DEMO

Setup

- Starting Catalog items:

Name	Description	Price
Antidote	Cures poison	7
Potion	Restores a small amount of HP	5
Iron sword	A standard sword made of iron	33
Ether	Restores a small amount of MP	6

- Delete the inventoryItems collection from Inventory DB
- Ensure the 3 queues are created and empty
- Sign out of RabbitMQ portal
- Stop all web servers
- docker-compose down
- Prepare all Postman tabs
- Remove browser extensions

Start

Let's take a quick look at the microservices based application that you will build in this course.

1. Start with the VS Code instance that has D:\projects in a terminal

In this Visual Studio Code instance, I have opened a terminal into the directory where I have all the directories and files that make up the microservices application. Let me quickly describe what each directory is for.

2. Describe all the dirs

3. Switch to Play.Infra dir and show docker-compose file

4. docker-compose up -d

5. Switch to the frontend VS Code instance

6. Start the frontend and load it in the browser

7. Explain the home page

This is a very simple frontend portal that users can use to exercise the features of the microservices based application. As you can see at the top there is a section to browse the catalog and another one to explore the user's inventory bag.

8. Switch to Play.Catalog in VS Code

9. Switch to src\Play.Catalog.Service

10. dotnet run

11. Browse to Catalog section

12. Open the Add Item dialog

13. Show client-side validations (price or name empty)

14. Show server-side validations (negative price)

15. Create a new item:
Name: Starter shield
Description: A basic starter shield
Price: 25

16. Update an item:
Name: Potion => Mega-potion
Description: Restores a ~~small~~ large amount of HP
Price: 5 -> 18

17. Delete an item:
Ether

18. Go to Rabbit MQ from Home page

19. Start Inventory microservice

20. Open the grant item dialog

21. Copy the User Id

22. Grant items to user:
Antidote (3)
Starter shield (1)

23. Switch to Inventory section

24. Get the inventory for copied user id

25. Update an item:
Name: Starter shield => Bronze shield
Description: A basic shield made of bronze

Price: same

26. Stop Catalog

So how Inventory is able to present data owned by the Catalog microservice without ever talking to Catalog is one of the key things you will learn by the end of this course.

27. Restart Catalog

28. Go to Catalog's Swagger UI from Home page

29. Try out Catalog's GET api from Swagger UI

30. Show the Inventory Swagger UI

31. Open Postman

32. Try out GET /items

33. Show the DBs in the MONGODB VS code extension

I hope you are as excited as me to go through the journey that will allow us to build this nice collaboration between microservices.

In the next lesson I'll show you how you can get access to all the source code and the other resources included with this course.