

TAP Auction Site Component

Generated by Doxygen 1.9.3

1 TAP Auction Site Component	1
1.1 Introduction	1
1.2 Global requirements	2
1.3 Implementation requirements	2
2 Namespace Index	3
2.1 Packages	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 Namespace Documentation	9
5.1 TAP21_22_AuctionSite Namespace Reference	9
5.2 TAP21_22_AuctionSite.Interface Namespace Reference	9
6 Class Documentation	11
6.1 AuctionSiteArgumentException Class Reference	11
6.1.1 Detailed Description	12
6.1.2 Constructor & Destructor Documentation	12
6.1.2.1 AuctionSiteArgumentException() [1/6]	12
6.1.2.2 AuctionSiteArgumentException() [2/6]	12
6.1.2.3 AuctionSiteArgumentException() [3/6]	12
6.1.2.4 AuctionSiteArgumentException() [4/6]	12
6.1.2.5 AuctionSiteArgumentException() [5/6]	13
6.1.2.6 AuctionSiteArgumentException() [6/6]	13
6.1.3 Property Documentation	13
6.1.3.1 ParamName	13
6.2 AuctionSiteArgumentNullException Class Reference	13
6.2.1 Detailed Description	14
6.2.2 Constructor & Destructor Documentation	14
6.2.2.1 AuctionSiteArgumentNullException() [1/4]	14
6.2.2.2 AuctionSiteArgumentNullException() [2/4]	14
6.2.2.3 AuctionSiteArgumentNullException() [3/4]	14
6.2.2.4 AuctionSiteArgumentNullException() [4/4]	14
6.3 AuctionSiteArgumentOutOfRangeException Class Reference	15
6.3.1 Detailed Description	15
6.3.2 Constructor & Destructor Documentation	16
6.3.2.1 AuctionSiteArgumentOutOfRangeException() [1/7]	16
6.3.2.2 AuctionSiteArgumentOutOfRangeException() [2/7]	16
6.3.2.3 AuctionSiteArgumentOutOfRangeException() [3/7]	16
6.3.2.4 AuctionSiteArgumentOutOfRangeException() [4/7]	16

6.3.2.5 AuctionSiteArgumentOutOfRangeException() [5/7]	16
6.3.2.6 AuctionSiteArgumentOutOfRangeException() [6/7]	16
6.3.2.7 AuctionSiteArgumentOutOfRangeException() [7/7]	17
6.3.3 Property Documentation	17
6.3.3.1 ActualValue	17
6.4 AuctionSiteConcurrentChangeException Class Reference	17
6.4.1 Detailed Description	18
6.4.2 Constructor & Destructor Documentation	18
6.4.2.1 AuctionSiteConcurrentChangeException() [1/4]	18
6.4.2.2 AuctionSiteConcurrentChangeException() [2/4]	18
6.4.2.3 AuctionSiteConcurrentChangeException() [3/4]	18
6.4.2.4 AuctionSiteConcurrentChangeException() [4/4]	18
6.5 AuctionSiteException Class Reference	19
6.5.1 Detailed Description	19
6.5.2 Constructor & Destructor Documentation	19
6.5.2.1 AuctionSiteException() [1/4]	19
6.5.2.2 AuctionSiteException() [2/4]	20
6.5.2.3 AuctionSiteException() [3/4]	20
6.5.2.4 AuctionSiteException() [4/4]	20
6.6 AuctionSiteInexistentNameException Class Reference	20
6.6.1 Detailed Description	21
6.6.2 Constructor & Destructor Documentation	21
6.6.2.1 AuctionSiteInexistentNameException() [1/4]	21
6.6.2.2 AuctionSiteInexistentNameException() [2/4]	21
6.6.2.3 AuctionSiteInexistentNameException() [3/4]	21
6.6.2.4 AuctionSiteInexistentNameException() [4/4]	22
6.6.3 Property Documentation	22
6.6.3.1 Name	22
6.7 AuctionSiteInvalidOperationException Class Reference	22
6.7.1 Detailed Description	23
6.7.2 Constructor & Destructor Documentation	23
6.7.2.1 AuctionSiteInvalidOperationException() [1/4]	23
6.7.2.2 AuctionSiteInvalidOperationException() [2/4]	23
6.7.2.3 AuctionSiteInvalidOperationException() [3/4]	23
6.7.2.4 AuctionSiteInvalidOperationException() [4/4]	23
6.8 AuctionSiteNameAlreadyInUseException Class Reference	24
6.8.1 Detailed Description	24
6.8.2 Constructor & Destructor Documentation	24
6.8.2.1 AuctionSiteNameAlreadyInUseException() [1/4]	25
6.8.2.2 AuctionSiteNameAlreadyInUseException() [2/4]	25
6.8.2.3 AuctionSiteNameAlreadyInUseException() [3/4]	25
6.8.2.4 AuctionSiteNameAlreadyInUseException() [4/4]	25

6.8.3 Property Documentation	25
6.8.3.1 Name	25
6.9 AuctionSiteUnavailableDbException Class Reference	26
6.9.1 Detailed Description	26
6.9.2 Constructor & Destructor Documentation	26
6.9.2.1 AuctionSiteUnavailableDbException() [1/4]	26
6.9.2.2 AuctionSiteUnavailableDbException() [2/4]	27
6.9.2.3 AuctionSiteUnavailableDbException() [3/4]	27
6.9.2.4 AuctionSiteUnavailableDbException() [4/4]	27
6.10 AuctionSiteUnavailableTimeMachineException Class Reference	27
6.10.1 Detailed Description	28
6.10.2 Constructor & Destructor Documentation	28
6.10.2.1 AuctionSiteUnavailableTimeMachineException() [1/4]	28
6.10.2.2 AuctionSiteUnavailableTimeMachineException() [2/4]	28
6.10.2.3 AuctionSiteUnavailableTimeMachineException() [3/4]	28
6.10.2.4 AuctionSiteUnavailableTimeMachineException() [4/4]	28
6.11 DomainConstraints Class Reference	29
6.11.1 Detailed Description	29
6.11.2 Member Data Documentation	29
6.11.2.1 MaxSiteName	29
6.11.2.2 MaxTimeZone	29
6.11.2.3 MaxUserName	30
6.11.2.4 MinSiteName	30
6.11.2.5 MinTimeZone	30
6.11.2.6 MinUserName	30
6.11.2.7 MinUserPassword	30
6.12 IAuction Interface Reference	30
6.12.1 Detailed Description	31
6.12.2 Member Function Documentation	31
6.12.2.1 Bid()	31
6.12.2.2 CurrentPrice()	32
6.12.2.3 CurrentWinner()	33
6.12.2.4 Delete()	33
6.12.3 Property Documentation	33
6.12.3.1 Description	33
6.12.3.2 EndsOn	33
6.12.3.3 Id	33
6.12.3.4 Seller	33
6.13 IHost Interface Reference	34
6.13.1 Detailed Description	34
6.13.2 Member Function Documentation	34
6.13.2.1 CreateSite()	34

6.13.2.2 GetSiteInfos()	35
6.13.2.3 LoadSite()	35
6.14 IHostFactory Interface Reference	36
6.14.1 Detailed Description	36
6.14.2 Member Function Documentation	36
6.14.2.1 CreateHost()	36
6.14.2.2 LoadHost()	37
6.15 ISession Interface Reference	37
6.15.1 Detailed Description	38
6.15.2 Member Function Documentation	38
6.15.2.1 CreateAuction()	38
6.15.2.2 Logout()	39
6.15.3 Property Documentation	39
6.15.3.1 Id	39
6.15.3.2 User	39
6.15.3.3 ValidUntil	39
6.16 ISite Interface Reference	39
6.16.1 Detailed Description	40
6.16.2 Member Function Documentation	40
6.16.2.1 CreateUser()	40
6.16.2.2 Delete()	41
6.16.2.3 Login()	41
6.16.2.4 Now()	42
6.16.2.5 ToyGetAuctions()	42
6.16.2.6 ToyGetSessions()	42
6.16.2.7 ToyGetUsers()	43
6.16.3 Property Documentation	43
6.16.3.1 MinimumBidIncrement	43
6.16.3.2 Name	43
6.16.3.3 SessionExpirationInSeconds	43
6.16.3.4 Timezone	43
6.17 IUser Interface Reference	44
6.17.1 Detailed Description	44
6.17.2 Member Function Documentation	44
6.17.2.1 Delete()	44
6.17.2.2 WonAuctions()	44
6.17.3 Property Documentation	45
6.17.3.1 Username	45
6.18 TapDbContext Class Reference	45
6.18.1 Detailed Description	45
6.18.2 Constructor & Destructor Documentation	46
6.18.2.1 TapDbContext() [1/2]	46

6.18.2.2 TapDbContext() [2/2]	46
6.18.3 Property Documentation	46
6.18.3.1 NumberOfCreatedContexts	46
Index	47

Chapter 1

TAP Auction Site Component

1.1 Introduction

The namespace `TAP21_22.AuctionSite` contains the declarations of a set of interfaces modeling the required types for managing toy auction sites (think ebay, think smaller, smaller, smaller...).

