

Table of Contents

Articles

[Introduction](#)

[pi-cli](#)

[Introduction](#)

[Concepts](#)

[Options](#)

[Hosting](#)

[Deployment](#)

[Pricing](#)

[Details](#)

[Roadmap](#)

[Samples](#)

[Repo](#)

[protocols](#)

[Introduction](#)

[Repo](#)

[Featured Samples](#)

Classes

[PerpetualIntelligence.Cli](#)

[Errors](#)

[Terminal](#)

[PerpetualIntelligence.Cli.Authentication](#)

[AuthenticationDelegateHandler](#)

[ClientCrossPlatformNoTokenCache](#)

[IClientCrossPlatformTokenCache](#)

[IMsalPublicClientApplicationFactory](#)

[MsalInteractiveAuthenticationProvider](#)

[MsalPublicClientApplicationFactory](#)

[MsalPublicClientApplicationFactoryContext](#)

[MsalPublicClientApplicationFactoryResult](#)

[PerpetualIntelligence.Cli.Commands](#)

[Argument](#)

[ArgumentDescriptor](#)

ArgumentDescriptors

Arguments

ArgumentString

ArgumentStrings

Command

CommandDescriptor

CommandString

PerpetualIntelligence.Cli.Commands.Checkers

ArgumentChecker

ArgumentCheckerContext

ArgumentCheckerResult

CommandChecker

CommandCheckerContext

CommandCheckerResult

IArgumentChecker

ICommandChecker

PerpetualIntelligence.Cli.Commands.Comparers

StringComparisonComparer

PerpetualIntelligence.Cli.Commands.Extractors

ArgumentExtractor

ArgumentExtractorContext

ArgumentExtractorResult

CommandExtractor

CommandExtractorContext

CommandExtractorResult

IArgumentExtractor

ICommandExtractor

PerpetualIntelligence.Cli.Commands.Handlers

CommandHandler

CommandHandlerContext

CommandHandlerResult

ICommandHandler

PerpetualIntelligence.Cli.Commands.Mappers

ArgumentDataTypeMapperContext

ArgumentDataTypeMapperResult

DataAnnotationsArgumentDataTypeMapper

IArgumentDataTypeMapper

PerpetualIntelligence.Cli.Commands.Providers

- DefaultArgumentProvider
- DefaultArgumentProviderContext
- DefaultArgumentProviderResult
- DefaultArgumentValueProvider
- DefaultArgumentValueProviderContext
- DefaultArgumentValueProviderResult
- IDefaultArgumentProvider
- IDefaultArgumentValueProvider

PerpetualIntelligence.Cli.Commands.Publishers

- ErrorPublisher
- ErrorPublisherContext
- ExceptionPublisher
- ExceptionPublisherContext
- IPublisher
- IExceptionPublisher

PerpetualIntelligence.Cli.Commands.Routers

- CommandRouter
- CommandRouterContext
- CommandRouterResult
- ICommandRouter

PerpetualIntelligence.Cli.Commands.Runners

- ClearScreenRunner
- CommandRunner
- CommandRunnerContext
- CommandRunnerResult
- ExitRunner
- ICommandRunner
- LicInfoRunner
- RunRunner

PerpetualIntelligence.Cli.Commands.Stores

- ICommandDescriptorStore

PerpetualIntelligence.Cli.Configuration.Options

- AuthenticationOptions
- CheckerOptions
- CliOptions
- ExtractorOptions

- HostingOptions
- HttpOptions
- LicensingOptions
- LoggingOptions
- TerminalOptions
- PerpetualIntelligence.Cli.Extensions
 - CommandExtensions
 - ICliBuilderExtensions
 - IHostExtensions
 - IServiceCollectionExtensions
- PerpetualIntelligence.Cli.Integration
 - CliBuilder
 - CliHostedService
 - ICliBuilder
- PerpetualIntelligence.Cli.Licensing
 - ILicenseChecker
 - ILicenseExtractor
 - ILicenseProviderResolver
 - License
 - LicenseChecker
 - LicenseCheckerContext
 - LicenseCheckerResult
 - LicenseExtractor
 - LicenseExtractorContext
 - LicenseExtractorResult
 - LicenseLimits
 - LicenseProviderResolver
- PerpetualIntelligence.Cli.Services
 - ConsoleClearPosition
 - ConsoleHelper
- PerpetualIntelligence.Cli.Stores.InMemory
 - InMemoryCommandDescriptorStore
- PerpetualIntelligence.Protocols
 - Constants
- PerpetualIntelligence.Protocols.Abstractions
 - IChecker<TContext, TResult>
 - IExtractor<TContext, TResult>

