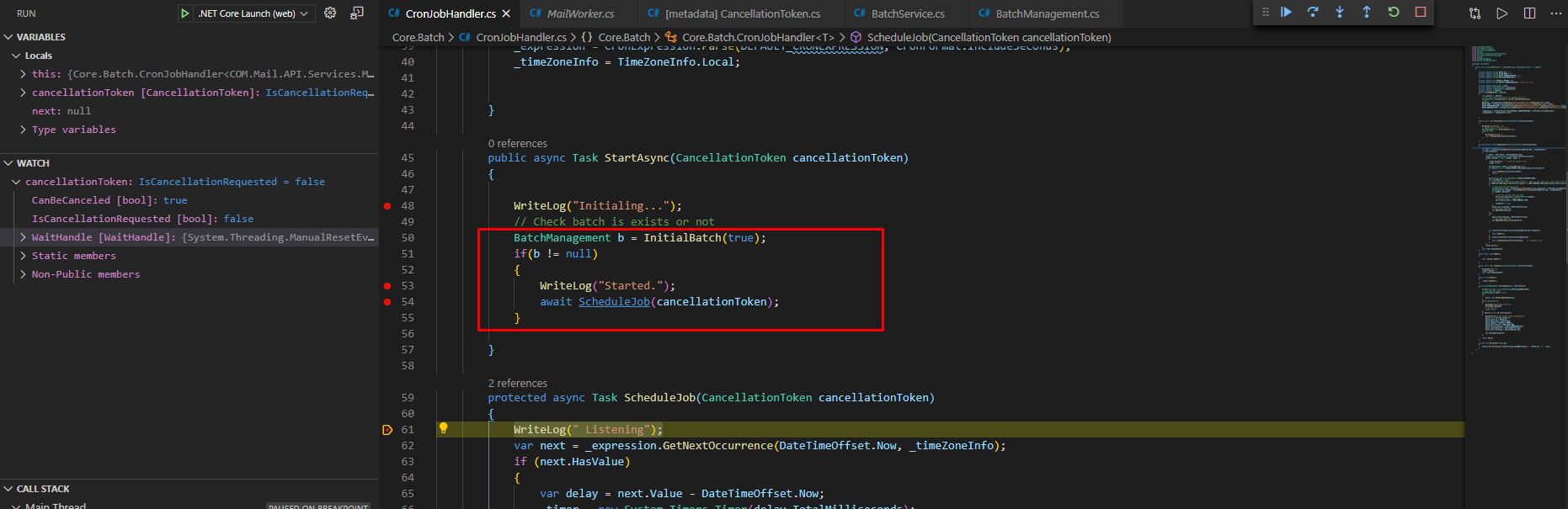
Try catch chổ getbatch để test connection