Implementations of the interfaces in `AlarmClock.Interfaces` will be provided by the `AuctionSite` clients and are not part of this project.

Only interfaces in `AuctionSite.Interfaces` are implemented by the `AuctionSite` component.

Using the `IHostFactory` a client of this component can initialize or load a hosting service for auction sites.

On such a hosting system `IHost`, new sites can be created and existing ones can be loaded and made available to their final users.

An `ISite` represents a site for online auctions, managing users, their sessions, and their auctions. It has a unique name, which is a non empty nor null string, and defines

- the timezone used to get the time;
- the minimum increment allowed in bidding;
- the time out of inactive sessions.

It is the root of an aggregate, in the sense that it owns its users, sessions and auctions, that have no meaning outside their site.

Users, represented by `IUser`, must have a valid session to act on a site, and no user can have two valid sessions at the same time.

Sessions, represented by `ISession`, start with a login, and may be deleted by an explicit logout, or may time out because of a prolonged inactivity. Each time users access their sessions by an explicit login, by creating an auction or by bidding on an auction, the expiration time is reset to the site expiration time. Every five minutes expired sessions must be deleted from the database.

Time management is provided by a required component implementing the interface `TAP21_22.AlarmClock.IAlarmClock`. The main type in such component is `IAlarmClock`, and it represents a clock synchronized on a specific time zone. `IAlarmClock` has methods for checking the current time, and setting an alarm, that is, firing the ringing event at a given time.

This is just an overview; each type and each method has its own, more detailed, specific documentation.

1.2 Global requirements

Names and Id, when provided for a type, are unique within the context where the type is meaningful. Thus, for instance, the name of an auction site is unique in a host, as well as the name of a user within an auction site (but the same name can be repeated in different auction sites).

Any attempt to create two different elements of the same type with the same name (or Id) must fail by throwing `AuctionSiteNameAlreadyInUseException`.

Each method that receives:

- a null argument must throw the exception `AuctionSiteArgumentNullException`, unless explicitly specified
- a string that is too short or too long must throw `AuctionSiteArgumentException`; the allowed string length ranges are contained in `DomainConstraints`

Any method invocation on a deleted object, that is an object whose persistent counterpart has been removed, must throw `AuctionSiteInvalidOperationException`.

Any generic failure to persist or retrieve the data to/from the DB must be communicated by throwing `AuctionSiteUnavailableDbException`.

Any detected attempt to save an entity on the DB which has been concurrently modified must be communicated by throwing `AuctionSiteConcurrentChangeException`. Concurrency management is not required for a project to be acceptable. But, it is an element of evaluation, and if you want to do it, then you must use `AuctionSiteConcurrentChangeException`.

Your implementation must not throw any exception that has not been explicitly listed here.

Please note that these requirements are intentionally *not* repeated for each and every method of the specification, and must be met by all your methods (implementing the interfaces described by this document; private methods can behave as they please ;-)

1.3 Implementation requirements

You're required to provide a Visual Studio 2019 solution for .NET 5 containing an implementation of this specification. The solution must be pushed on your personal repository created on GitHub when you accept the invitation to the project assignment.

All type declarations must be contained in a namespace whose name correspond to your family name (if it includes unacceptable chars, replace them by `_`).

With the obvious exception of connecting with the DB referenced by the connection string passed to the methods of the `IHostFactory`, it is forbidden to open or create files/registry keys/..., or establish database/network/... connections and so on.

To interact with the databases, you must use EF Core 5 and your `DbContext` class must extend `TAP21_22.AuctionSite.Interfaces.TapDb`.

Your DLL must not depend on any library, other than the `TAP21_22.AuctionSite.Interfaces`, `TAP21_22.AlarmClock.Interfaces`, EF Core 5 and standard .NET assemblies. You can use other libraries only if they have been explicitly approved in the TAP Forum by the teacher (that is, if you think there is a useful library out there, just ask on the forum if you can use it... the answer will most probably be yes, but you have nonetheless to explicitly ask for it).

Before delivering your work, please keep in mind that passing the tests is a minimum requirement only. The fact that your implementation passes these test does not imply, of course, its correctness. Indeed, many other tests and code inspection will be used to evaluate your work and it will be evaluated *once*.

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

TAP21_22_AuctionSite	9
TAP21_22_AuctionSite.Interface	9

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

DbContext	
TapDbContext	45
DomainConstraints	29
Exception	
AuctionSiteException	19
AuctionSiteArgumentException	11
AuctionSiteArgumentOutOfRangeException	15
AuctionSiteArgumentNullException	13
AuctionSiteConcurrentChangeException	17
AuctionSiteInexistentNameException	20
AuctionSiteInvalidOperationException	22
AuctionSiteNameAlreadyInUseException	24
AuctionSiteUnavailableDbException	26
AuctionSiteUnavailableTimeMachineException	27
IAuction	30
IHost	34
IHostFactory	36
ISession	37
ISite	39
IUser	44

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AuctionSiteArgumentException	11
Thrown when an argument of a method call violates the method preconditions	
AuctionSiteArgumentNullException	13
Thrown when an unacceptable null argument is used for a method call	
AuctionSiteArgumentOutOfRangeException	15
Thrown when an argument of a method call does not meet the expected value range for the corresponding parameter	
AuctionSiteConcurrentChangeException	17
Thrown to notify the attempt to save an entity on the DB which has been concurrently modified (concurrency management is not required; but, if you want to do it use this exception)	
AuctionSiteException	19
The root for the exception hierarchy of this component. To be used only if no more specific exception is available for the error (it should not happen that you ever need to throw it).	
AuctionSiteInexistentNameException	20
Thrown to notify the attempt use its name to access an entity which is not available on the database.	
AuctionSiteInvalidOperationException	22
Thrown to notify an attempt at invoking an operation on an object in an invalid state (for instance a deleted entity)	
AuctionSiteNameAlreadyInUseException	24
Thrown to notify the attempt to create an entity whose name is already in use. For instance, a site with the same name of another site managed by the same host, or a user with the same name of another user on the same site.	
AuctionSiteUnavailableDbException	26
Thrown to notify that no connection with the database has been established.	
AuctionSiteUnavailableTimeMachineException	27
Thrown to notify the attempt to create an auction with expiration date in the past.	
DomainConstraints	29
Numeric constants used to express constraints on names, passwords, and time zone	
IAuction	30
The auctions managed by sites. Equals on auctions must be true iff the two auctions belong to the same site and have the same Id.	
IHost	34
The hosting service A Host own a database and uses it to save the data of the sites it manages. Any Host may create and load its sites.	

[IHostFactory](#)

This component factory. All its methods have a parameter `connectionString` representing the connection string of a Microsoft SQL Server DB, used to permanently store the data of the auction sites managed by a specific Host and to implicitly characterize the Host. Many different objects of type [IHost](#) may at the same time represent the same Host, concurrently working on the same database. 36

[ISession](#)

The sessions of the users on the site. It may become invalid (expire) if the user is idle for the expiration time set for the site. Equals on sessions must be true iff the two sessions have the same Id. 37

[ISite](#)

The auction sites. Equals on sites must be true iff the two sites have the same Name. 39

[IUser](#)

The users of the auction system. Equals on users must be true iff the two users belong to the same site and have the same Username. 44

[TapDbContext](#)

To be extended by the component DbContext, to simplify advanced testing 45

Chapter 5

Namespace Documentation

5.1 TAP21_22_AuctionSite Namespace Reference

Namespaces

- namespace [Interface](#)

