Querying the state machine via a controller action

Script start

Before we start adding more meaningful logic to our state machine, let's add a new controller action to our Trading microservice so that we can easily query the current state of our state machine instances.

In Trading repo

1. Update Contracts.cs:

```
namespace Play.Trading.Service.Contracts
{
    public record PurchaseRequested(Guid UserId, Guid ItemId, int Quantity, Guid CorrelationId);
    public record GetPurchaseState(Guid CorrelationId);
}
```

2. Update PurchaseStateMachine.cs:

```
public class PurchaseStateMachine : MassTransitStateMachine<PurchaseState>
{
    …
    public Event<PurchaseRequested> PurchaseRequested { get; }
    public Event<GetPurchaseState> GetPurchaseState { get; }

    public PurchaseStateMachine()
    {
        …
        ConfigureInitialState();
        ConfigureAny();
    }

    private void ConfigureEvents()
    {
        Event(() => PurchaseRequested);
        Event(() => GetPurchaseState);
    }

    private void ConfigureInitialState()
    {
      …
    }

    private void ConfigureAny()
```

```
    {
        DuringAny(
            When(GetPurchaseState)
                .Respond(x => x.Instance)
        );
    }
}
```

3. Update Dtos.cs:

```
namespace Play.Trading.Service.Dtos
{
    public record SubmitPurchaseDto([Required] Guid ItemId, [Range(1, 100)] int Quantity);

    public record PurchaseDto(
        Guid UserId,
        Guid ItemId,
        decimal? PurchaseTotal,
        int Quantity,
        string State,
        string Reason,
        DateTimeOffset Received,
        DateTimeOffset LastUpdated);
}
```

4. Update PurchaseController.cs:

```
public class PurchaseController : ControllerBase
{
    readonly IPublishEndpoint publishEndpoint;
    private readonly IRequestClient<GetPurchaseState> purchaseClient;

    public PurchaseController(
        IPublishEndpoint publishEndpoint,
        IRequestClient<GetPurchaseState> purchaseClient)
    {
        this.publishEndpoint = publishEndpoint;
        this.purchaseClient = purchaseClient;
    }

    [HttpGet("status/{correlationId}")]
    public async Task<ActionResult<PurchaseDto>> GetStatusAsync(Guid correlationId)
    {
        var response = await purchaseClient.GetResponse<PurchaseState>(
```

```
            new GetPurchaseState(correlationId));

        var purchaseState = response.Message;

        var purchase = new PurchaseDto(
            purchaseState.UserId,
            purchaseState.ItemId,
            purchaseState.PurchaseTotal,
            purchaseState.Quantity,
            purchaseState.CurrentState,
            purchaseState.ErrorMessage,
            purchaseState.Received,
            purchaseState.LastUpdated);

        return Ok(purchase);
    }

    [HttpPost]
    public async Task<IActionResult> PostAsync(SubmitPurchaseDto purchase)
    {
        var userId = User.FindFirstValue("sub");
        var correlationId = NewId.NextGuid();

        var message = new PurchaseRequested(
            Guid.Parse(userId),
            purchase.ItemId,
            purchase.Quantity,
            correlationId);

        await publishEndpoint.Publish(message);

        return AcceptedAtAction(nameof(GetStatusAsync), new { correlationId }, new { correlationId });
    }
}
```

5. Update Startup.cs:

```
public class Startup
{
    …
    // This method gets called by the runtime. Use this method to add services to the container.
    public void ConfigureServices(IServiceCollection services)
    {
        …
```

```
            services.AddControllers(options =>
            {
                options.SuppressAsyncSuffixInActionNames = false;
            })
            .AddJsonOptions(options => options.JsonSerializerOptions.IgnoreNullValues = true);
            …
        }

        …

        private void AddMassTransit(IServiceCollection services)
        {
            …
            services.AddMassTransitHostedService();
            services.AddGenericRequestClient();
        }
    }
```

6. Start Identity and Trading services

In Postman

7. Start a purchase

8. Copy the location header from Postman console

9. Do a GET for the state machine:

GET https://localhost:5007/purchase/status/{correlationId}

In the next lesson we will update our Trading service so it can consume the prices of the Catalog items.