**The Singleton Pattern**

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

* We make constructor of the class ‘private’ 🡪 which means outside people cannot construct that class with ‘new’ keyword.
* We have public static method that allows us to instantiate the class only one time, provide a global point of access to it!

|  |
| --- |
| **Singleton.ts** |
| **class** Singleton {    **private static** *uniqueInstance*: Singleton;    **private constructor**(){ }    **public static** *getInstance*(): Singleton {  **if**(**this**.*uniqueInstance* == **null**) {  **this**.*uniqueInstance* = **new** Singleton();  }  **return this**.*uniqueInstance*;  } } |