5.2 TAP21_22_AuctionSite.Interface Namespace Reference

Classes

- class [AuctionSiteArgumentException](#)
Thrown when an argument of a method call violates the method preconditions
- class [AuctionSiteArgumentNullException](#)
Thrown when an unacceptable null argument is used for a method call
- class [AuctionSiteArgumentOutOfRangeException](#)
Thrown when an argument of a method call does not meet the expected value range for the corresponding parameter
- class [AuctionSiteConcurrentChangeException](#)
Thrown to notify the attempt to save an entity on the DB which has been concurrently modified (concurrency management is not required; but, if you want to do it use this exception)
- class [AuctionSiteException](#)
The root for the exception hierarchy of this component. To be used only if no more specific exception is available for the error (it should not happen that you ever need to throw it).
- class [AuctionSiteInexistentNameException](#)
Thrown to notify the attempt use its name to access an entity which is not available on the database.
- class [AuctionSiteInvalidOperationException](#)
Thrown to notify an attempt at invoking an operation on an object in an invalid state (for instance a deleted entity)
- class [AuctionSiteNameAlreadyInUseException](#)
Thrown to notify the attempt to create an entity whose name is already in use. For instance, a site with the same name of another site managed by the same host, or a user with the same name of another user on the same site.
- class [AuctionSiteUnavailableDbException](#)
Thrown to notify that no connection with the database has been established.
- class [AuctionSiteUnavailableTimeMachineException](#)
Thrown to notify the attempt to create an auction with expiration date in the past.
- class [DomainConstraints](#)

Numeric constants used to express constraints on names, passwords, and time zone

- interface [IAuction](#)

The auctions managed by sites. Equals on auctions must be true iff the two auctions belong to the same site and have the same Id.

- interface [IHost](#)

The hosting service A Host own a database and uses it to save the data of the sites it manages. Any Host may create and load its sites.

- interface [IHostFactory](#)

This component factory. All its methods have a parameter `connectionString` representing the connection string of a Microsoft SQL Server DB, used to permanently store the data of the auction sites managed by a specific Host and to implicitly characterize the Host. Many different objects of type [IHost](#) may at the same time represent the same Host, concurrently working on the same database.

- interface [ISession](#)

The sessions of the users on the site. It may become invalid (expire) if the user is idle for the expiration time set for the site. Equals on sessions must be true iff the two sessions have the same Id.

- interface [ISite](#)

The auction sites. Equals on sites must be true iff the two sites have the same Name.

- interface [IUser](#)

The users of the auction system. Equals on users must be true iff the two users belong to the same site and have the same Username.

- class [TapDbContext](#)

To be extended by the component `DbContext`, to simplify advanced testing

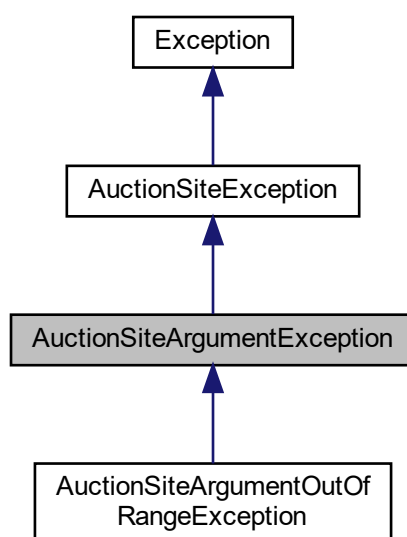
Chapter 6

Class Documentation

6.1 AuctionSiteArgumentException Class Reference

Thrown when an argument of a method call violates the method preconditions

Inheritance diagram for AuctionSiteArgumentException:



Public Member Functions

- [AuctionSiteArgumentException](#) ()
- [AuctionSiteArgumentException](#) (string? message)
- [AuctionSiteArgumentException](#) (string? message, Exception inner)
- [AuctionSiteArgumentException](#) (string? message, string? paramName)
- [AuctionSiteArgumentException](#) (string? message, string? paramName, Exception inner)

Protected Member Functions

- [AuctionSiteArgumentException](#) (SerializationInfo info, StreamingContext context)

Properties

- virtual ? string [ParamName](#) [get]

6.1.1 Detailed Description

Thrown when an argument of a method call violates the method preconditions

6.1.2 Constructor & Destructor Documentation

6.1.2.1 AuctionSiteArgumentException() [1/6]

```
AuctionSiteArgumentException ( )
```

6.1.2.2 AuctionSiteArgumentException() [2/6]

```
AuctionSiteArgumentException (
    string? message )
```

6.1.2.3 AuctionSiteArgumentException() [3/6]

```
AuctionSiteArgumentException (
    string? message,
    Exception inner )
```

6.1.2.4 AuctionSiteArgumentException() [4/6]

```
AuctionSiteArgumentException (
    string? message,
    string? paramName )
```

6.1.2.5 AuctionSiteArgumentException() [5/6]

```
AuctionSiteArgumentException (
    string? message,
    string? paramName,
    Exception inner )
```

6.1.2.6 AuctionSiteArgumentException() [6/6]

```
AuctionSiteArgumentException (
    SerializationInfo info,
    StreamingContext context ) [protected]
```

6.1.3 Property Documentation

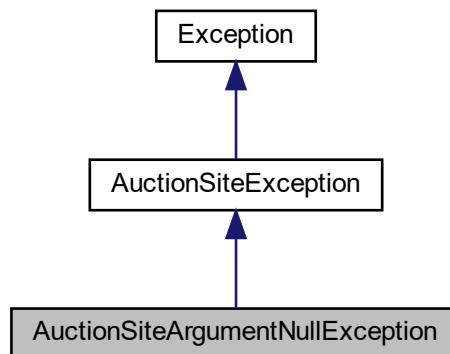
6.1.3.1 ParamName

```
virtual ? string ParamName [get]
```

6.2 AuctionSiteArgumentNullException Class Reference

Thrown when an unacceptable null argument is used for a method call

Inheritance diagram for AuctionSiteArgumentNullException:



Public Member Functions

- [AuctionSiteArgumentNullException](#) ()
- [AuctionSiteArgumentNullException](#) (string? message)
- [AuctionSiteArgumentNullException](#) (string? message, Exception inner)

Protected Member Functions

- [AuctionSiteArgumentNullException](#) (SerializationInfo info, StreamingContext context)

6.2.1 Detailed Description

Thrown when an unacceptable null argument is used for a method call

6.2.2 Constructor & Destructor Documentation

6.2.2.1 AuctionSiteArgumentNullException() [1/4]

```
AuctionSiteArgumentNullException ( )
```

6.2.2.2 AuctionSiteArgumentNullException() [2/4]

```
AuctionSiteArgumentNullException (
    string? message )
```

6.2.2.3 AuctionSiteArgumentNullException() [3/4]

```
AuctionSiteArgumentNullException (
    string? message,
    Exception inner )
```

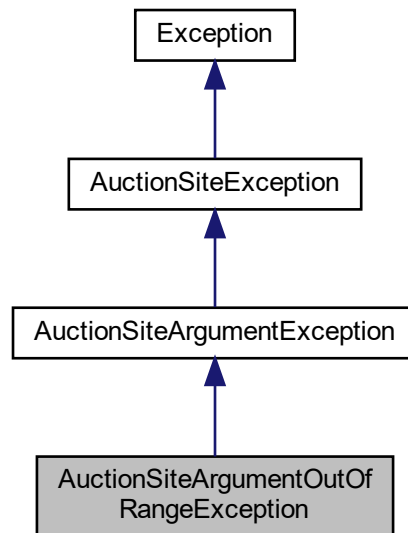
6.2.2.4 AuctionSiteArgumentNullException() [4/4]

```
AuctionSiteArgumentNullException (
    SerializationInfo info,
    StreamingContext context ) [protected]
```

6.3 AuctionSiteArgumentOutOfRangeException Class Reference

Thrown when an argument of a method call does not meet the expected value range for the corresponding parameter

Inheritance diagram for AuctionSiteArgumentOutOfRangeException:



Public Member Functions

- [AuctionSiteArgumentOutOfRangeException](#) ()
- [AuctionSiteArgumentOutOfRangeException](#) (string? message)
- [AuctionSiteArgumentOutOfRangeException](#) (string? message, Exception inner)
- [AuctionSiteArgumentOutOfRangeException](#) (string? paramName, string? message)
- [AuctionSiteArgumentOutOfRangeException](#) (string? paramName, string? message, Exception inner)
- [AuctionSiteArgumentOutOfRangeException](#) (string? paramName, object? value, string? message)

Protected Member Functions

- [AuctionSiteArgumentOutOfRangeException](#) (SerializationInfo info, StreamingContext context)

Properties

- virtual ? object [ActualValue](#) [get]

6.3.1 Detailed Description

Thrown when an argument of a method call does not meet the expected value range for the corresponding parameter

