C#.NET Mock ASSESSMENT 7

CHALLENGE A

OVERVIEW

Create a .Net Core Web API to work as a Videogame player app.

SETUP

Use the starter application provided in the assessment repository.

BUILD SPECIFICATIONS

The assessment is worth ten points, one for each of the test cases below. **Pay special attention to the spelling and capitalization of the items in bold**.

HINT: For making your own APIs, Postman will be very helpful for testing your APP.

Player model with the following properties:

1. Int **Id**
2. String **UserName**
3. Int **Level**
4. PlayerClass **CurrentClass**

PlayerClass model with the following properties:

1. Int **Id**
2. String **Name**
3. String **Type**

GameDB:

For the sake of having data, the following lists have been supplied for you in the **GameDB** class so as to mimic a database. The values have already been input into the class so you can focus on the API end of things.

For sample Players we the following values:

ID	**UserName**	**Level**	**CurrentClass**
0	GrantChirpus	100	1

1	Gamer	50	0
2	Green-Bean-Gaming	75	2
3	Jeffery	80	0
4	FunnyFrog2	90	3

For sample PlayerClasses, we have the following playerclasses:

ID	**Name**	**Type**
0	Archer	Damage
1	Healer	Support
2	Knight	Tank
3	Wizard	Damage
4	Thief	Damage

GamingController

In your API create the following endpoints:

1. **Players** - an endpoint that returns all players. - 1pt
    a. Return back all Players as a list
2. **Classes** - an endpoint that returns all PlayerClasses. - 1pt
    a. Return back all PlayerClasses as a list
3. **PlayersMinLevel** - this endpoint takes in a parameter of int level and returns back all players above or equal to that level - 1pt
4. **PlayersSortLevel** - This endpoint returns all of the players sorted by their levels. (highest level first) - 1 pt
5. **PlayersOfClass**- this endpoint will take in a string and search for all Players with that PlayerClass name - 2 pt (1 for working endpoint and return. 1 for correct info being return)
6. **PlayersOfType** - this endpoint will take in a string and search for all Players with that PlayerClass type - 2 pt (1 for working endpoint and return. 1 for correct info being return)
7. **AllPlayedClasses** - this endpoint will filter through the Players and return all CurrentClasses (PlayerClass) being used. The returned list should not return duplicates

(this should not return Thief) - 2 pt (1 for working endpoint and return. 1 for correct info being returned)

SUBMISSION

When finished, push your changes to the same GitHub repository you cloned your project from.