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

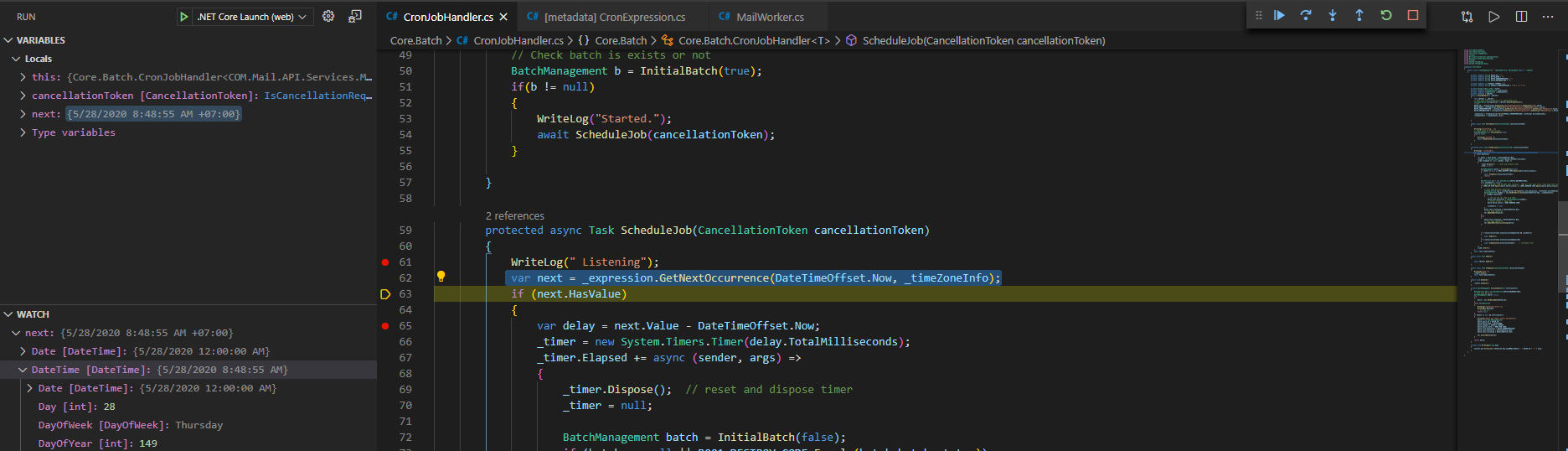
Trả về batch đó



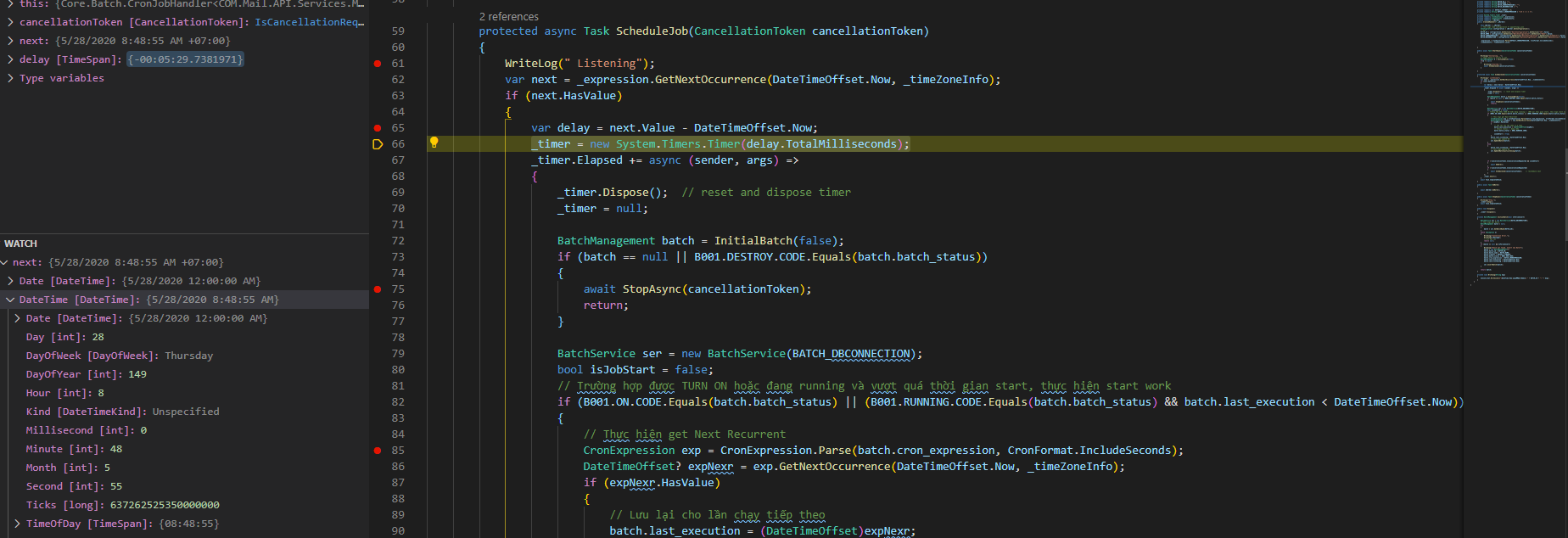
Biến cancellationToken.js được tạo từ class CancellationToken



