Namespace Solstice.Domain.Exceptions

Classes

[CoreException](#)

Represents an exception that is thrown when there is an error.

[CoreExceptionEnumExtension](#)

Provides extension methods for the [CoreExceptionEnum](#) enumeration.

Enums

[CoreExceptionEnum](#)

Represents the possible exceptions that can be thrown by the Solstice.Domain classes.

Class CoreException

Namespace: [Solstice](#).[Domain](#).[Exceptions](#)

Assembly: Solstice.Domain.dll

Represents an exception that is thrown when there is an error.

```
public class CoreException : Exception, ISerializable
```

Inheritance

[object](#) ← [Exception](#) ← CoreException

Implements

[ISerializable](#)

Inherited Members

[Exception.GetBaseException()](#) , [Exception.GetType()](#) , [Exception.ToString()](#) ,
[Exception.Data](#) , [Exception.HelpLink](#) , [Exception.HResult](#) , [Exception.InnerException](#) ,
[Exception.Message](#) , [Exception.Source](#) , [Exception.StackTrace](#) , [Exception.TargetSite](#) ,
[Exception.SerializeObjectState](#) , [object.Equals(object)](#) , [object.Equals(object, object)](#) ,
[object.GetHashCode()](#) , [object.MemberwiseClone()](#) ,
[object.ReferenceEquals(object, object)](#)

Extension Methods

[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)](#)
,
[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)](#) ,
[MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)](#) ,
[MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)](#)

Constructors

CoreException(int, string)

```
public CoreException(int statusCode, string message)
```

Parameters

statusCode `int`⧉

message `string`⧉

CoreException(int, string, Exception)

```
public CoreException(int statusCode, string message, Exception innerException)
```

Parameters

statusCode `int`⧉

message `string`⧉

innerException `Exception`⧉

CoreException(string)

```
public CoreException(string message)
```

Parameters

message `string`⧉

CoreException(string, Exception)

```
public CoreException(string message, Exception innerException)
```

Parameters

message `string`⧉

innerException `Exception`⧉

Properties

StatusCode

```csharp
public int StatusCode { get; }
```

Property Value

[int](#)

Methods

Error400(params object[])

Generate a 400 error with or without a message.

```csharp
public static CoreException Error400(params object[] values)
```

Parameters

values [object](#)[]

Returns

[CoreException](#)

Error401(params object[])

Generate a 401 error with or without a message.

```csharp
public static CoreException Error401(params object[] values)
```

Parameters

values [object](#)[]

Returns

[CoreException](#)

Error403(params object[])

Generate a 403 error with or without a message.

```
public static CoreException Error403(params object[] values)
```

Parameters

values object[]

Returns

CoreException

Error404(params object[])

Generate a 404 error with or without a message.

```
public static CoreException Error404(params object[] values)
```

Parameters

values object[]

Returns

CoreException

Error500(params object[])

Generate a 500 error with or without a message.

```
public static CoreException Error500(params object[] values)
```

Parameters

values object[]

Returns

[CoreException](#)

Format(CoreExceptionEnum)

Used to generate exception message based on the provided [CoreExceptionEnum](#).

```
public static CoreException Format(CoreExceptionEnum coreExceptionEnum)
```

Parameters

coreExceptionEnum [CoreExceptionEnum](#)

Returns

[CoreException](#)

Format(CoreExceptionEnum, params object[])

```
public static CoreException Format(CoreExceptionEnum coreExceptionEnum, params
object[] values)
```

Parameters

coreExceptionEnum [CoreExceptionEnum](#)

values object[]

Returns

[CoreException](#)

Format(Exception, CoreExceptionEnum, params object[])

```
public static CoreException Format(Exception ex, CoreExceptionEnum
coreExceptionEnum, params object[] values)
```

Parameters

ex [Exception](⧉)

coreExceptionEnum [CoreExceptionEnum](link)

values [object](⧉)[]

Returns

[CoreException](link)

Format(Exception, int, CoreExceptionEnum, params object[])

```
public static CoreException Format(Exception ex, int statusCode, CoreExceptionEnum
coreExceptionEnum, params object[] values)
```

Parameters

ex [Exception](⧉)

statusCode [int](⧉)

coreExceptionEnum [CoreExceptionEnum](link)

values [object](⧉)[]

Returns

[CoreException](link)

Format(int, CoreExceptionEnum, params object[])

```
public static CoreException Format(int statusCode, CoreExceptionEnum
coreExceptionEnum, params object[] values)
```

Parameters

`statusCode` [int](#)

`coreExceptionEnum` [CoreExceptionEnum](#)