[IFormatter<TContext, TResult>](#)

[IHandler<TContext, TResult>](#)

[IMapper<TContext, TResult>](#)

[IProcessor<TContext>](#)

[IProducer<TContext, TResult>](#)

[IProvider<TContext, TResult>](#)

[IPublisher<TContext, TResult>](#)

[IPublisherNoResult<TContext>](#)

[IRecorder<TContext, TResult>](#)

[IRouter<TContext, TResult, THandler>](#)

[IRunner<TContext, TResult>](#)

[ISubscriber<TContext, TResult>](#)

[PerpetualIntelligence.Protocols.Abstractions.Authorization](#)

[IAuthHandler<TContext, TResult>](#)

[PerpetualIntelligence.Protocols.Abstractions.Comparers](#)

[IClaimComparer](#)

[IStringComparer](#)

[PerpetualIntelligence.Protocols.Abstractions.Concurrency](#)

[IThreadLock<T>](#)

[PerpetualIntelligence.Protocols.Abstractions.Events](#)

[IEvent](#)

[IEventPublisher<TContext, TResult>](#)

[IEventRecorder<TContext, TResult>](#)

[IEventSubscriber<TContext, TResult>](#)

[PerpetualIntelligence.Protocols.Abstractions.Stores](#)

[IEntity](#)

[IEntityTenancy](#)

[IMarketplaceEntity](#)

[IStore<TContext, TMContext, TResult, TMResult>](#)

[IStoreMultipleResult<TEntity>](#)

[IStoreSingleResult<TEntity>](#)

[ITrackedStore<TContext, TMContext, TResult, TMResult>](#)

[PerpetualIntelligence.Protocols.Authorization](#)

[AppTypes](#)

[ClaimTypes](#)

[HeaderKeys](#)

[IAppIdentifiers](#)