Nếu b!= null thì thực hiện job

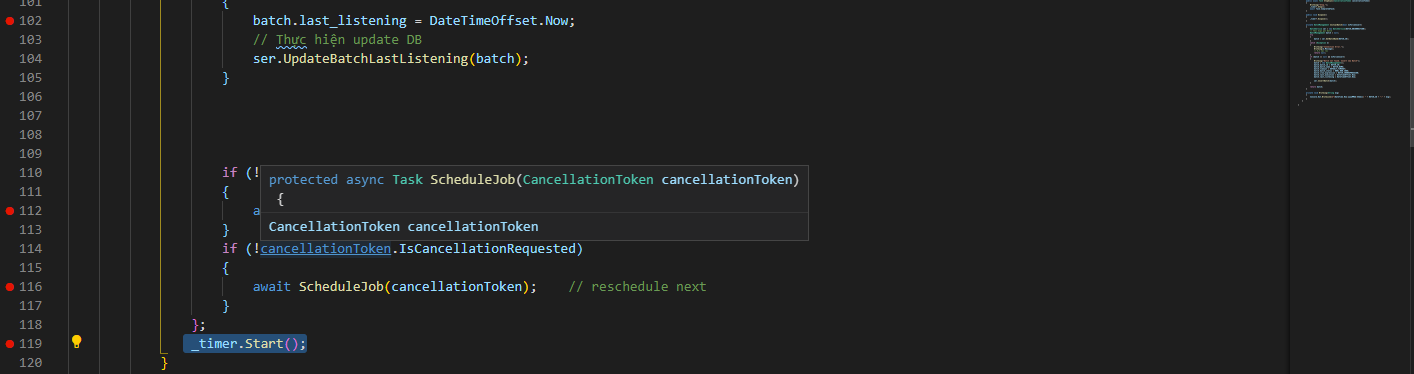


Next lưu lần thực hiện kế tiếp



Lấy giá trị next.value trừ đi thời gian hiện tại để ra khoảng delay giữa các lần chạy, delay sẽ được đưa vào inteval của timer

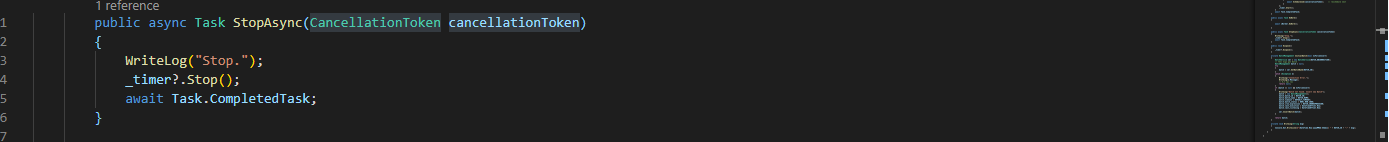
\_timer.Elapsed chạy hàm ở dưới khi khoảng thời gian interval trôi qua, \_timer phải start thì mới chạy hàm này