6.3.2 Constructor & Destructor Documentation

6.3.2.1 AuctionSiteArgumentOutOfRangeException() [1/7]

```
AuctionSiteArgumentOutOfRangeException ( )
```

6.3.2.2 AuctionSiteArgumentOutOfRangeException() [2/7]

```
AuctionSiteArgumentOutOfRangeException (
    string? message )
```

6.3.2.3 AuctionSiteArgumentOutOfRangeException() [3/7]

```
AuctionSiteArgumentOutOfRangeException (
    string? message,
    Exception inner )
```

6.3.2.4 AuctionSiteArgumentOutOfRangeException() [4/7]

```
AuctionSiteArgumentOutOfRangeException (
    string? paramName,
    string? message )
```

6.3.2.5 AuctionSiteArgumentOutOfRangeException() [5/7]

```
AuctionSiteArgumentOutOfRangeException (
    string? paramName,
    string? message,
    Exception inner )
```

6.3.2.6 AuctionSiteArgumentOutOfRangeException() [6/7]

```
AuctionSiteArgumentOutOfRangeException (
    string? paramName,
    object? value,
    string? message )
```


6.3.2.7 AuctionSiteArgumentOutOfRangeException() [7/7]

```
AuctionSiteArgumentOutOfRangeException (
    SerializationInfo info,
    StreamingContext context ) [protected]
```

6.3.3 Property Documentation

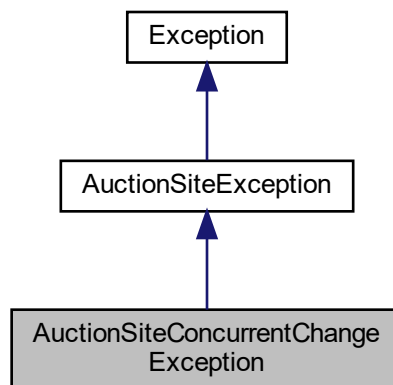
6.3.3.1 ActualValue

```
virtual ? object ActualValue [get]
```

6.4 AuctionSiteConcurrentChangeException Class Reference

Thrown to notify the attempt to save an entity on the DB which has been concurrently modified (concurrency management is not required; but, if you want to do it use this exception)

Inheritance diagram for AuctionSiteConcurrentChangeException:



Public Member Functions

- [AuctionSiteConcurrentChangeException \(\)](#)
- [AuctionSiteConcurrentChangeException \(string? message\)](#)
- [AuctionSiteConcurrentChangeException \(string? message, Exception inner\)](#)

Protected Member Functions

- [AuctionSiteConcurrentChangeException](#) (SerializationInfo info, StreamingContext context)

6.4.1 Detailed Description

Thrown to notify the attempt to save an entity on the DB which has been concurrently modified (concurrency management is not required; but, if you want to do it use this exception)

6.4.2 Constructor & Destructor Documentation

6.4.2.1 AuctionSiteConcurrentChangeException() [1/4]

```
AuctionSiteConcurrentChangeException ( )
```

6.4.2.2 AuctionSiteConcurrentChangeException() [2/4]

```
AuctionSiteConcurrentChangeException (
    string? message )
```

6.4.2.3 AuctionSiteConcurrentChangeException() [3/4]

```
AuctionSiteConcurrentChangeException (
    string? message,
    Exception inner )
```

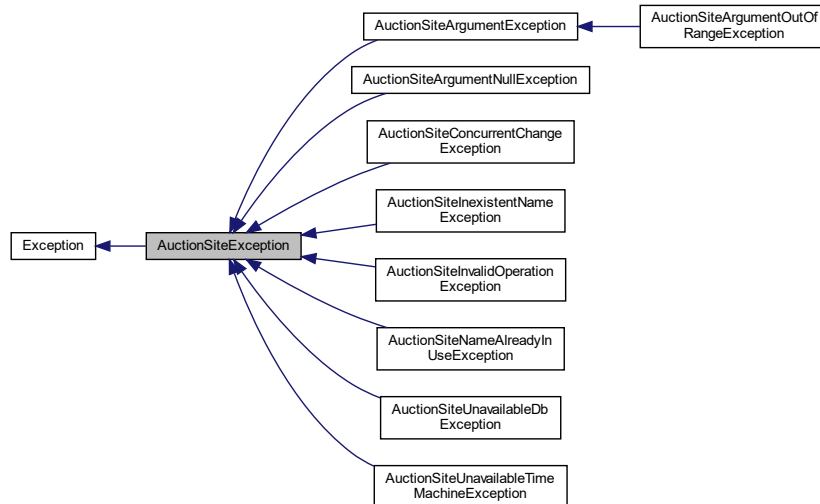
6.4.2.4 AuctionSiteConcurrentChangeException() [4/4]

```
AuctionSiteConcurrentChangeException (
    SerializationInfo info,
    StreamingContext context ) [protected]
```

6.5 AuctionSiteException Class Reference

The root for the exception hierarchy of this component. To be used only if no more specific exception is available for the error (it should not happen that you ever need to throw it).

Inheritance diagram for AuctionSiteException:



Protected Member Functions

- [AuctionSiteException](#) ()
- [AuctionSiteException](#) (string? message)
- [AuctionSiteException](#) (string? message, Exception inner)
- [AuctionSiteException](#) (SerializationInfo info, StreamingContext context)

6.5.1 Detailed Description

The root for the exception hierarchy of this component. To be used only if no more specific exception is available for the error (it should not happen that you ever need to throw it).

6.5.2 Constructor & Destructor Documentation

6.5.2.1 AuctionSiteException() [1/4]

```
AuctionSiteException ( ) [protected]
```

6.5.2.2 AuctionSiteException() [2/4]

```
AuctionSiteException (
    string? message ) [protected]
```

6.5.2.3 AuctionSiteException() [3/4]

```
AuctionSiteException (
    string? message,
    Exception inner ) [protected]
```

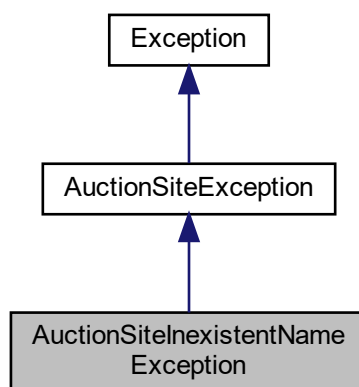
6.5.2.4 AuctionSiteException() [4/4]

```
AuctionSiteException (
    SerializationInfo info,
    StreamingContext context ) [protected]
```

6.6 AuctionSiteInexistentNameException Class Reference

Thrown to notify the attempt use its name to access an entity which is not available on the database.

Inheritance diagram for AuctionSiteInexistentNameException:



Public Member Functions

- [AuctionSiteInexistentNameException](#) (string? name)
- [AuctionSiteInexistentNameException](#) (string? name, string? message)
- [AuctionSiteInexistentNameException](#) (string? name, string? message, Exception inner)
- [AuctionSiteInexistentNameException](#) (SerializationInfo info, StreamingContext context)

Properties

- string? [Name](#) [get]

Additional Inherited Members

6.6.1 Detailed Description

Thrown to notify the attempt use its name to access an entity which is not available on the database.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 AuctionSiteInexistentNameException() [1/4]

```
AuctionSiteInexistentNameException (
    string? name )
```

6.6.2.2 AuctionSiteInexistentNameException() [2/4]

```
AuctionSiteInexistentNameException (
    string? name,
    string? message )
```

6.6.2.3 AuctionSiteInexistentNameException() [3/4]

```
AuctionSiteInexistentNameException (
    string? name,
    string? message,
    Exception inner )
```

6.6.2.4 AuctionSiteInexistentNameException() [4/4]

```
AuctionSiteInexistentNameException (
    SerializationInfo info,
    StreamingContext context )
```

6.6.3 Property Documentation

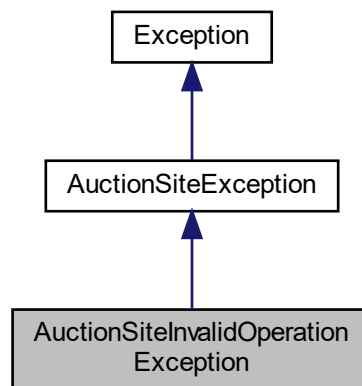
6.6.3.1 Name

```
string? Name [get]
```

6.7 AuctionSiteInvalidOperationException Class Reference

Thrown to notify an attempt at invoking an operation on an object in an invalid state (for instance a deleted entity)

Inheritance diagram for AuctionSiteInvalidOperationException:



