Purchase Saga Overview

In this lesson you will get an overview of the orchestrated saga that you will build to drive the purchase flow.

The purchase saga

Let's see which are the steps and the participants involved in our upcoming purchase saga.

It all starts when the client sends a request to the Trading microservice to purchase some items. Since this process involves a long running operation, and we don't want to keep the client waiting, Trading only publishes an event with the details of this purchase request and then it immediately replies to the client with an accepted response.

Now, since Trading is the coordinator in this saga, it includes a state machine component that will decide which are the next steps and will keep track of the current step of the overall process. This state machine is triggered when the purchase requested event arrives to the queue Trading it's listening to, and the first step it takes is to calculate the total amount of gil required to complete the purchase.

With that done, Trading sends a command to the Inventory microservice to grant the requested items to the specified user. Notice this is an asynchronous message, not a REST API call. After sending the message, the purchase state machine switches to the Accepted state.

Once Inventory is done granting the items to the user's inventory bag, it publishes an items granted event, which Trading is subscribed to. When Trading receives this message, it immediately follows by sending a new command to the Identity microservice, asking it to debit the calculated total amount of gil from the user. The state machine then switches to the Items Granted state.

Eventually, Identity publishes a gil debited event which, after receiving it, signals Trading that all the steps in the purchase operation are done and therefore it switches the state machine to the Completed state.

There are of course some error scenarios to account for here, but we will take a look at those in a future module.

Preparing the participants

Now, there are quite a few things to build to enable this purchase saga, but in this module we will focus on the commands and events that our existing microservices need to expose so that later on the purchase state machine can take advantage of them.

This involves the grant items command and the items granted event on the Inventory microservice, the debit gil command and the gil debited event in the Identity microservice, and also, we will make a small update to the events published by our Catalog microservice so that they also include the prices of any created or updated items.

Let's get started.