Adding idempotency to a controller action

Script start

Let’s see what’s the effect of not having an idempotent Purchase operation in our Trading microservice.

In Trading repo

1. Add a delay to PurchaseController.PostAsync:

```
public async Task<IActionResult> PostAsync(SubmitPurchaseDto purchase)
{
   …
   await publishEndpoint.Publish(message);

   await Task.Delay(TimeSpan.FromSeconds(5));

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

2. Start Inventory, Identity and Trading services

In Postman

3. Open Settings and set a Request timeout of 3000 ms

4. Start the purchase

5. Notice it times out

6. Notice the gil was debited and the items granted

7. Do the purchase again

8. Notice gil was debited and the items granted again

In Trading repo

9. Update Dtos.cs:

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

        …
}
```

10. Update PurchaseController.cs:

```
public class PurchaseController : ControllerBase
{
    …

    [HttpGet("status/{idempotencyId}")]
    public async Task<ActionResult<PurchaseDto>> GetStatusAsync(Guid idempotencyId)
    {
        var response = await purchaseClient.GetResponse<PurchaseState>(
            new GetPurchaseState(idempotencyId));
        …
    }

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

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

        await publishEndpoint.Publish(message);

        await Task.Delay(TimeSpan.FromSeconds(5));

        return AcceptedAtAction(nameof(GetStatusAsync), new { purchase.IdempotencyId }, new { purchase.IdempotencyId });
    }
}
```

11. Start Trading service

In Postman

12. Perform a purchase with IdempotencyId

    ```
    {
      "ItemId": "[some id]",
      "Quantity": 1,
      "IdempotencyId": "{{$guid}}"
    }
    ```

13. Get the current gil of the user

14. Grab the IdempotencyId from the Postman console

15. Perform the purchase again with same idempotency id

16. Get the current gil of the user. Notice it did not change.

17. Notice the VS Code console error in the SAGA

18. Remove the delay

In the next lesson we will update our state machine to properly handle the idempotency id