Singleton Pattern

29 May 2021 06:13 AM

Q. Why do we need Singleton Pattern?

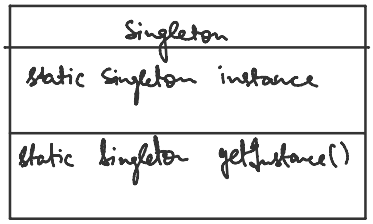
① If you need global access to an instance (object)
② And you need only one copy of that instance

The Singleton Pattern ensures a class has only one instance, and provides a global point of access to it.

Q. Why not to use?

① It gives global point of access → that means data is leaked in the whole codebase, it is difficult to handle as there will be a lot of code that is out of context.
② When the scope is global, the object can easily be changed by any operation throughout the app without the developer's knowledge.
③ Using Singleton is based on an assumption that you'll be requiring only one instance of that class.
④ Unit testing becomes difficult as we get the same instance and can't mock it.

Q. How it works?



It works by making the constructor PRIVATE
↓
static method in the Singleton creates an instance of Singleton

Singleton.getInstance() ✓

~~Singleton S = new Singleton()~~
~~S.getInstance()~~