Public Member Functions

- [AuctionSiteInvalidOperationException](#) ()
- [AuctionSiteInvalidOperationException](#) (string? message)
- [AuctionSiteInvalidOperationException](#) (string? message, Exception inner)

Protected Member Functions

- [AuctionSiteInvalidOperationException](#) (SerializationInfo info, StreamingContext context)

6.7.1 Detailed Description

Thrown to notify an attempt at invoking an operation on an object in an invalid state (for instance a deleted entity)

6.7.2 Constructor & Destructor Documentation

6.7.2.1 AuctionSiteInvalidOperationException() [1/4]

```
AuctionSiteInvalidOperationException ( )
```

6.7.2.2 AuctionSiteInvalidOperationException() [2/4]

```
AuctionSiteInvalidOperationException (
    string? message )
```

6.7.2.3 AuctionSiteInvalidOperationException() [3/4]

```
AuctionSiteInvalidOperationException (
    string? message,
    Exception inner )
```

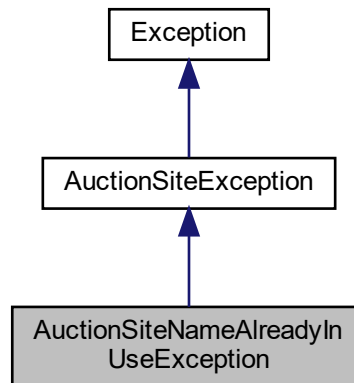
6.7.2.4 AuctionSiteInvalidOperationException() [4/4]

```
AuctionSiteInvalidOperationException (
    SerializationInfo info,
    StreamingContext context ) [protected]
```

6.8 AuctionSiteNameAlreadyInUseException Class Reference

Thrown to notify the attempt to create an entity whose name is already in use. For instance, a site with the same name of another site managed by the same host, or a user with the same name of another user on the same site.

Inheritance diagram for AuctionSiteNameAlreadyInUseException:



Public Member Functions

- [AuctionSiteNameAlreadyInUseException](#) (string? name)
- [AuctionSiteNameAlreadyInUseException](#) (string? name, string? message)
- [AuctionSiteNameAlreadyInUseException](#) (string? name, string? message, Exception inner)
- [AuctionSiteNameAlreadyInUseException](#) (SerializationInfo info, StreamingContext context)

Properties

- string? [Name](#) [get]

Additional Inherited Members

6.8.1 Detailed Description

Thrown to notify the attempt to create an entity whose name is already in use. For instance, a site with the same name of another site managed by the same host, or a user with the same name of another user on the same site.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 AuctionSiteNameAlreadyInUseException() [1/4]

```
AuctionSiteNameAlreadyInUseException (
    string? name )
```

6.8.2.2 AuctionSiteNameAlreadyInUseException() [2/4]

```
AuctionSiteNameAlreadyInUseException (
    string? name,
    string? message )
```

6.8.2.3 AuctionSiteNameAlreadyInUseException() [3/4]

```
AuctionSiteNameAlreadyInUseException (
    string? name,
    string? message,
    Exception inner )
```

6.8.2.4 AuctionSiteNameAlreadyInUseException() [4/4]

```
AuctionSiteNameAlreadyInUseException (
    SerializationInfo info,
    StreamingContext context )
```

6.8.3 Property Documentation

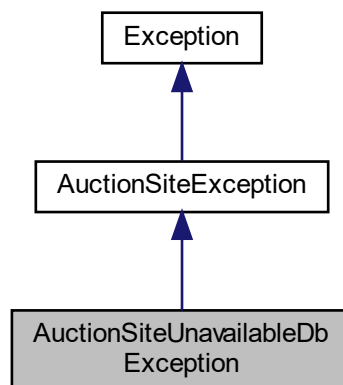
6.8.3.1 Name

```
string? Name [get]
```

6.9 AuctionSiteUnavailableDbException Class Reference

Thrown to notify that no connection with the database has been established.

Inheritance diagram for AuctionSiteUnavailableDbException:



Public Member Functions

- [AuctionSiteUnavailableDbException](#) ()
- [AuctionSiteUnavailableDbException](#) (string? message)
- [AuctionSiteUnavailableDbException](#) (string? message, Exception inner)
- [AuctionSiteUnavailableDbException](#) (SerializationInfo info, StreamingContext context)

Additional Inherited Members

6.9.1 Detailed Description

Thrown to notify that no connection with the database has been established.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 AuctionSiteUnavailableDbException() [1/4]

[AuctionSiteUnavailableDbException](#) ()

6.9.2.2 AuctionSiteUnavailableDbException() [2/4]

```
AuctionSiteUnavailableDbException (
    string? message )
```

6.9.2.3 AuctionSiteUnavailableDbException() [3/4]

```
AuctionSiteUnavailableDbException (
    string? message,
    Exception inner )
```

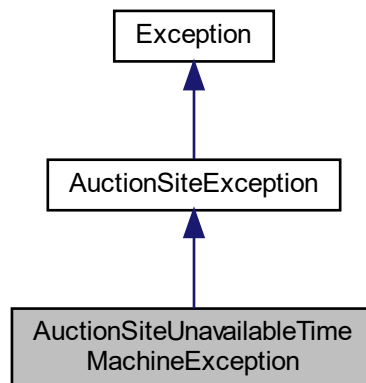
6.9.2.4 AuctionSiteUnavailableDbException() [4/4]

```
AuctionSiteUnavailableDbException (
    SerializationInfo info,
    StreamingContext context )
```

6.10 AuctionSiteUnavailableTimeMachineException Class Reference

Thrown to notify the attempt to create an auction with expiration date in the past.

Inheritance diagram for AuctionSiteUnavailableTimeMachineException:



Public Member Functions

- [AuctionSiteUnavailableTimeMachineException](#) ()
- [AuctionSiteUnavailableTimeMachineException](#) (string? message)
- [AuctionSiteUnavailableTimeMachineException](#) (string? message, Exception inner)
- [AuctionSiteUnavailableTimeMachineException](#) (SerializationInfo info, StreamingContext context)

Additional Inherited Members

6.10.1 Detailed Description

Thrown to notify the attempt to create an auction with expiration date in the past.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 [AuctionSiteUnavailableTimeMachineException](#)() [1/4]

```
AuctionSiteUnavailableTimeMachineException ( )
```

6.10.2.2 [AuctionSiteUnavailableTimeMachineException](#)() [2/4]

```
AuctionSiteUnavailableTimeMachineException (  
    string? message )
```

6.10.2.3 [AuctionSiteUnavailableTimeMachineException](#)() [3/4]

```
AuctionSiteUnavailableTimeMachineException (  
    string? message,  
    Exception inner )
```

6.10.2.4 [AuctionSiteUnavailableTimeMachineException](#)() [4/4]

```
AuctionSiteUnavailableTimeMachineException (  
    SerializationInfo info,  
    StreamingContext context )
```

6.11 DomainConstraints Class Reference

Numeric constants used to express constraints on names, passwords, and time zone

Static Public Attributes

- const int [MinSiteName](#) = 1
The minimal length of a string to be a well formed site name
- const int [MaxSiteName](#) = 128
The maximal length of a string to be a well formed site name
- const int [MinUserName](#) = 3
The minimal length of a string to be a well formed user name
- const int [MaxUserName](#) = 64
The maximal length of a string to be a well formed user name
- const int [MinUserPassword](#) = 4
The minimal length of a string to be an acceptable password
- const int [MinTimeZone](#) = -12
The minimal value acceptable as a time zone
- const int [MaxTimeZone](#) = 12
The maximal value acceptable as a time zone

6.11.1 Detailed Description

Numeric constants used to express constraints on names, passwords, and time zone

6.11.2 Member Data Documentation

6.11.2.1 MaxSiteName

```
const int MaxSiteName = 128 [static]
```

The maximal length of a string to be a well formed site name

6.11.2.2 MaxTimeZone

```
const int MaxTimeZone = 12 [static]
```

The maximal value acceptable as a time zone

6.11.2.3 MaxUserName

```
const int MaxUserName = 64 [static]
```

The maximal length of a string to be a well formed user name

6.11.2.4 MinSiteName

```
const int MinSiteName = 1 [static]
```

The minimal length of a string to be a well formed site name

6.11.2.5 MinTimeZone

```
const int MinTimeZone = -12 [static]
```

The minimal value acceptable as a time zone

6.11.2.6 MinUserName

```
const int MinUserName = 3 [static]
```

The minimal length of a string to be a well formed user name

6.11.2.7 MinUserPassword

```
const int MinUserPassword = 4 [static]
```

The minimal length of a string to be an acceptable password

6.12 IAuction Interface Reference

The auctions managed by sites. Equals on auctions must be true iff the two auctions belong to the same site and have the same Id.