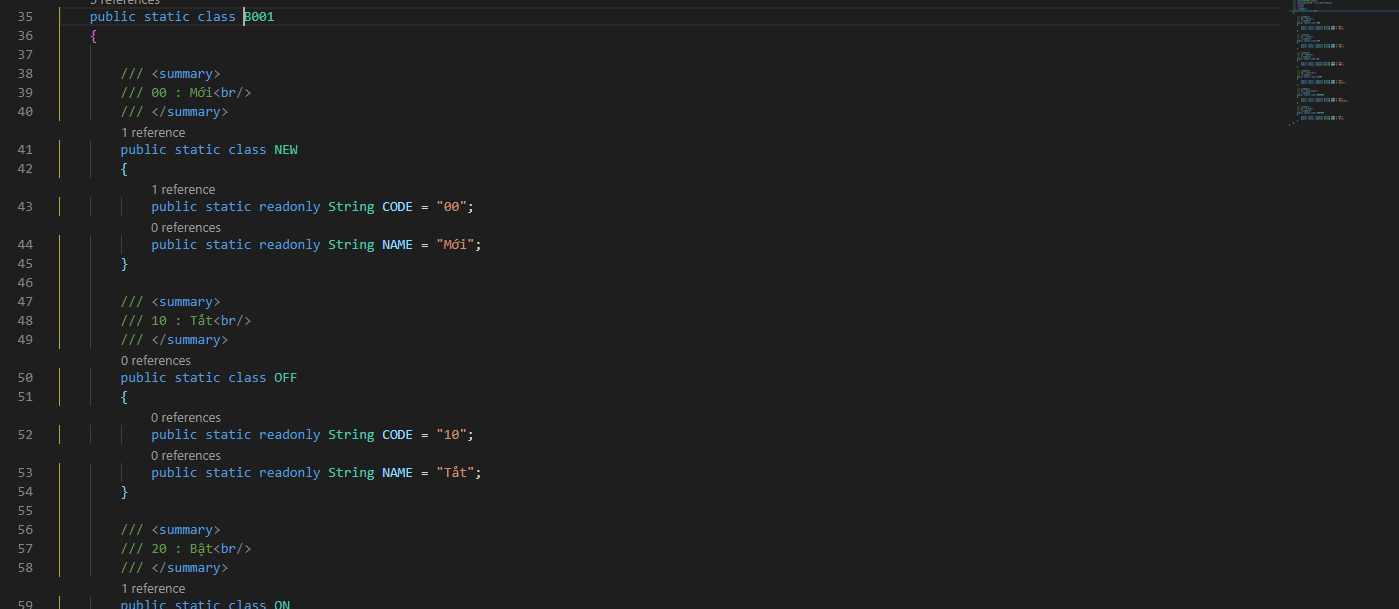
Hàm Elapsed được chạy sau hàm start

Khi vừa chạy thì reset đưa \_timer về null

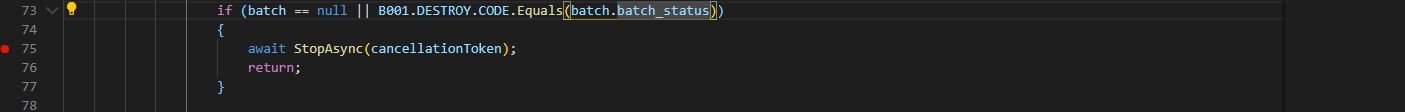
Thực hiện chạy lại initialBatch để lấy batch của iworker hiện tại, với thông số truyền vào false, nghĩa là nếu batch null thì ko cho tạo mới batch, vì lần initial đầu đã tạo rồi



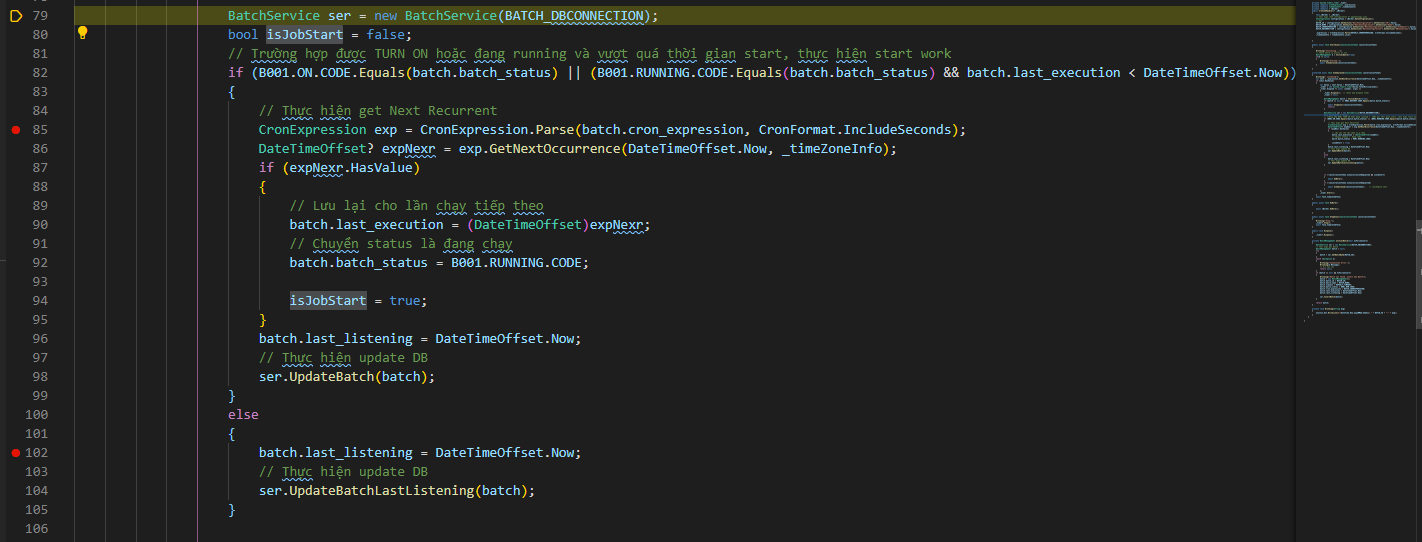
Nên nếu batch null thì gọi stopAsnc với cancellationToken luôn, hàm stop sẽ dừng time lại