`values` [object](#)[]

Returns

[CoreException](#)

Enum CoreExceptionEnum

Namespace: Solstice.Domain.Exceptions

Assembly: Solstice.Domain.dll

Represents the possible exceptions that can be thrown by the Solstice.Domain classes.

```
public enum CoreExceptionEnum
```

Extension Methods
CoreExceptionEnumExtension.Get(CoreExceptionEnum)

Fields

HTTP_400 = 3

HTTP_400_WITH_MESSAGE = 4

HTTP_401 = 5

HTTP_401_WITH_MESSAGE = 6

HTTP_403 = 7

HTTP_403_WITH_MESSAGE = 8

HTTP_404 = 9

HTTP_404_WITH_MESSAGE = 10

HTTP_500 = 11

HTTP_500_WITH_MESSAGE = 12

NO_QUERY_DTO = 2

NO_REPOSITORY = 0

NO_SERVICE = 1

Class CoreExceptionEnumExtension

Namespace: [Solstice](#).[Domain](#).[Exceptions](#)

Assembly: Solstice.Domain.dll

Provides extension methods for the [CoreExceptionEnum](#) enumeration.

```
public static class CoreExceptionEnumExtension
```

Inheritance

[object](#) ← CoreExceptionEnumExtension

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Methods

Get(CoreExceptionEnum)

Gets the error message associated with the specified [CoreExceptionEnum](#) value.

```
public static string Get(this CoreExceptionEnum coreExceptionEnum)
```

Parameters

`coreExceptionEnum` [CoreExceptionEnum](#)

The [CoreExceptionEnum](#) value.

Returns

[string](#)

The error message associated with the specified [CoreExceptionEnum](#) value.

Namespace Solstice.Domain.Extensions

Classes

EnumerableExtensions

QueryableExtensions

Class EnumerableExtensions

Namespace: [Solstice](#).[Domain](#).[Extensions](#)

Assembly: Solstice.Domain.dll

```
public static class EnumerableExtensions
```

Inheritance

[object](#) ← EnumerableExtensions

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) ,
[object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) ,
[object.ToString()](#)

Methods

IsNotNullOrEmpty<T>(IEnumerable<T>)

```
public static bool IsNotNullOrEmpty<T>(this IEnumerable<T> list)
```

Parameters

list [IEnumerable](#)<T>

Returns

[bool](#)

Type Parameters

T

IsNullOrEmpty<T>(IEnumerable<T>)

```csharp
public static bool IsNullOrEmpty<T>(this IEnumerable<T> list)
```

Parameters

list [IEnumerable](#)⧉ <T>

Returns

[bool](#)⧉

Type Parameters

T

Class QueryableExtensions

Namespace: [Solstice](#).[Domain](#).[Extensions](#)

Assembly: Solstice.Domain.dll

```
public static class QueryableExtensions
```

Inheritance

[object](#) ← QueryableExtensions

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) ,
[object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) ,
[object.ToString()](#)

Methods

IsNotNullOrEmpty\<T>(IQueryable\<T>)

```
public static bool IsNotNullOrEmpty<T>(this IQueryable<T> list)
```

Parameters

`list` [IQueryable](#)\<T>

Returns

[bool](#)

Type Parameters

`T`

IsNullOrEmpty\<T>(IQueryable\<T>)

```csharp
public static bool IsNullOrEmpty<T>(this IQueryable<T> list)
```

Parameters

list IQueryable<T>

Returns

bool

Type Parameters

T

Namespace Solstice.Domain.Models

Classes

CoreModel

CoreModel class provides a structure to encapsulate information related to Solstice functionality in the application. The information mainly includes id associated to each Solstice transaction or operation.

Page

Represents a page in a Solstice context with a specific page number and size.

PageExtension

Provides extension methods on a Page instance.

[Paged<T>](Paged)

The 'Paged' class provides a simple way to encapsulate a collection of items with a certain count. It's generic, able to support any type of object (indicated by 'T') for pagination purposes.

