* What’s causing codes to never return on Windows.

IEntropyService is IEnumerator<byte> type, hence using count() will not to be able calculate the items in the Enumerator object as there is while loop which is in continuous loop. By using Any() it will return true if it find at least one element in sequence . Instead, we should be using Any()(InvitationCodeService), which solves our problem.

* What’s causing codes to only return sometimes on Linux.

While generating the code by reading the file based on path /dev/random, sometimes entropy level gets dropped down, at this point of time it won’t be able to generate the new code. This might be the reason of failing in generation of code sometimes.

* QA reports that they can sometimes get an invitation code without accepting the terms and conditions first.

This is happening because in current backend code we are not configuring the acceptedTerms based on the user who is accessing the end points. It has property \_acceptedTerms which is set on class level and it will be common for all the users. In order to justify the above statement I have added the example below.

For example : If there are 2 users A and B. Imagine if user A has hit the end point Accept terms, we set \_acceptedTerms = true. And now if the user B hit end point for getting invitation code it is generating the code as backend code is checking on same flag \_acceptedTerms == true and treating it as user has already accepted the terms.

Fix for it is instead of checking for same flag, there should be a flag which is configured for each and every user. In that way it can determine if the user has accepted the terms or not.

How to configure the database?

For maintaining the database, I am using Entity framework code first migration to apply the database schema and insert the testing data in to the tables (SQL Server).

In order to apply the db schema,

1. Firstly, need to update the connection string in appSettings.json to point to your local instance of Sql Server.
2. Run the command “Add-Migrations Initial\_Setup” in Nuget Package Manager Console keeping default project as “CohesionIB.ApiEngineer.CodeChallenge.Repository”.
3. Run Update-Database
4. After executing step 3, we will be seeing the Database created with 3 tables.
5. Dbo.Users 🡪 It contains details related to User
6. Dbo.Devices -> It contains device details and user configured to which device.
7. Dbo.UserInvitationCodes-> This table is responsible for capturing all the invitation codes generated by user and registers to device id .

Running the Web Api

1. Run the application keeping “CohesionIB.ApiEngineer.CodeChallenge” as startup project.
2. Using postman call the endpoints by passing in the username and password which are configured in dbo.Users table.

Endpoints :

1. [GET] <https://localhost:5001/api/InvitationCode/acceptTerms> ---> This will allow the user to accept the terms
2. [GET] <https://localhost:5001/api/InvitationCode/getinvitationcode> ---> Gives the invitation code.
3. [PUT] <https://localhost:5001/api/InvitationCode/registerDevice?code=6128077080957801768&deviceId=10000000003> 🡪 Registers the device with particular invitation code.
4. [GET] <https://localhost:5001/api/Device/getRegisteredCodes> --> Gets the registered device code for current user.