**StarLabs 2022 - Documentation**

***Moqk***

Moq is a mocking framework for C#/. NET. It is used in unit testing to isolate your class under test from its dependencies and ensure that the proper methods on the dependent objects are being called.

***How do you mock a service in C#?***

To mock you need an Interface, if not, you need to mark the methods that you want to mock as virtual. Under the hood the mocking framework is going to create a new implementation for you that behaves as you configured the mock

***Can we mock class in C#?***

You can use Moq to create mock objects that simulate or mimic a real object. Moq can be used to mock both classes and interfaces. However, there are a few limitations you should be aware of. The classes to be mocked can't be static or sealed, and the method being mocked should be marked as virtual

***Why?***

The library was created mainly for developers who aren't currently using any mocking library (or are displeased with the complexities of some other implementation), and who are typically manually writing their own mocks (with more or less "fanciness"). Most developers in this situation also happen to be quite pragmatic and adhere to state (or classic) TDD. It's the result of feeling that the barrier of entry from other mocking libraries is a bit high, and a simpler, more lightweight and elegant approach is possible. Moq achieves all this by taking full advantage of the elegant and compact C# and VB language features collectively known as LINQ (they are not just for queries, as the acronym implies).