

# LinksPlatform's Platform.Timestamps Class Library

./Timestamp.cs

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

```

68
69     /// <summary>
70     /// Определяет, равна ли текущая отметка времени другой отметке времени.
71     /// Indicates whether the current Timestamp is equal to another Timestamp.
72     /// </summary>
73     /// <param name="other">Other Timestamp. Другая отметка времени.</param>
74     /// <returns>True if the current Timestamp is equal to the other Timestamp; otherwise,
75     ↪ false. Истину, если текущая отметка времени равна другой отметке времени; иначе
76     ↪ ложь.</returns>
77     public bool Equals(Timestamp other) => Ticks == other.Ticks;
78 }
79

```

## ./UniqueTimestampFactory.cs

```

1  using System;
2  using Platform.Interfaces;
3
4  namespace Platform.Timestamps
5  {
6      /// <summary>
7      /// Represents a factory for creating unique timestamps.
8      /// Представляет фабрику по созданию уникальных отметок времени.
9      /// </summary>
10     public class UniqueTimestampFactory : IFactory<Timestamp>
11     {
12         private ulong _lastTicks;
13
14         /// <summary>
15         /// Creates a timestamp corresponding to the current UTC date and time or next unique
16         ↪ timestamp
17         /// Создаёт отметку времени соответствующую текущей дате и времени по UTC или следующую
18         ↪ уникальную отметку времени.
19         /// </summary>
20         public Timestamp Create()
21         {
22             var utcTicks = (ulong)DateTime.UtcNow.Ticks;
23             return utcTicks <= _lastTicks ? new Timestamp(_lastTicks++) : new
24             ↪ Timestamp(utcTicks);
25         }
26     }
27 }
28

```

## Index

./Timestamp.cs, 1

./UniqueTimestampFactory.cs, 2