B001 là class chứa các constant



Nếu giá trị của batch\_status = destroy thì nghĩa là hủy batch này thì cũng chạy hàm stopAsync



Trường hợp batch không null thực hiện lấy batch service

Đặt cờ isJobStart để check job đã start hay chưa

Trường hợp batch ở trạng thái on hoặc đang chạy thì lấy next recurrent bằng cách

Khai báo lại biến exp

Dùng biến exp để lấy next occurnece

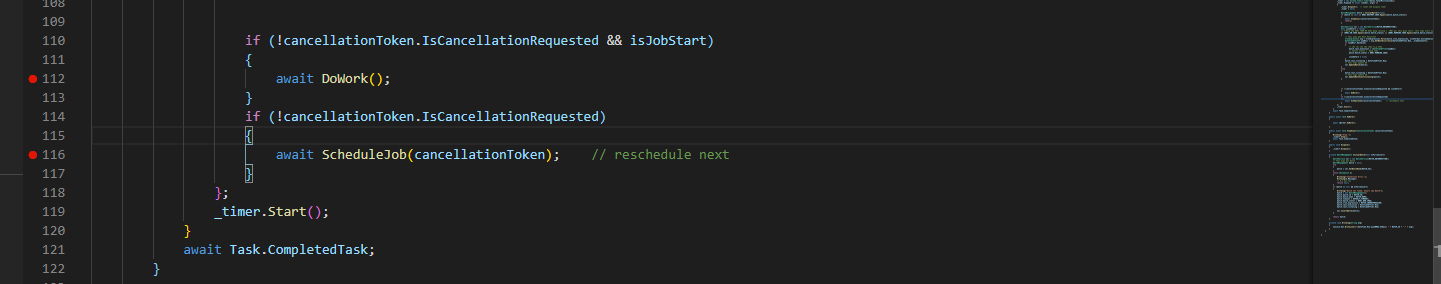
Nếu có giá trị của lần chạy tiếp theo thì lưu lại last\_execution vào db để so sảnh ở lần chạy tiếp theo

Chuyển status của batch sang running

Và đưa cờ isJobStart về true

Đưa batch last\_listening gắn cho thời gian hiện tại và update batch vào db

Các trường hợp batch\_status khác thì batch ko chạy, thực hiện update batch.last\_listening, nghĩa là các trường hợp batch chưa chạy chỉ listening thì cứ qua 1 chu kỳ listening thì update last\_listening vào db



Trường hợp có không có cancel request và biến isJobstart thì gọi hàm Dowork,



Hàm này gọi hàm DoWork được implement ở apiApp

Sau khi do work xong Trường hợp không có cancel nhưng job chưa start thì gọi lại hàm schedule job

cancellationToken: theo dõi và check nếu user request cancel như nhấn f5, ctrl-c