Policies

WellKnownBaseAddress

PerpetualIntelligence.Protocols.Defaults.Comparers

OrderIndependentComparer

PerpetualIntelligence.Protocols.Defaults.Concurrency

SemaphoreSlimThreadLock<T>

PerpetualIntelligence.Protocols.Licensing

SaaSCheckModes

SaaSKeySources

SaaSPlans

SaaSProviders

SaaSUsages

PerpetualIntelligence.Protocols.Licensing.Models

LicenseCheckModel

LicenseClaimsModel

LicenseKeyJsonFileModel

LicenseKeysModel

LicenseProvisioningModel

PerpetualIntelligence.Protocols.Security

HexSaltGenerator

IdGenerator

PerpetualIntelligence.Protocols.Security.Certificates

X509Store

PerpetualIntelligence.Protocols.Security.Secrets

ExtractedSecret

Secret

SecretComparer

SecretTypes

PerpetualIntelligence.Shared.Abstractions

ICommunicationSender

IComponentIdGenerator

ISender

IEncoder

IdAccessor

IdentityAccessor

IdGenerator

ILocalizer

IModeAccessor

INameAccessor

IOneImlxResult

ITextMessageSender

PerpetualIntelligence.Shared.Attributes

ActionAttribute

ArchitectureAttribute

ConceptAttribute

FutureAttribute

InternalInfrastructureAttribute

MustDoAttribute

PerformanceAttribute

RedundancyAttribute

RefactorAttribute

RenameAttribute

TodoAttribute

WriteApiAttribute

WriteDocumentationAttribute

WriteIntegrationTestAttribute

WriteUnitTestAttribute

PerpetualIntelligence.Shared.Attributes.Api

EndpointAttribute

HiddenPropertyAttribute

ServiceAttribute

PerpetualIntelligence.Shared.Attributes.Rendering

GroupOrderAttribute

OverrideDisplayAttribute

SubObjectPropertyAttribute

PerpetualIntelligence.Shared.Attributes.Validation

EmailListAttribute

OneOfAttribute

SelectItemAttribute

StringLengthListAttribute

UrlListAttribute

PerpetualIntelligence.Shared.Exceptions

ErrorException

MultiErrorException

PerpetualIntelligence.Shared.Extensions

IEnumerableExtensions

ILoggerExtensions

JsonElementExtensions

StringExtensions

PerpetualIntelligence.Shared.Infrastructure

CustomDataTypes

Error

Identity

LocalStorageKeys

LoggingOptions

ModelIdentity

Modes

OrgConstants

RegexPatterns

ResultDelegate<TContext, TResult>

TryResultOrError<T>

PerpetualIntelligence.Shared.Patterns

Decorator<TObject>

DecoratorService<TService, TImpl>

DisposableDecorator<TObject>

DisposableDecoratorService<TService, TImpl>

PerpetualIntelligence.Shared.Services

InfraHelper

UrlSafeBase64Encoder

PerpetualIntelligence.Test

ContextTests

HttpClientTests

InitializerTests

LoggerTests<T>

TestIdAttribute

PerpetualIntelligence.Test.Services

TestHelper

TestLogger

About

Introduction

Welcome to the conceptual and class documentation for [Perpetual Intelligence's](#) managed services and frameworks.

- Browse our [Nuget packages](#)
- Browse code on [GitHub](#)
- Browse working [samples](#)

Note: This is a *preview* release. It is also subject to design changes without any advance notice.

Introducing pi-cli

`pi-cli` is the cross-platform developer framework for building all your CLI terminals or command-line systems in the .NET ecosystem. The framework makes it easy to build CLIs for your company, product, service, SaaS, or development and testing needs. Enterprises can create CLIs with few flags or advanced complex CLIs with organization commands, grouped commands, subcommands, arguments, and options.

Take your app or service to the command line with Unicode support and build your front-end CLI in any language.

Craftsmanship

We crafted the `pi-cli` framework to be cross-platform, hosting and deployment agnostic, and fully customizable. We strongly believe .NET provides a rich set of [DSL](#) and [DDD](#) tools and languages, and `pi-cli` directly supports the .NET (traditional), .NET core, ASP.NET Core, and NET6+ framework. Thus it is naturally the defacto standard in developing cross-platform CLI systems for your apps, services, and developer tools in the entire .NET ecosystem. It lets enterprises build ground-up CLI terminals or migrate their existing CLI apps and terminals with the modern and scalable micro-services-based architecture.

In short, if what you want to achieve is doable in the .NET ecosystem, it is possible with `pi-cli`.

Getting Started

Open Source

Our entire source code is on [GitHub](#). It enables community collaboration, troubleshoot issues, and helps get us your feedback on the features and documentation. It also promotes a better understanding of architecture and design.

***See our [licensing terms](#) and [pricing](#) **.*

OS

Our CICD pipeline builds the framework with Github [hosted runners](#) for the following OS platform. However, it supports all the additional platforms that .NET supports.

macos 11

linux ubuntu-20.04

windows 2022

Packaging

The licensed libraries can be accessed via Nuget:

`PerpetualIntelligence.Cli v2.4.1-preview.220421591`

Build

The [GitHub](#) repo contains all the build and release artifacts to build, test and publish the `pi-cli` source.



Terminal UX

The `pi-cli` framework does not enforce any specific terminal or console UX experience because this is always custom to the project. However, we provide you with a starting point for your terminal lifetime and UX customization.

- [CliHostedService](#)

Classes

[Classes and Object browser](#)

Learn to Use

With the `pi-cli` framework, you don't have to be a microservices or distributed systems expert to build a modern and scalable CLI terminal. You build and learn as you go on, and eventually, you become an expert :) similar to an [eventually-consistent system](#). You can quickly build CLIs for simple use cases, a CLI terminal that handles authentication, or a CLI terminal that interacts with a complex distributed system via protected APIs. We believe in an agile development and agile learning. So, pick a learning model that works for you!