Class CoreModel

Namespace: [Solstice](#).[Domain](#).[Models](#)

Assembly: Solstice.Domain.dll

CoreModel class provides a structure to encapsulate information related to Solstice functionality in the application. The information mainly includes id associated to each Solstice transaction or operation.

```
public class CoreModel
```

Inheritance

[object](#) ← CoreModel

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Extension Methods

[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)](#)

Properties

Id

Gets or sets the unique identifier for the Solstice operation.

```
public int Id { get; set; }
```

Property Value

[int](#)

Class Page

Namespace: [Solstice](#).[Domain](#).[Models](#)

Assembly: Solstice.Domain.dll

Represents a page in a Solstice context with a specific page number and size.

```
public record Page : IEquatable<Page>
```

Inheritance

[object](#) ← Page

Implements

[IEquatable](#)<[Page](#)>

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) ,
[object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) ,
[object.ToString()](#)

Extension Methods

[PageExtension.LineToSkip(Page)](#) ,
[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)](#)
,
[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)](#) ,
[MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)](#) ,
[MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)](#)

Constructors

Page(int, int)

Represents a page in a Solstice context with a specific page number and size.

```
public Page(int PageNumber = 1, int PageSize = 10)
```

Parameters

PageNumber int⧉

  The number of the page, default is 1.

PageSize int⧉

  The size of the page, default is 10.

Properties

PageNumber

The number of the page, default is 1.

```
public int PageNumber { get; init; }
```

Property Value

int⧉

PageSize

The size of the page, default is 10.

```
public int PageSize { get; init; }
```

Property Value

int⧉

Class PageExtension

Namespace: [Solstice](#).[Domain](#).[Models](#)

Assembly: Solstice.Domain.dll

Provides extension methods on a [Page](#) instance.

```
public static class PageExtension
```

Inheritance

[object](#) ← PageExtension

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) ,
[object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) ,
[object.ToString()](#)

Methods

LineToSkip(Page)

Calculates how many entries need to be skipped for pagination.

```
public static int LineToSkip(this Page page)
```

Parameters

`page` [Page](#)

A [Page](#) instance.

Returns

[int](#)

Returns the number of entries that needs to be skipped to get the correct page context.

Class Paged<T>

Namespace: [Solstice](#).[Domain](#).[Models](#)

Assembly: Solstice.Domain.dll

The 'Paged' class provides a simple way to encapsulate a collection of items with a certain count. It's generic, able to support any type of object (indicated by 'T') for pagination purposes.

```
public class Paged<T>
```

Type Parameters

T

   The type of items held within the 'Paged' object.

Inheritance

[object](#) ← Paged<T>

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Extension Methods

[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)](#)

Remarks

Properties include:

- 'Items': A collection of items of type 'T'. Nullable.

- 'Count': The total count of items present in the 'Items' collection.

Properties

Count

```
public int Count { get; set; }
```

Property Value

[int ↗](#)

Items

```
public ICollection<T>? Items { get; set; }
```

Property Value

[ICollection ↗](#)<T>

Namespace Solstice.Domain.Utils

Classes

Error

MatchExtensions

ResultExtensions

[Result<TSuccess, TError>](Result)

Class Error

Namespace:

```
public record Error : IEquatable<Error>
```

Inheritance

[object](#) ← Error

Implements

[IEquatable](#)<[Error](#)>

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Extension Methods

[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)](#)

Constructors

Error(string, string)

```
public Error(string Code, string Message)
```

Parameters

`Code` [string](#)

`Message` [string](#)

Fields

None

```
public static Error None
```

Field Value

[Error](#)

NullValue

```
public static Error NullValue
```

Field Value

[Error](#)

Properties

AlreadyExistsEmail

```
public static Error AlreadyExistsEmail { get; }
```

Property Value

[Error](#)

AlreadyExistsUsername

```
public static Error AlreadyExistsUsername { get; }
```

Property Value

Error

Code

```
public string Code { get; init; }
```

Property Value

[string](#)

InvalidCredentials

```
public static Error InvalidCredentials { get; }
```

Property Value

[Error](#)

InvalidUser

```
public static Error InvalidUser { get; }
```

Property Value

[Error](#)

Message

```
public string Message { get; init; }
```

Property Value

[string](#)

Class MatchExtensions

```
public static class MatchExtensions
```

Inheritance

[object](#) ← MatchExtensions

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Methods

Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)

```
public static Task<TResult> Some<T, TResult>(this T? value, Func<T, Task<TResult>>
onSome, Func<Task<TResult>> onNone) where T : struct
```

Parameters

`value` T?

