

## Sending and consuming Identity messages from the state machine

### Script start

Let's add the final step to our state machine, where it reaches out to the Identity microservice to debit gil from the user and complete the purchase.

### In Trading repo

1. Add the Identity contracts NuGet package:

```
dotnet add package Play.Identity.Contracts
```

2. Update PurchaseStateMachine:

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

    public PurchaseStateMachine()
    {
        ...
        ConfigureAccepted();
        ConfigureItemsGranted();
    }

    private void ConfigureEvents()
    {
        ...
        Event(() => InventoryItemsGranted);
        Event(() => GilDebited);
    }

    ...

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

```

    })
    .Send(context => new DebitGil(
        context.Instance.UserId,
        context.Instance.PurchaseTotal.Value,
        context.Instance.CorrelationId))
    .TransitionTo(ItemsGranted)
    );
}

private void ConfigureItemsGranted()
{
    During(ItemsGranted,
        When(GilDebited)
        .Then(context =>
            {
                context.Instance.LastUpdated = DateTimeOffset.UtcNow;
            })
        .TransitionTo(Completed)
    );
}
...
}

```

3. Update appsettings.json:

```

"QueueSettings": {
  "GrantItemsQueueAddress": "queue:inventory-grant-items",
  "DebitGilQueueAddress": "queue:identity-debit-gil"
},

```

4. Update QueueSettings.cs:

```

public class QueueSettings
{
    public string GrantItemsQueueAddress { get; init; }
    public string DebitGilQueueAddress { get; init; }
}

```

5. Update Startup.cs:

```

private void AddMassTransit(IServiceCollection services)
{
    ...
    EndpointConvention.Map<GrantItems>(new Uri(queueSettings.GrantItemsQueueAddress));
}

```

```
EndpointConvention.Map<DebitGil>(new Uri(queueSettings.DebitGilQueueAddress));  
  
services.AddMassTransitHostedService();  
...  
}
```

6. Start Inventory, Identity and Trading services

In Postman

7. Get the gil of the user about to purchase
8. Get the inventory of the user about to purchase
9. Do the purchase
10. Confirm inventory increased and gil decreased

In the next module we will refine our state machine to be more resilient to the different errors that could occur during its lifetime.