Documentation

Open Quiz

cemuka

2019

Introduction

First of all, I've started this project to learn and implement a public API in Unity. But when I found that there is literally no implementation of Open Trivia DB I couldn't believe it. In Unity AssetStore there is bunch of quiz-trivia clones and packets were already there but all of them relied on local or offline workflow. So my intention on building this was only bare bones of a how to use a public api in unity and making a working system out of it. Since OpenTDB API is not a big complex service to work with and it is free to use, I've decided to build a trivia game with using this API.

This project intended to using a public API service inside Unity with C#. I don't recommend or believe to use this on production or release. Consider it as a template or learning resources.

Unity HTTP Methods

In Open Quiz, whole data flow relied on Unity's `UnityEngine.Networking` namespace and `GET` http connection. So `RestClientAPI` class responsible for getting data from the OpenTDB, parsing JSON data and making them new classes based on our contacts.

Basicly, `Get()` method in `RestClientAPI` class does the heavy work. Recomended way to to that using a method that returning an `IEnumerator`, like an async operation getting the data from a service.

I've only used Unity's `JSONUtility` class on purpose. I know that this is not the best solution on this topic, there is limitations but third party parsers may or may not crash your unity game. So I needed to write a helper class and methods to solve this limits.

If you want to dive deep on JSONUtility class I recommend this link to take a look at.

https://stackoverflow.com/questions/36239705/serialize-and-deserialize-json-and-json-array-in-unity

Understanding the JSON Parsing in Unity

We have talked about some limitations in unity's json utils. I've created a JsonHelper class to walk around that. Json util class unfortunately doesn't support all sort of schemes on JSON and I highly recommend that how and why you should make contract classes. For more information please check links below.

useful json schemes and using in unity

https://stackoverflow.com/questions/36239705/serialize-and-deserialize-json-and-json-array-in-unity

https://docs.unity3d.com/Manual/JSONSerialization.html

Contract Classes

OpenTDB responds with a response code and result object. With the JsonHelper class, I fixed that json string with wrappers and removed chars from it to match the contract fields with the exact same public variables(no getter and setter attributes).

https://opentdb.com/api_category.php

When you ask for categories, api responds with a trivia_categories object. So we must correct that scheme to match our contract like below.

```
[Serializable]
```

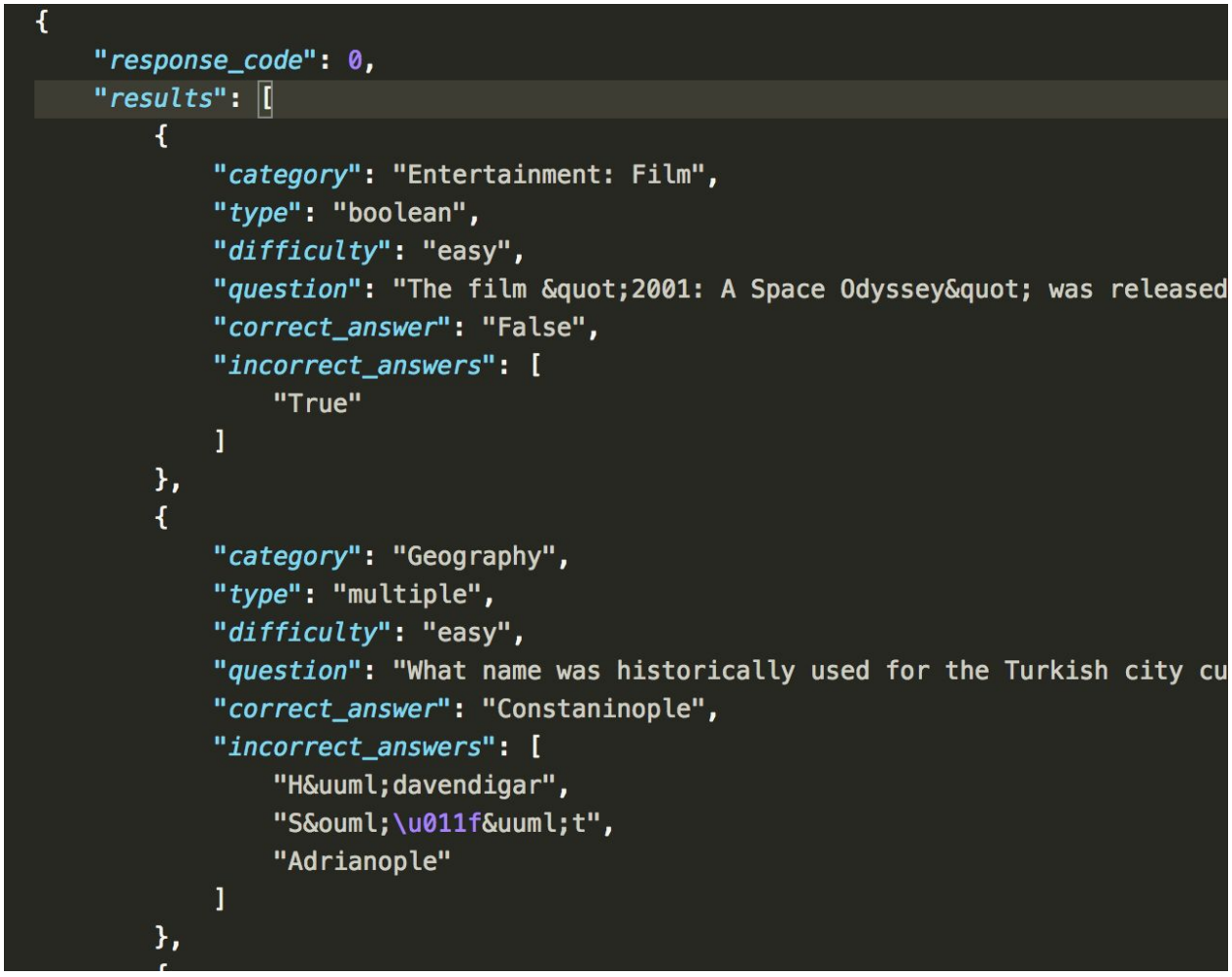
```
public class Category{

   public int id;

   public string name;

}
```

We are making a Category collection in CategoryData class. This is why we have to make it `System.Serializable`. Same rules applies same for other contracts except token contract. We don't have to make a collection of it. This way, your contract couldn't have to serializable.



Last example for the Question contract. As you see above, response code and result objects doesn't match the contract and we have to fix it with JsonHelper class. But another challenge to solve is that different types contains in the same response. Rather than creating different conracts, I've decided to make a generic one and decide to how to display or modificate in game scene. This is a choice by design but this is not the only way to do it.

Player Files in Resources

In resources folder in unity, PlayerData, CategoryData and QuizData files already created as scriptable object. You can create with right click in your resources or create button. if you delete or change them. But it's necessary to work project and relies on them. These data files must placed in Resources folder.

I think scriptable object very useful to carry data and use them as persistence data resource. But since they are resetting after you close the application. Working in editor there is no problem but on real device must think about saving data to device. I've used PlayerPrefs class but consider it's easy to find and change.