I want to create my first modern CLI and learn as I go on

- Please create an account with us at <https://www.perpetualintelligence.com>
- Pick a pricing plan that works for you. Our community edition is free for educational, research, and non-commercial use. You must have a commercial license to use the service in a non-educational, commercial, or production environment.
- Browse our code samples [here](#), clone the [repo](#)
- Set your License Key (a valid license key is required for both community and commercial licenses)
- Build, debug make changes and learn the concepts and you go on
- We cant wait to see the fantastic CLI terminals you build !

I want to understand the concepts first

Continue reading, and we will explain all the concepts. We recommend you get familiar with the common architectural principles first as they enable our framework to be extendible, customizable, and remain scalable.

- [Dependency Injection](#)
- [Dependency Inversion](#)
- [Options Patterns](#)
- [Separation of concerns](#)
- [Single Responsibility](#)
- [Bounded Context](#)

Test License

If you want to evaluate the pi-cli framework quickly, you can use our test license with shared limits.

We recommend creating an account to generate a license for your specific use case. Our community edition is free for educational, research and development, and non-commercial use. You do need a commercial license for a non-educational or production environment.

Note: Our test license feature is under development and will be available soon. See <https://github.com/perpetualintelligence/cli/issues/15> for more status.

Issues and feature requests

Please report [issue](#) or [feature request](#) directly on our official github repo.

References

- [Microservices](#)
- [.NET application architecture](#)

Note: This is a *preview* release. It is also subject to design changes without any advance notice.

Concepts

Terminal

Terminals, also known as command lines, consoles, or CLI applications, allow organizations and users to accomplish and automate tasks on a computer without using a graphical user interface. If a CLI app supports user interaction, the UX is the terminal.

Commands

The [PerpetualIntelligence.Cli.Commands](#) namespace defines all the code constructs to describe the command and its arguments, extract command from the command string, route to the registered command handler, perform data type and strict type checks, and finally run the command.

CommandString

The [PerpetualIntelligence.Cli.Commands.CommandString](#) class is an immutable Unicode textual form representing the command and its arguments or options that a user or an application wants to execute.

Example:

```
gh issue list  
gh issue create --label bug  
dotnet build --runtime ubuntu.18.04-x64
```

CommandDescriptor

The [PerpetualIntelligence.Cli.Commands.CommandDescriptor](#) class defines the command identity and its supported arguments that an end-user or an application can use. You can also describe the command behavior, such as whether the command is a root, grouped, or subcommand.

Root Command

A root command is the top CLI command. It can represent your organization, a product, or a service. For instance, Github CLI [gh](#) is an example of an organization root command. Microsoft, however, uses [dotnet](#) as a root command for the .NET CLI.

Grouped Command

A grouped command provides a context for a set of related sub-commands. For instance, Github CLI [gh auth](#) is an example of a grouped command to authenticate gh and git with GitHub.

Sub Command

A subcommand is an individual executable command that performs a specific action. For instance, Github CLI [gh auth login](#) is an example of a subcommand that authenticates with GitHub host. [dotnet build](#) is a sub-command that builds a project and all of its dependencies.

Note: A command is a subcommand unless you designate it as a root or a grouped command.

Command

The [PerpetualIntelligence.Cli.Commands.Command](#) class is a runtime validated representation of an actual command and its argument values passed by a user or an application. It represents a specific action or a set of actions that a user or an application requests the underlying system to perform. It can be a simple action such as invoking a system method or an OS command or representing a complex operation that calls a set of protected APIs over the internal or external network. A command can virtually do anything in the context of your application or service.

ArgumentDescriptor

The [PerpetualIntelligence.Cli.Commands.ArgumentDescriptor](#) class defines the command argument identity, data type, and data validation behavior. We also refer to arguments as command options or command flags. An argument [Id](#) is always unique within a command. By design it implements the default equality `<xref:System.IEquatable`1>` using [Id](#) property. Thus, two arguments with the same id are equal irrespective of other property values. This is done to improve performance during lookup and avoid multiple arguments with same identifiers.