Public Member Functions

- `IUser? CurrentWinner ()`
Returns the user, if any, who has submitted the highest bid so far. In case no bids have been offered yet, it returns null. It may also return null in case of closed auction whose winner has been deleted from the site (after the auction ended).
- `double CurrentPrice ()`
Returns the current price, which is the lowest amount needed to best the second highest bid if two or more bids have been offered; otherwise, it coincides with the starting price.
- `void Delete ()`
Disposes of the auction and all associated resources, if any.
- `bool Bid (ISession session, double offer)`
Makes a bid for this auction on behalf of the session owner; only possible for still open auctions.

Properties

- `int Id [get]`
Gets the unique key used to identify the auctions.
- `IUser Seller [get]`
Gets the user who is selling the object/service.
- `string Description [get]`
Gets the description of the offered object/service.
- `DateTime EndsOn [get]`
Gets the expiring time of the auction; no bid will be accepted after it.

6.12.1 Detailed Description

The auctions managed by sites. Equals on auctions must be true iff the two auctions belong to the same site and have the same Id.

6.12.2 Member Function Documentation

6.12.2.1 Bid()

```
bool Bid (  
    ISession session,  
    double offer )
```

Makes a bid for this auction on behalf of the session owner; only possible for still open auctions.

Parameters

<i>session</i>	A valid session. The expiration time of the session is reset to the expiration time of the site for each bid using correct parameters, that is, not throwing an exception, disregarding whether the bid is accepted or not.
<i>offer</i>	The amount offered as bid, that is, the maximum amount the user is willing to pay for the item. This maximum amount must remain confidential.

Returns

True iff the bid is accepted, so that the status of the auction is changed. The bid is rejected, so that the result is false, iff either of the following occurs

- the bidder is (already) the current winner and *offer* is lower than the maximum offer increased by `minimumBidIncrement`
- the bidder is not the current winner and *offer* is lower than the current price
- the bidder is not the current winner and *offer* is lower than the current price increased by `minimumBidIncrement` AND this is not the first bid

In all other cases, the bid is accepted, the result is true, and the status of the auction is changed as follows:

- if this is the first bid, then the maximum offer is set to *offer*, the current price is not changed (that is, it remains the starting price), and the bidder becomes the current winner;
- if the bidder was already winning this auction, the maximum offer is set to *offer*, current price and current winner are unchanged;
- if this is NOT the first bid, the bidder is NOT the current winner, and *offer* is higher than the current maximum offer, in the following denoted by CMO, then the current price is set to the minimum between *offer* and `CMO+minimumBidIncrement`, the maximum offer is set to *offer*, and the bidder becomes the current winner;
- if this is NOT the first bid, the bidder is NOT the current winner, and *offer* is NOT higher than the current maximum offer, in the following denoted by CMO, then the current price is set to the minimum between CMO and *offer* + `minimumBidIncrement`, and the current winner does not change.

Exceptions

<i>AuctionSiteInvalidOperationException</i>	The auction is already closed (or the corresponding permanent object does not exist anymore).
<i>AuctionSiteArgumentOutOfRangeException</i>	If <i>offer</i> is negative.
<i>AuctionSiteArgumentNullException</i>	If <i>session</i> is null.
<i>AuctionSiteArgumentException</i>	If <i>session</i> is not null and, one of the following conditions is true: <ul style="list-style-type: none"> • the session is not valid anymore; • the logged user is also the Seller of this auction; • the logged user is a user of a site different from the site of the Seller.

6.12.2.2 CurrentPrice()

```
double CurrentPrice ( )
```

Returns the current price, which is the lowest amount needed to best the second highest bid if two or more bids have been offered; otherwise, it coincides with the starting price.

6.12.2.3 CurrentWinner()

```
IUser? CurrentWinner ( )
```

Returns the user, if any, who has submitted the highest bid so far. In case no bids have been offered yet, it returns null. It may also return null in case of closed auction whose winner has been deleted from the site (after the auction ended).

6.12.2.4 Delete()

```
void Delete ( )
```

Disposes of the auction and all associated resources, if any.

6.12.3 Property Documentation

6.12.3.1 Description

```
string Description [get]
```

Gets the description of the offered object/service.

6.12.3.2 EndsOn

```
DateTime EndsOn [get]
```

Gets the expiring time of the auction; no bid will be accepted after it.

6.12.3.3 Id

```
int Id [get]
```

Gets the unique key used to identify the auctions.

6.12.3.4 Seller

```
IUser Seller [get]
```

Gets the user who is selling the object/service.

6.13 IHost Interface Reference

The hosting service A Host own a database and uses it to save the data of the sites it manages. Any Host may create and load its sites.

Public Member Functions

- `IEnumerable<(string Name, int TimeZone)> GetSiteInfos ()`
Yields the names and corresponding time zones of managed sites.
- `void CreateSite (string name, int timezone, int sessionExpirationTimeInSeconds, double minimumBidIncrement)`
Create a new site, identified by its name.
- `ISite LoadSite (string name)`
Yields the [ISite](#) object corresponding to an existing Site.

6.13.1 Detailed Description

The hosting service A Host own a database and uses it to save the data of the sites it manages. Any Host may create and load its sites.

6.13.2 Member Function Documentation

6.13.2.1 CreateSite()

```
void CreateSite (
    string name,
    int timezone,
    int sessionExpirationTimeInSeconds,
    double minimumBidIncrement )
```

Create a new site, identified by its name.

Parameters

<i>name</i>	The name of the site (a unique identifier, whose length is between <code>DomainConstraints.MinSiteName</code> and <code>DomainConstraints.MaxSiteName</code> chars).
<i>timezone</i>	The timezone, an integer between <code>DomainConstraints.MinTimeZone</code> and <code>DomainConstraints.MaxTimeZone</code> (inclusive).
<i>sessionExpirationTimeInSeconds</i>	Session timeout, in seconds, a positive number.
<i>minimumBidIncrement</i>	Minimum bid increment, a positive number.

Exceptions

<i>AuctionSiteNameAlreadyInUseException</i>	Thrown if the name of the site is already in use as name of an existing site.
<i>AuctionSiteArgumentNullException</i>	If <i>name</i> is null.
<i>AuctionSiteArgumentException</i>	If <i>name</i> is not null, but its length is (strictly) smaller than <code>DomainConstraints.MinSiteName</code> or (strictly) larger than <code>DomainConstraints.MaxSiteName</code> .
<i>AuctionSiteArgumentOutOfRangeException</i>	If <i>timezone</i> is (strictly) smaller than <code>DomainConstraints.MinTimeZone</code> or (strictly) larger than <code>DomainConstraints.MaxTimeZone</code> or if <i>sessionExpirationTimeInSeconds</i> or <i>minimumBidIncrement</i> are not positive.
<i>AuctionSiteUnavailableDbException</i>	The DB server is not responding or returns an unexpected error.

6.13.2.2 GetSiteInfos()

```
IEnumerable<(string Name, int TimeZone)> GetSiteInfos ( )
```

Yields the names and corresponding time zones of managed sites.

Returns

The names of the managed sites and their time zones.

Exceptions

<i>AuctionSiteUnavailableDbException</i>	If the DB server is not responding or returns an unexpected error.
--	--

6.13.2.3 LoadSite()

```
ISite LoadSite (
    string name )
```

Yields the [ISite](#) object corresponding to an existing Site.

Parameters

<i>name</i>	The name of the site.
-------------	-----------------------

Returns

A new instance for the site.

Exceptions

<i>AuctionSiteUnavailableDbException</i>	The DB server is not responding or returns an unexpected error.
<i>AuctionSiteArgumentNullException</i>	If <i>name</i> is null.
<i>AuctionSiteArgumentException</i>	If <i>name</i> is not null, but its length is (strictly) smaller than <code>DomainConstraints.MinSiteName</code> or (strictly) larger than <code>DomainConstraints.MaxSiteName</code> .
<i>AuctionSiteInexistentNameException</i>	If <i>name</i> is a non-null string of the correct length, but the corresponding site is not present in the DB.

6.14 IHostFactory Interface Reference

This component factory. All its methods have a parameter `connectionString` representing the connection string of a Microsoft SQL Server DB, used to permanently store the data of the auction sites managed by a specific Host and to implicitly characterize the Host. Many different objects of type [IHost](#) may at the same time represent the same Host, concurrently working on the same database.

