

LinksPlatform's Platform.Timestamps Class Library

1.1 ./Timestamp.cs

```
1 using System;
2
3 namespace Platform.Timestamps
4 {
5     /// <summary>
6     /// <para>Represents a timestamp.</para>
7     /// <para>Представляет метку времени.</para>
8     /// </summary>
9     /// <remarks>
10    /// <para>To make this timestamp truly unique, it is recommended to use <see
11    ↪ cref="UniqueTimestampFactory"/>.</para>
12    /// <para>Чтобы эта метка времени была действительно уникальна рекомендуется использовать
13    ↪ <see cref="UniqueTimestampFactory"/>.</para>
14    /// </remarks>
15    public struct Timestamp : IEquatable<Timestamp>
16    {
17        /// <summary>
18        /// <para>Returns a string containg the default DateTime format for Timestamp.</para>
19        /// <para>Возвращает строку, содержащую формат даты и времени по умолчанию для метки
20        ↪ времени.</para>
21        /// </summary>
22        public static readonly string DefaultFormat = "yyyy.MM.dd hh:mm:ss.ffffff";
23
24        /// <summary>
25        /// <para>Gets or sets the number of ticks that represent the date and time in
26        ↪ UTC.</para>
27        /// <para>Возвращает или устанавливает количество тиков, которые представляют дату и
28        ↪ время в UTC.</para>
29        /// </summary>
30        public readonly ulong Ticks;
31
32        /// <summary>
33        /// <para>Creates a timestamp.</para>
34        /// <para>Создаёт метку времени.</para>
35        /// </summary>
36        /// <param name="ticks"><para>A number representing the number of
37        ↪ ticks.</para><para>Число представляющие количество тиков.</para></param>
38        public Timestamp(ulong ticks) => Ticks = ticks;
39
40        /// <summary>
41        /// <para>Defines an implicit conversion of a DateTime to a Timestamp.</para>
42        /// <para>Определяет неявное преобразование DateTime в метку времени.</para>
43        /// </summary>
44        /// <param name="dateTime"><para>The DateTime struct.</para><para>Структура
45        ↪ DateTime.</para></param>
46        public static implicit operator Timestamp(DateTime dateTime) => new
47        ↪ Timestamp((ulong)dateTime.ToUniversalTime().Ticks);
48
49        /// <summary>
50        /// <para>Defines an implicit conversion of a Timestamp to a DateTime.</para>
51        /// <para>Определяет неявное преобразование метки времени в DateTime.</para>
52        /// </summary>
53        /// <param name="timestamp"><para>The Timestamp.</para><para>Отметка
54        ↪ времени.</para></param>
55        public static implicit operator DateTime(Timestamp timestamp) => new
56        ↪ DateTime((long)timestamp.Ticks, DateTimeKind.Utc);
57
58        /// <summary>
59        /// <para>Defines an implicit conversion of a 64-bit unsigned integer to a
60        ↪ Timestamp.</para>
61        /// <para>Определяет неявное преобразование 64-разрядного целого числа без знака в метку
62        ↪ времени.</para>
63        /// </summary>
64        /// <param name="ticks"><para>The number of ticks represented as a 64-bit
65        ↪ integer.</para><para>Количество тиков представленное в виде 64-разрядного целого
66        ↪ числа.</para></param>
67        public static implicit operator Timestamp(ulong ticks) => new Timestamp(ticks);
68
69        /// <summary>
70        /// <para>Defines an implicit conversion of a Timestamp to a 64-bit unsigned
71        ↪ integer.</para>
72        /// <para>Определяет неявное преобразование метки времени в 64-разрядное целое число без
73        ↪ знака.</para>
74        /// </summary>
```

```

59     /// <param name="timestamp"><para>The Timestamp.</para><para>Отметка
    ↪ времени.</para></param>
60     public static implicit operator ulong(Timestamp timestamp) => timestamp.Ticks;
61
62     /// <summary>
63     /// <para>Returns a string that represents the current Timestamp.</para>
64     /// <para>Возвращает строку, которая представляет текущую метку времени.</para>
65     /// </summary>
66     /// <returns><para>A string that represents the current Timestamp.</para><para>Строка,
    ↪ представляющая текущую метку времени.</para></returns>
67     public override string ToString() => ((DateTime)this).ToString(DefaultFormat);
68
69     /// <summary>
70     /// <para>Определяет, равна ли текущая отметка времени другой отметке времени.</para>
71     /// <para>Indicates whether the current Timestamp is equal to another Timestamp.</para>
72     /// </summary>
73     /// <param name="other"><para>Other Timestamp.</para><para>Другая отметка
    ↪ времени.</para></param>
74     /// <returns><para>True if the current Timestamp is equal to the other Timestamp;
    ↪ otherwise, false.</para><para>Истину, если текущая отметка времени равна другой
    ↪ отметке времени; иначе ложь.</para></returns>
75     public bool Equals(Timestamp other) => Ticks == other.Ticks;
76
77     /// <summary>
78     /// <para>Determines whether the specified object is equal to the current object.</para>
79     /// <para>Определяет, равен ли указанный объект текущему объекту.</para>
80     /// </summary>
81     /// <param name="obj"><para>The object to compare with the current
    ↪ object.</para><para>Объект для сравнения с текущим объектом.</para></param>
82     /// <returns><para>True if the specified object is equal to the current object;
    ↪ otherwise, false.</para><para>Истину, если указанный объект равен текущему объекту;
    ↪ иначе ложь.</para></returns>
83     public override bool Equals(object obj) => obj is Timestamp timestamp ?
    ↪ Equals(timestamp) : false;
84
85     /// <summary>
86     /// <para>Serves as the default hash function.</para>
87     /// <para>Служит в качестве хэш-функции по умолчанию.</para>
88     /// </summary>
89     /// <returns><para>A hash code for the current object.</para><para>Хеш-код для текущего
    ↪ объекта.</para></returns>
90     public override int GetHashCode() => Ticks.GetHashCode();
91 }
92 }

```

1.2 ./UniqueTimestampFactory.cs

```

1  using System;
2  using Platform.Interfaces;
3
4  namespace Platform.Timestamps
5  {
6      /// <summary>
7      /// <para>Represents a factory for creating unique timestamps.</para>
8      /// <para>Представляет фабрику по созданию уникальных отметок времени.</para>
9      /// </summary>
10     public class UniqueTimestampFactory : IFactory<Timestamp>
11     {
12         private ulong _lastTicks;
13
14         /// <summary>
15         /// <para>Creates a timestamp corresponding to the current UTC date and time or next
    ↪ unique timestamp.</para>
16         /// <para>Создаёт отметку времени соответствующую текущей дате и времени по UTC или
    ↪ следующую уникальную отметку времени.</para>
17         /// </summary>
18         public Timestamp Create()
19         {
20             var utcTicks = (ulong)DateTime.UtcNow.Ticks;
21             return utcTicks <= _lastTicks ? new Timestamp(_lastTicks++) : new
    ↪ Timestamp(utcTicks);
22         }
23     }
24 }

```

Index

./Timestamp.cs, 1

./UniqueTimestampFactory.cs, 2