Argument

The [PerpetualIntelligence.Cli.Commands.Argument](#) is a runtime validated representation of an actual command argument, option, or a flag and its value passed by a user or an application.

Integration

The [PerpetualIntelligence.Cli.Integration](#) namespace defines all the code constructs to integrate your CLI terminal with the pi-cli framework. It provides a service builder for [dependency injection](#) and hosts a service to manage terminal lifetime and customization.

CliHostedService

The [PerpetualIntelligence.Cli.Integration.CliHostedService](#) is a hosted service that manages application lifetime, performs licensing checks, and enables terminal UX customization.

Terminal Lifetime

You can override the following terminal lifetime methods in your application context.

RegisterHostApplicationEventsAsync

Allows the application to register its custom events. The default implementation registers `OnStarted`, `OnStopping`, and `OnStopped` events that application authors can override.

```
protected virtual Task RegisterHostApplicationEventsAsync(IHostApplicationLifetime
hostApplicationLifetime)
{
    hostApplicationLifetime.ApplicationStarted.Register(OnStarted);
    hostApplicationLifetime.ApplicationStopping.Register(OnStopping);
    hostApplicationLifetime.ApplicationStopped.Register(OnStopped);
    return Task.CompletedTask;
}
```

Note: `OnStarted`, `OnStopping`, and `OnStopped` are not triggered if you override `RegisterHostApplicationEventsAsync` and register your custom events.

OnStarted

Triggered when the pi-cli application host has fully started.

```
protected virtual void OnStarted()
{
    Console.WriteLine("Server started on {0}.", DateTime.UtcNow.ToLocalTime().ToString());
    Console.WriteLine();
}
```

OnStopping

Triggered when the pi-cli application host is starting a graceful shutdown. Shutdown will block until all callbacks registered on this token have completed.

```
protected virtual void OnStopping()
{
    Console.WriteLine("Stopping server...");
}
```

OnStopped

Triggered when the pi-cli application host has completed a graceful shutdown. The application will not exit until all callbacks registered on this token have completed.

```
protected virtual void OnStopped()
{
    ConsoleHelper.WriteLineColor(ConsoleColor.Red, "Server stopped on {0}.",
    DateTime.UtcNow.ToLocalTime().ToString());
}
```

Terminal Header

You can override the following method to print the terminal header in your application context.

PrintHostApplicationHeaderAsync

Allows the host application to print the custom header.

```
protected virtual Task PrintHostApplicationHeaderAsync()
{
    Console.WriteLine("-----");
    Console.WriteLine("Copyright (c) Perpetual Intelligence L.L.C. All Rights Reserved.");
    Console.WriteLine("For license, terms, and data policies, go to:");
    Console.WriteLine("https://terms.perpetualintelligence.com");
    Console.WriteLine("-----");

    Console.WriteLine($"Starting server \"{Protocols.Constants.CliUrn}\" version=
{typeof(CliHostedService).Assembly.GetCustomAttribute<AssemblyInformationalVersionAttribute>
()}.InformationalVersion ?? " < none > "}");
    return Task.CompletedTask;
}
```

Terminal Licensing Information

You can override the following method to print the terminal licensing information in your application context.

PrintHostApplicationLicensingAsync

Allows host application to print custom licensing information.

```

        protected virtual Task PrintHostApplicationLicensingAsync(PerpetualIntelligence.Cli.Licensing.License
license)
        {
            // Print the license information
            ConsoleHelper.WriteLineColor(ConsoleColor.Cyan, $"consumer={license.Claims.Name}
({license.Claims.TenantId})");
            ConsoleHelper.WriteLineColor(ConsoleColor.Cyan, $"country={license.Claims.TenantCountry}");
            ConsoleHelper.WriteLineColor(ConsoleColor.Cyan, $"subject={cliOptions.Licensing.Subject}");
            ConsoleHelper.WriteLineColor(ConsoleColor.Cyan, $"check={license.CheckMode}");
            ConsoleHelper.WriteLineColor(ConsoleColor.Cyan, $"usage={license.Usage}");
            ConsoleHelper.WriteLineColor(ConsoleColor.Green, $"edition={license.Plan}");
            ConsoleHelper.WriteLineColor(ConsoleColor.Cyan, $"key_source={cliOptions.Licensing.KeySource}");
            if (license.LicenseKeySource == SaaSKeySources.JsonFile)
            {
                // Don't dump the key, just the lic file path
                ConsoleHelper.WriteLineColor(ConsoleColor.Cyan, $"key_file={license.LicenseKey}");
            }
            return Task.CompletedTask;
        }
    }