Public Member Functions

- void [CreateHost](#) (string `connectionString`)
Creates a new DB (dropping existing previous version, if any), and initialize it with all the necessary DB elements for the Host.
- [IHost LoadHost](#) (string `connectionString`, `IArmClockFactory alarmClockFactory`)
Yields the Host managing a group of Sites having their data resident on the same database.

6.14.1 Detailed Description

This component factory. All its methods have a parameter `connectionString` representing the connection string of a Microsoft SQL Server DB, used to permanently store the data of the auction sites managed by a specific Host and to implicitly characterize the Host. Many different objects of type [IHost](#) may at the same time represent the same Host, concurrently working on the same database.

6.14.2 Member Function Documentation

6.14.2.1 CreateHost()

```
void CreateHost (
    string connectionString )
```

Creates a new DB (dropping existing previous version, if any), and initialize it with all the necessary DB elements for the Host.

Parameters

<i>connectionString</i>	A valid connection string for a Microsoft SQL Server DB.
-------------------------	--

Exceptions

<i>AuctionSiteArgumentNullException</i>	If <i>connectionString</i> is null.
<i>AuctionSiteUnavailableDbException</i>	If <i>connectionString</i> is (non-null but) malformed, the DB server is not responding or returns an unexpected error.

6.14.2.2 LoadHost()

```
IHost LoadHost (
    string connectionString,
    IAlarmClockFactory alarmClockFactory )
```

Yields the Host managing a group of Sites having their data resident on the same database.

Parameters

<i>connectionString</i>	The connection string.
<i>alarmClockFactory</i>	The alarm clock factory.

Returns

A new instance for the host based on that database.

Exceptions

<i>AuctionSiteUnavailableDbException</i>	The connection string is (non-null but) malformed, the DB server is not responding or returns an unexpected error.
<i>AuctionSiteArgumentNullException</i>	If <i>connectionString</i> or <i>alarmClockFactory</i> are null.

6.15 ISession Interface Reference

The sessions of the users on the site. It may become invalid (expire) if the user is idle for the expiration time set for the site. Equals on sessions must be true iff the two sessions have the same Id.

Public Member Functions

- void [Logout](#) ()
Deletes the session and disposes of all associated resources, if any.
- [IAuction CreateAuction](#) (string description, DateTime endsOn, double startingPrice)
Yields an auction for the described object/service. As a side effect, the expiration time of the session is reset (to the same value as if the session was newly created).

Properties

- string `Id` [get]
Gets the unique key used to identify the sessions.
- DateTime `ValidUntil` [get]
Gets the current expiration time of the session.
- IUser `User` [get]
Gets the user owner of the session.

6.15.1 Detailed Description

The sessions of the users on the site. It may become invalid (expire) if the user is idle for the expiration time set for the site. Equals on sessions must be true iff the two sessions have the same Id.

6.15.2 Member Function Documentation

6.15.2.1 CreateAuction()

```
IAuction CreateAuction (
    string description,
    DateTime endsOn,
    double startingPrice )
```

Yields an auction for the described object/service. As a side effect, the expiration time of the session is reset (to the same value as if the session was newly created).

Parameters

<i>description</i>	The description of the object/service for sale, a non-null and non-empty string.
<i>endsOn</i>	The expiring time of the auction; no bid will be accepted after it.
<i>startingPrice</i>	The starting price of the auction, a non-negative number; the first bid must be greater than or equal to this value.

Returns

Returns the newly created auction, whose Id is an automatically-generated unique identifier.

Exceptions

<i>AuctionSiteInvalidOperationException</i>	The session is not valid (or the corresponding permanent object does not exist anymore).
<i>AuctionSiteArgumentNullException</i>	If <i>description</i> is null.
<i>AuctionSiteArgumentException</i>	If <i>description</i> is not null but empty.
<i>AuctionSiteArgumentOutOfRangeException</i>	If <i>startingPrice</i> is negative
<i>AuctionSiteUnavailableTimeMachineException</i>	If <i>endsOn</i> precedes the current time (according to the IAlarmClock of the <i>ISite</i>)

6.15.2.2 Logout()

```
void Logout ( )
```

Deletes the session and disposes of all associated resources, if any.

6.15.3 Property Documentation

6.15.3.1 Id

```
string Id [get]
```

Gets the unique key used to identify the sessions.

6.15.3.2 User

```
IUser User [get]
```

Gets the user owner of the session.

6.15.3.3 ValidUntil

```
DateTime ValidUntil [get]
```

Gets the current expiration time of the session.

6.16 ISite Interface Reference

The auction sites. Equals on sites must be true iff the two sites have the same Name.

Public Member Functions

- `IEnumerable< IUser > ToyGetUsers ()`
Yields all the users of the site. In a realistic example, this method would be more complex, using some sort of pagination
- `IEnumerable< ISession > ToyGetSessions ()`
Yields all the sessions of the site. In a realistic example, this method would be more complex, using some sort of pagination
- `IEnumerable< IAuction > ToyGetAuctions (bool onlyNotEnded)`
Yields all the (not yet ended) auctions of the site. In a realistic example, this method would be more complex, using some sort of pagination
- `ISession? Login (string username, string password)`
Yields the session for the user, new iff no valid session for him/her exists. No user can have two valid sessions on the same site at the same time.
- `void CreateUser (string username, string password)`
Add a user of the site.
- `void Delete ()`
Disposes of the site and all its associated resources.
- `DateTime Now ()`
Returns the current time

Properties

- `string Name [get]`
The name of the auction site.
- `int Timezone [get]`
The timezone of the auction site.
- `int SessionExpirationInSeconds [get]`
The number of seconds needed for the session of an idle user to time out. A positive number
- `double MinimumBidIncrement [get]`
The minimum amount allowed as increment (from the starting price) for a bid. A positive number

6.16.1 Detailed Description

The auction sites. Equals on sites must be true iff the two sites have the same Name.

6.16.2 Member Function Documentation

6.16.2.1 CreateUser()

```
void CreateUser (
    string username,
    string password )
```

Add a user of the site.

Parameters

<i>username</i>	User login (minimum length=DomainConstraints.MinUserName, maximum=DomainConstraints.MaxUserName chars).
<i>password</i>	User password (minimum length=DomainConstraints.MinUserPassword chars).

Exceptions

<i>AuctionSiteNameAlreadyInUseException</i>	If <i>username</i> is already in use for a user of the site.
<i>AuctionSiteArgumentNullException</i>	If <i>username</i> or <i>password</i> are null.
<i>AuctionSiteArgumentException</i>	If <i>username</i> is not null, but its length is (strictly) smaller than DomainConstraints.MinUserName or (strictly) larger than DomainConstraints.MaxUserName, or if <i>password</i> is not null, but its length is (strictly) smaller than DomainConstraints.MinUserPassword.

6.16.2.2 Delete()

```
void Delete ( )
```

Disposes of the site and all its associated resources.

6.16.2.3 Login()

```
ISession? Login (
    string username,
    string password )
```

Yields the session for the user, new iff no valid session for him/her exists. No user can have two valid sessions on the same site at the same time.

Parameters

<i>username</i>	User login.
<i>password</i>	User password.

Returns

The session for the user or null if *username* and *password* do not correspond to a user of the site.

Exceptions

<i>AuctionSiteArgumentNullException</i>	If <i>username</i> or <i>password</i> are null.
---	---

Exceptions

AuctionSiteArgumentException	If <i>username</i> is not null, but its length is (strictly) smaller than <code>DomainConstraints.MinUserName</code> or (strictly) larger than <code>DomainConstraints.MaxUserName</code> , or if <i>password</i> is not null, but its length is (strictly) smaller than <code>DomainConstraints.MinUserPassword</code> .
--	---

6.16.2.4 Now()

```
DateTime Now ( )
```

Returns the current time

Returns

The current time, as provided by the internal `IArmClock`

6.16.2.5 ToyGetAuctions()

```
IEnumerable< IAuction > ToyGetAuctions (
    bool onlyNotEnded )
```

Yields all the (not yet ended) auctions of the site. In a realistic example, this method would be more complex, using some sort of pagination

Parameters

<i>onlyNotEnded</i>	If true, only the auctions not yet ended are taken into account.
---------------------	--

Returns

If not *onlyNotEnded* , all the auctions, otherwise only those not yet ended.