`onSome` [Func](#)<T, [Task](#)<TResult>>

`onNone` [Func](#)<[Task](#)<TResult>>

Returns

[Task](#)<TResult>

Type Parameters

`T`

TResult

Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)

```
public static Task<TResult> Some<T, TResult>(this T? value, Func<T, Task<TResult>>
onSome, Func<TResult> onNone) where T : struct
```

Parameters

value T?

onSome Func <T, Task <TResult>>

onNone Func <TResult>

Returns

Task <TResult>

Type Parameters

T

TResult

Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)

```
public static Task<TResult> Some<T, TResult>(this T? value, Func<T, TResult> onSome,
Func<Task<TResult>> onNone) where T : struct
```

Parameters

value T?

onSome Func <T, TResult>

onNone **Func**☑<**Task**☑<TResult>>

Returns

Task☑<TResult>

Type Parameters

T

TResult

Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)

```
public static TResult Some<T, TResult>(this T? value, Func<T, TResult> onSome,
Func<TResult> onNone) where T : struct
```

Parameters

value T?

onSome **Func**☑<T, TResult>

onNone **Func**☑<TResult>

Returns

TResult

Type Parameters

T

TResult

Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)

```
public static Task<TResult> Some<T, TResult>(this T? value, Func<T, Task<TResult>>
onSome, Func<Task<TResult>> onNone) where T : class
```

Parameters

value T

onSome Func☑<T, Task☑<TResult>>

onNone Func☑<Task☑<TResult>>

Returns

Task☑<TResult>

Type Parameters

T

TResult

Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)

```
public static Task<TResult> Some<T, TResult>(this T? value, Func<T, Task<TResult>>
onSome, Func<TResult> onNone) where T : class
```

Parameters

value T

onSome Func☑<T, Task☑<TResult>>

onNone Func☑<TResult>

Returns

Task☑<TResult>

Type Parameters

T

TResult

Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)

```
public static Task<TResult> Some<T, TResult>(this T? value, Func<T, TResult> onSome,
Func<Task<TResult>> onNone) where T : class
```

Parameters

value T

onSome Func <T, TResult>

onNone Func <Task <TResult>>

Returns

Task <TResult>

Type Parameters

T

TResult

Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)

```
public static TResult Some<T, TResult>(this T? value, Func<T, TResult> onSome,
Func<TResult> onNone) where T : class
```

Parameters

`value` T

`onSome` [Func]⧉ `<T, TResult>`

`onNone` [Func]⧉ `<TResult>`

Returns

TResult

Type Parameters

`T`

`TResult`

Class ResultExtensions

Namespace: [Solstice](#).[Domain](#).[Utils](#)

Assembly: Solstice.Domain.dll

```
public static class ResultExtensions
```

Inheritance

[object](#) ← ResultExtensions

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) ,
[object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) ,
[object.ToString()](#)

Methods

Match<TSuccess, TError, TResult>(Result<TSuccess, TError>, Func<TSuccess, TResult>, Func<TError, TResult>)

```
public static TResult Match<TSuccess, TError, TResult>(this Result<TSuccess, TError>
result, Func<TSuccess, TResult> onSuccess, Func<TError, TResult> onError)
```

Parameters

`result` [Result](#)<TSuccess, TError>

`onSuccess` [Func](#)<TSuccess, TResult>

`onError` [Func](#)<TError, TResult>

Returns

TResult

Type Parameters

TSuccess

TError

TResult

TSuccess

TError

Class Result<TSuccess, TError>

Namespace: [Solstice](#).[Domain](#).[Utils](#)

Assembly: Solstice.Domain.dll

```
public class Result<TSuccess, TError>
```

Type Parameters

TSuccess

TError

Inheritance

[object](#) ← Result<TSuccess, TError>

Inherited Members

[object.Equals(object)](#) , [object.Equals(object, object)](#) , [object.GetHashCode()](#) , [object.GetType()](#) , [object.MemberwiseClone()](#) , [object.ReferenceEquals(object, object)](#) , [object.ToString()](#)

Extension Methods

[MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, Task<TResult>>, Func<TResult>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<Task<TResult>>)](#) , [MatchExtensions.Some<T, TResult>(T?, Func<T, TResult>, Func<TResult>)](#) , [ResultExtensions.Match<TSuccess, TError, TResult>(Result<TSuccess, TError>, Func<TSuccess, TResult>, Func<TError, TResult>)](#)

Constructors

Result(TSuccess, TError, bool)

```
protected Result(TSuccess success, TError error, bool isSuccess)
```

Parameters

success TSuccess

error TError

isSuccess [bool⊘](#)

Properties

Error

```
public TError Error { get; }
```

Property Value

TError

IsSuccess

```
public bool IsSuccess { get; }
```

Property Value

[bool⊘](#)

Success

```
public TSuccess Success { get; }
```

Property Value

TSuccess

Methods

Err(TError)

```
public static Result<TSuccess, TError> Err(TError error)
```

Parameters

error TError

Returns

[Result](#)<TSuccess, TError>

Ok(TSuccess)

```
public static Result<TSuccess, TError> Ok(TSuccess success)
```

Parameters

success TSuccess

Returns

[Result](#)<TSuccess, TError>