```

ICliBuilder

You enable the pi-cli framework to any .NET, .NET Core, or .NET6+ console application by adding the relevant services to the [dependency injection \(DI\)](#) system.

```

public static async Task Main(string[] args)
{
    // The cancellation token routing commands.
    CancellationTokenSource cancellationTokenSource = new();

    // Setup the host builder
    IHostBuilder hostBuilder = CreateHostBuilder(args, Startup.ConfigureServices);

    // Start the host. We don't call Run as it will block the thread. We want to listen to user inputs.
    using (var host = await hostBuilder.StartAsync(cancellationTokenSource.Token))
    {
        // Run the router loop to listen to the user input and process the command string
        await host.RunRouterAsync("_> ", cancellationTokenSource.Token);
    }
}

// Add pi-cli to the console app
public void ConfigureServices(IServiceCollection services)
{
    ICliBuilder builder = services.AddCli(options => { ... });
}

```

Many of the fundamental CLI terminal configuration settings can be set on the options. See the [CliOptions](#) for more details. The [PerpetualIntelligence.Cli.Extensions.ICliBuilderExtensions](#) provides dependency injection extension methods to register required and optional pi-cli services.

Note: This is a *preview* release. It is also subject to design changes without any advance notice.

CliOptions

The pi-cli framework uses the options pattern to provide strongly typed access to groups of related settings. When configuration settings are isolated by scenario into separate classes, the host CLI terminal adheres to two crucial software engineering principles:

- [Interface Segregation Principle \(ISP\) or Encapsulation](#): Scenarios (classes) that depend on configuration settings depend only on the configuration settings that they use.
- [Separation of Concerns](#): Settings for different parts of the app aren't dependent or coupled to one another. Options also provide a mechanism to validate configuration data. For more information, see the Options validation section.

This article provides information on all the supported configuration options by the pi-cli framework.

AuthenticationOptions

CheckerOptions

ExtractorOptions

HostingOptions

HttpOptions

LicensingOptions

LoggingOptions

References

- [Options pattern in .NET](#)
- [Architectural principles](#)

Note: This is a ***preview*** release. It is also subject to design changes without any advance notice.

Hosting

The pi-cli framework is hosting agnostic, meaning no hosting limitations at all. Application authors can host their CLI apps, terminals, or servers on their self-hosting environment, use a managed-hosting environment, or rely on a third party to provide a hosting environment. You can configure your server and provide your self-hosting implementations for stores and host in an environment of your choice, e.g., Windows, Linux, macOS, Docker, Kubernetes, etc.

Note: This is a ***preview*** release. It is also subject to design changes without any advance notice.

Deployment

With pi-cli, you build deployment agnostic secured CLI terminals and services, test them in local environments and deploy the production apps and services on-premise, cloud (public, private, or government), or hybrid. You can also automate the deployment of your apps and services as portable, self-sufficient containers that can run on the cloud or on-premises.

Public Client CLI Terminals

Server Deployed CLI Terminals

Note: This is a *preview* release. It is also subject to design changes without any advance notice.

Pricing Plans

Select a pricing plan that best suits your needs. Go to features for a detailed feature comparison based on a pricing plan.

Community

The community edition is free for educational, research, and non-commercial use.

Note: You must have a commercial license to use the service in a non-educational or production environment.

Micro

The micro edition is a commercial license for solo users or micro-businesses with up to 50 employees.

SMB

The SMB edition is a commercial license for small and medium businesses with up to 500 employees.

