Table of Contents

Quick Start

Create a class that inherits from [Singleton](#).

```csharp
public class SomeManager : Singleton<SomeManager>
{
    public string HelloWorld = "Hello, World!";
}
```

That's it! Now, whenever `SomeManager.Instance` is called, `Singleton` will search for an existing instance of `SomeManager` in the scene or create a new one. Now, any script can access `HelloWorld` without needing to have a local reference of `SomeManager`:

```csharp
public class SomeOtherScript : MonoBehaviour
{
    private void Start()
    {
        Debug.Log(SomeManager.Instance.HelloWorld);
    }
}
```

If multiple instances of `SomeManager` are found, they are deleted. This check is only done when `Instance == null` and `Singleton` has to search for existing instances to return.

If you do not want a new `SomeManager` to be created when `Instance == null`, use [DontCreateNewSingleton](#) instead.