Exploring the front-end client

(Demo prep)

- Starting Catalog items:

Name	Description	Price
Potion	Restores some HP	5
Antidote	Cures poison	7
Hi-Potion	Restores a medium amount of HP	9

- Start with Admin, Player1 and Player2
- Player1 and Player2 have 100 gil (Admin has 0)
- Player1 has 3 Potions in Inventory bag
- Disable Edge extensions and favorites
- Apply VS Code settings (%APPDATA%\Code\User\settings.json)
- Apply PowerShell environment settings

Exploring the front-end client

Let's now explore in more detail how the updated front-end portal is taking advantage of OpenID Connect ID Tokens to secure and dynamically render UI elements based on the claims of the signed in user.

In Play.Frontend

1. Start signed out in front-end
2. Notice most nav bar elements are not there
3. Sign in as Player1 in the home page
4. Notice the dynamic nav bar and dynamic elements in main body
5. Explain the use of the User object and ID Token in NavMenu.js
6. Stop the front-end server
7. Place a breakpoint in last line of AuthorizeService.js → getUser()
8. F5 front-end server/client
9. Decode the id_token to jwt.ms
10. Notice email and role are included in the claims

11. Notice email and role are also available in user.Profile object

12. Log out

13. Click on “Check your Inventory”

14. Notice the redirection to Login

15. Open App.js and explain AuthorizeRoute

16. Try browsing to http://localhost:3000/Users

17. Notice same behavior

18. Log in as Admin

19. Notice the updated nav bar

In the next lesson we will explore how the front-end portal is taking advantage of OAuth 2.0 access tokens to make authorized calls to the microservices that we secured in previous modules.