Getting Started

# What is this?

This is an API wrapper for the website and database Open Trivia.

<https://opentdb.com/>

It aims make implementing a quiz system into your application nice and easy.

# ☄ FEATURES

# ✓ 19 Categories

# ✓ 3 Difficulties

# ✓ Multiple Choice and True/False Questions

# ✓ Up to 50 questions

# ✓ 'Using' statements - Clean memory management!

# ✓ Native Unity classes - Efficient code stripping and minimal build sizes!

# ✓ Assembly Definition File - Reduces compile time! (2017.3+ Only)

# Is there some documentation I can view?

There certainly is!

<https://gaminganonymous.github.io/OpenTriviaUnityDocumentation/>

# Is the documentation available offline?

Unfortunately, I will not be including the documentation in an offline format (such as a PDF).  
This is so I do not have to resubmit to the asset store for a documentation update. (Which can take up to 2 weeks)

# Issues?

Email me!  
Carl.Gfinity@gmail.com

# Getting Started

***In the latest version there have been quite a few changes. Because of this, the documentation a behind of where I would like.***

Firstly, you want to add the Trivia prefab into your scene!  
(Found within Plugins/SimplyTrivia/Prefabs)

Now within your script handling this trivia, you should add the Namespaces:

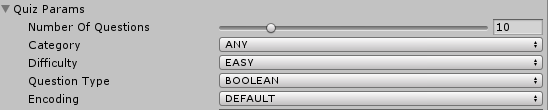
|  |
| --- |
| using SimplyTrivia;  using SimplyTrivia.Data; |

Now you can subscribe to events such as:

|  |
| --- |
| TriviaClient.Instance.OnTriviaQuizReceived  TriviaClient.Instance.OnTriviaErrorReceived |

Remember! You can modify the Trivia you are requesting from the editor, by creating a QuizParams field like so:

|  |
| --- |
| [SerializeField] private QuizParams quizParams = new QuizParams(); |



Now, make your first request!

|  |
| --- |
| TriviaClient.Instance.CreateQuiz(quizParams); |

This will fire any methods subscribe to the OnTriviaQuizReceived should it succeed. It returns a list of questions with their answers.