Exploring the front-end to microservices communication

Let's now explore in more detail all the scenarios enabled in the front-end and how its different react components are interacting with the Catalog and Inventory microservices.

Script

1. Start Catalog, Inventory and Frontend

2. In the browser, go to the home page

3. Go to the Catalog section

4. Put breakpoints in populateItems() in Catalog.js

5. Put breakpoint in ItemsController GetAsync in Catalog service

6. Refresh the Catalog section in browser and walk through the breakpoints

7. Open ItemForm.js and put break points in createItem() method

8. Create a new item:
Name: Bronze shield
Description: A basic shield
Price: NULL

9. Show client side validations

10. Enter price -3 and Save

11. Enter price 6 and Save

12. Update the created item to:
Name: Bronze shield
Description: A basic shield
Price: 8

13. Delete some item

14. Open GrantItemForm.js

15. Add breakpoints to grantItem()

16. Add a breakpoint to ItemsController PostAsync in Inventory service

17. Grant the Bronze shield to some user

18. Open Inventory.js

19. Add a breakpoint to populateItems()

20. Get the inventory to the user used in 17.

21. Back to Catalog

22. Rename Bronze Shield to Basic Shield

23. Back to Inventory, query same user's inventory

24. Notice Bronze Shield shows a Basic shield now

That marks the end of this module. You have connected a modern single page application built with React and hosted in a Node.js server with your two .NET based microservices to light up a few basic end to end scenarios. And you learned all about CORS along the way.

I hope you enjoyed building this system with me so far. If you would like to know where to go from here, please check out the next and last lesson, which will provide you with a quick overview of the next course in the series.