6.16.2.6 ToyGetSessions()

```
IEnumerable< ISession > ToyGetSessions ( )
```

Yields all the sessions of the site. In a realistic example, this method would be more complex, using some sort of pagination

Returns

All the sessions of the site.

6.16.2.7 ToyGetUsers()

```
IEnumerable< IUser > ToyGetUsers ( )
```

Yields all the users of the site. In a realistic example, this method would be more complex, using some sort of pagination

Returns

All the users of the site.

6.16.3 Property Documentation

6.16.3.1 MinimumBidIncrement

```
double MinimumBidIncrement [get]
```

The minimum amount allowed as increment (from the starting price) for a bid. A positive number

6.16.3.2 Name

```
string Name [get]
```

The name of the auction site.

6.16.3.3 SessionExpirationInSeconds

```
int SessionExpirationInSeconds [get]
```

The number of seconds needed for the session of an idle user to time out. A positive number

6.16.3.4 Timezone

```
int Timezone [get]
```

The timezone of the auction site.

6.17 IUser Interface Reference

The users of the auction system. Equals on users must be true iff the two users belong to the same site and have the same Username.

Public Member Functions

- `IEnumerable< IAuction > WonAuctions ()`

Yields the auctions won by the user.

- `void Delete ()`

Disposes of the user and all its resources. Users cannot be deleted if they are owners or (current) winners of still open auctions. Thus, a call to Delete on a user who is owner or (current) winner of auctions yet to be adjudicated must throw [AuctionSiteInvalidOperationException](#) Ended owned auctions are disposed of, if any. Ended won auctions are updated, and the information of the winner is removed, if any.

Properties

- `string Username [get]`

Gets the unique key used to identify the user of a specific site in the system. It is a string of at least `DomainConstraints.MinUserName`, and at most `DomainConstraints.MaxUserName` characters. The same username may be used by different users on different sites.

6.17.1 Detailed Description

The users of the auction system. Equals on users must be true iff the two users belong to the same site and have the same Username.

6.17.2 Member Function Documentation

6.17.2.1 Delete()

```
void Delete ( )
```

Disposes of the user and all its resources. Users cannot be deleted if they are owners or (current) winners of still open auctions. Thus, a call to Delete on a user who is owner or (current) winner of auctions yet to be adjudicated must throw [AuctionSiteInvalidOperationException](#) Ended owned auctions are disposed of, if any. Ended won auctions are updated, and the information of the winner is removed, if any.

6.17.2.2 WonAuctions()

```
IEnumerable< IAuction > WonAuctions ( )
```

Yields the auctions won by the user.

Returns

The auctions won by the user (that is, all the ended auctions of this site where this user is the highest bidder).

6.17.3 Property Documentation

6.17.3.1 Username

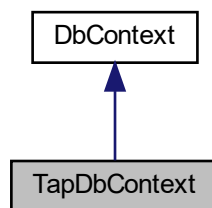
```
string Username [get]
```

Gets the unique key used to identify the user of a specific site in the system. It is a string of at least `DomainConstraints.MinUserName`, and at most `DomainConstraints.MaxUserName` characters. The same username may be used by different users on different sites.

6.18 TapDbContext Class Reference

To be extended by the component DbContext, to simplify advanced testing

Inheritance diagram for TapDbContext:



Protected Member Functions

- [TapDbContext](#) ()
- [TapDbContext](#) (DbContextOptions options)

Properties

- static int [NumberOfCreatedContexts](#) [get, set]
incremented by any constructor call

6.18.1 Detailed Description

To be extended by the component DbContext, to simplify advanced testing

6.18.2 Constructor & Destructor Documentation

6.18.2.1 TapDbContext() [1/2]

```
TapDbContext ( ) [protected]
```

6.18.2.2 TapDbContext() [2/2]

```
TapDbContext (
    DbContextOptions options ) [protected]
```

6.18.3 Property Documentation

6.18.3.1 NumberOfCreatedContexts

```
int NumberOfCreatedContexts [static], [get], [set]
```

incremented by any constructor call

Index

ActualValue
 AuctionSiteArgumentOutOfRangeException, 17
AuctionSiteArgumentException, 11
 AuctionSiteArgumentException, 12, 13
 ParamName, 13
AuctionSiteArgumentNullException, 13
 AuctionSiteArgumentNullException, 14
AuctionSiteArgumentOutOfRangeException, 15
 ActualValue, 17
 AuctionSiteArgumentOutOfRangeException, 16
AuctionSiteConcurrentChangeException, 17
 AuctionSiteConcurrentChangeException, 18
AuctionSiteException, 19
 AuctionSiteException, 19, 20
AuctionSiteInexistentNameException, 20
 AuctionSiteInexistentNameException, 21
 Name, 22
AuctionSiteInvalidOperationException, 22
 AuctionSiteInvalidOperationException, 23
AuctionSiteNameAlreadyInUseException, 24
 AuctionSiteNameAlreadyInUseException, 24, 25
 Name, 25
AuctionSiteUnavailableDbException, 26
 AuctionSiteUnavailableDbException, 26, 27
AuctionSiteUnavailableTimeMachineException, 27
 AuctionSiteUnavailableTimeMachineException, 28

Bid
 IAuction, 31

CreateAuction
 ISession, 38
CreateHost
 IHostFactory, 36
CreateSite
 IHost, 34
CreateUser
 ISite, 40
CurrentPrice
 IAuction, 32
CurrentWinner
 IAuction, 32

Delete
 IAuction, 33
 ISite, 41
 IUser, 44
Description
 IAuction, 33
DomainConstraints, 29

MaxSiteName, 29
MaxTimeZone, 29
MaxUserName, 29
MinSiteName, 30
MinTimeZone, 30
MinUserName, 30
MinUserPassword, 30

EndsOn
 IAuction, 33

GetSiteInfos
 IHost, 35

IAuction, 30
 Bid, 31
 CurrentPrice, 32
 CurrentWinner, 32
 Delete, 33
 Description, 33
 EndsOn, 33
 Id, 33
 Seller, 33

Id
 IAuction, 33
 ISession, 39
IHost, 34
 CreateSite, 34
 GetSiteInfos, 35
 LoadSite, 35
IHostFactory, 36
 CreateHost, 36
 LoadHost, 37
ISession, 37
 CreateAuction, 38
 Id, 39
 Logout, 39
 User, 39
 ValidUntil, 39
ISite, 39
 CreateUser, 40
 Delete, 41
 Login, 41
 MinimumBidIncrement, 43
 Name, 43
 Now, 42
 SessionExpirationInSeconds, 43
 Timezone, 43
 ToyGetAuctions, 42
 ToyGetSessions, 42

- ToyGetUsers, [42](#)
- IUser, [44](#)
 - Delete, [44](#)
 - Username, [45](#)
 - WonAuctions, [44](#)
- LoadHost
 - IHostFactory, [37](#)
- LoadSite
 - IHost, [35](#)
- Login
 - ISite, [41](#)
- Logout
 - ISession, [39](#)
- MaxSiteName
 - DomainConstraints, [29](#)
- MaxTimeZone
 - DomainConstraints, [29](#)
- MaxUserName
 - DomainConstraints, [29](#)
- MinimumBidIncrement
 - ISite, [43](#)
- MinSiteName
 - DomainConstraints, [30](#)
- MinTimeZone
 - DomainConstraints, [30](#)
- MinUserName
 - DomainConstraints, [30](#)
- MinUserPassword
 - DomainConstraints, [30](#)
- Name
 - AuctionSiteInexistentNameException, [22](#)
 - AuctionSiteNameAlreadyInUseException, [25](#)
 - ISite, [43](#)
- Now
 - ISite, [42](#)
- NumberOfCreatedContexts
 - TapDbContext, [46](#)
- ParamName
 - AuctionSiteArgumentException, [13](#)
- Seller
 - IAuction, [33](#)
- SessionExpirationInSeconds
 - ISite, [43](#)
- TAP21_22_AuctionSite, [9](#)
- TAP21_22_AuctionSite.Interface, [9](#)
- TapDbContext, [45](#)
 - NumberOfCreatedContexts, [46](#)
 - TapDbContext, [46](#)
- Timezone
 - ISite, [43](#)
- ToyGetAuctions
 - ISite, [42](#)
- ToyGetSessions
 - ISite, [42](#)
- ToyGetUsers
 - ISite, [42](#)
- User
 - ISession, [39](#)
- Username
 - IUser, [45](#)
- ValidUntil
 - ISession, [39](#)
- WonAuctions
 - IUser, [44](#)