

## LinksPlatform's Platform.Singletons Class Library

./Default[T].cs

```
1 using System;
2
3 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
4
5 namespace Platform.Singletons
6 {
7     /// <summary>
8     /// Представляет собой точку доступа к экземплярам типов по умолчанию (созданных с помощью
9     /// → конструктора без аргументов).
10    /// </summary>
11    /// <typeparam name="T">Тип экземпляра объекта.</typeparam>
12    public static class Default<T>
13    {
14        [ThreadStatic]
15        private static T _threadInstance;
16
17        public static readonly T Instance = new T();
18
19        /// <summary>
20        /// If you really need maximum performance, use this property.
21        /// This property should create only one instance per thread.
22        /// </summary>
23        public static T ThreadInstance => _threadInstance == null ? _threadInstance = new T() :
24        → _threadInstance; //-V3111
25    }
26 }
```

./Global.cs

```
1 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
2
3 namespace Platform.Singletons
4 {
5     public static class Global
6     {
7         /// <summary>
8         /// Представляет поле-помойку, куда можно сбрасывать ненужные значения.
9         /// В некоторых случаях это может помочь избежать нежелательной оптимизации
10        /// (исключения записи в переменную в конце функции)
11        /// и сделать вид, что значение действительно используется.
12        /// Такое может быть полезно при реализации тестов на производительность.
13        /// </summary>
14        public static object Trash;
15    }
16 }
```

./Singleton.cs

```
1 using System;
2 using Platform.Interfaces;
3
4 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
5
6 namespace Platform.Singletons
7 {
8     public static class Singleton
9     {
10        public static Singleton<T> Create<T>(Func<T> creator) => new Singleton<T>(creator);
11        public static Singleton<T> Create<T>(IFactory<T> factory) => new
12        → Singleton<T>(factory.Create);
13        public static T Get<T>(Func<T> creator) => new Singleton<T>(creator).Instance;
14        public static T Get<T>(IFactory<T> factory) => new Singleton<T>(factory.Create).Instance;
15    }
16 }
```

./Singleton[T].cs

```
1 using System;
2 using System.Collections.Concurrent;
3 using System.Reflection;
4 using Platform.Collections.Lists;
5 using Platform.Reflection;
6
7 #pragma warning disable CS1591 // Missing XML comment for publicly visible type or member
8
9 namespace Platform.Singletons
10 {
11     public struct Singleton<T>
12     {
```

```
13     private static readonly ConcurrentDictionary<Func<T>, byte[]> _functions = new
        ↳ ConcurrentDictionary<Func<T>, byte[]>();
14     private static readonly ConcurrentDictionary<byte[], T> _singletons = new
        ↳ ConcurrentDictionary<byte[], T>(Default<IListEqualityComparer<byte[]>>.Instance);
15     public T Instance { get; }
16     public Singleton(Func<T> creator) => Instance =
        ↳ _singletons.GetOrAdd(_functions.GetOrAdd(creator,
        ↳ creator.GetMethodInfo().GetILBytes()), key => creator());
17     }
18 }
```

## Index

./Default[T].cs, 1  
./Global.cs, 1  
./Singleton.cs, 1  
./Singleton[T].cs, 1