Sending and consuming Inventory messages from the state machine

Script start

The main purpose of our purchase state machine is to grant items to a user and debit from him the required amount of gil, which it can only do by interacting with the Inventory and Identity microservices.

Let's now address the first part of this equation by updating the state machine so it can send the GrantItems command to Inventory.

In Trading repo

1. Add the Inventory contracts NuGet package:

```
dotnet add package Play.Inventory.Contracts
```

2. Update PurchaseStateMachine.cs:

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

   public PurchaseStateMachine()
   {
      …
      ConfigureGetPurchaseState();
      ConfigureAccepted();
   }

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

   private void ConfigureInitialState()
   {
      Initially(
         When(PurchaseRequested)
            …
            .Activity(x => x.OfType<CalculatePurchaseTotalActivity>())
           .Send(context => new GrantItems(
                        context.Instance.UserId,
                        context.Instance.ItemId,
                        context.Instance.Quantity,
                        context.Instance.CorrelationId))
                .TransitionTo(Accepted)
        );
    }

    private void ConfigureAccepted()
    {
        During(Accepted,
            When(InventoryItemsGranted)
                .Then(context =>
                {
                    context.Instance.LastUpdated = DateTimeOffset.UtcNow;
                })
                .TransitionTo(ItemsGranted)
        );
    }
    ...
}
```

3. Update appsettings.json:

```
{
   ...
  "RabbitMQSettings": {
    "Host": "localhost"
  },
  "QueueSettings":
  {
    "GrantItemsQueueAddress": "queue:inventory-grant-items"
  },
  "AllowedHosts": "*"
}
```

4. Add the Settings directory

5. Add QueueSettings.cs:

```
namespace Play.Trading.Service.Settings
{
    public class QueueSettings
    {
        public string GrantItemsQueueAddress { get; init; }
    }
}
```

6. Update Startup:

```
private void AddMassTransit(IServiceCollection services)
{
    services.AddMassTransit(configure =>
    {
      ...
    });

    var queueSettings = Configuration.GetSection(nameof(QueueSettings)).Get<QueueSettings>();
    EndpointConvention.Map<GrantItems>(new Uri(queueSettings.GrantItemsQueueAddress));

    services.AddMassTransitHostedService();
    ...
}
```

7. Start Identity, Inventory and Trading services

In Postman

8. Verify the items in the user inventory (ITEMS API in Inventory)

9. Perform a purchase

10. Verify the state machine state

11. Notice it’s in Accepted state, not ItemsGranted

In the next lesson we will understand why the state machine did not arrive to the ItemsGranted state and what to do about this unexpected scenario.