Enterprise

The enterprise edition is a commercial license for organizations and large businesses with more than 500 employees.

ISV

The ISV (Independent Software Vendor) is a commercial enterprise license for scale and redistribution. It allows internal self-hosting, SaaS, or hosting on behalf of your customers and redistribution to a customer's on-premise or cloud infrastructure. ISV can redistribute the "pi-cli" framework as part of a standard or custom product.

ISV-U

The ISV-U (Independent Software Vendor) is an unlimited commercial enterprise license for massive scale and redistribution. It allows internal self-hosting, SaaS, or hosting on behalf of your customers and redistribution to a customer's on-premise or cloud infrastructure. ISV can redistribute the "pi-cli" framework as part of a standard or custom product.

Features

Starters Pack

FEATURES	COMMUNITY	SOLO OR MICRO	SMB
Pricing <div>\$ USD</div>	<div>\$ 0</div>	<div>\$ 19 / Month</div> <div>\$ 199 / Year</div>	<div>\$ 119 / Month</div> <div>\$ 1209 / Year</div>
Terminals or Console Apps	1	1	3
Redistribution	NO	NO	1000

FEATURES	COMMUNITY	SOLO OR MICRO	SMB
Root commands	1	1	1
Grouped commands	5	5	10
Sub-commands	50	50	100
Arguments, options or flags	500	500	1000
Argument alias	NO	NO	Yes
Default argument and value	NO	NO	Yes
Strict data type	NO	NO	Yes
Unicode	Yes	Yes	Yes
Error handling	standard	standard	standard
Stores	in-memory	in-memory	in-memory □ json
Routing and extraction	standard	standard	standard
License Check	online	online	online

□ - Feature planed for future release.

Pro Pack

FEATURES	ENTERPRISE	ISV	ISV-U
Pricing \$ USD	\$ 219 / Month \$ 2309 / Year	\$ 619 / Month \$ 6609 / Year	\$ 1219 / Month \$ 13109 / Year
Terminals or Console Apps	10	25	UNLIMITED
Redistribution	5000	10000	UNLIMITED
Root commands	3	5	10
Grouped commands	50	100	UNLIMITED
Sub-commands	500	1000	UNLIMITED
Arguments, options or flags	5000	10000	UNLIMITED
Argument alias	Yes	Yes	Yes
Default argument and value	Yes	Yes	Yes

FEATURES	ENTERPRISE	ISV	ISV-U
Strict data type	Yes	Yes	Yes
Unicode	Yes	Yes	Yes
Error handling	<div>standard</div> <div>custom</div>	<div>standard</div> <div>custom</div>	<div>standard</div> <div>custom</div>
Stores	<div>in-memory</div> <div><input type="checkbox"/> json</div> <div><input type="checkbox"/> custom</div>	<div>in-memory</div> <div><input type="checkbox"/> json</div> <div><input type="checkbox"/> custom</div>	<div>in-memory</div> <div><input type="checkbox"/> json</div> <div><input type="checkbox"/> custom</div>
Routing and extraction	<div>standard</div> <div><input type="checkbox"/> custom</div>	<div>standard</div> <div><input type="checkbox"/> custom</div>	<div>standard</div> <div><input type="checkbox"/> custom</div>
License Check	<div>online</div> <div><input type="checkbox"/> offline</div>	<div>online</div> <div><input type="checkbox"/> offline</div> <div><input type="checkbox"/> byol</div>	<div>online</div> <div><input type="checkbox"/> offline</div> <div><input type="checkbox"/> byol</div>

☐ - Feature planed for future release.

Behind the scenes

Runtime

Why Descriptor By Actual Classes

Default Values

Argument Alias

Routing

Extractors

Handlers

Comparers

Providers

Checkers

Runners

Publishers

Configurations and Customizations

Options

Regex Patterns

Unicode

Extensions

Integration

Stores

Licensing & Pricing

Errors

Authentication

Support

Pricing

Note: This is a ***preview*** release. It is also subject to design changes without any advance notice.

Note: This is a *preview* release. It is also subject to design changes without any advance notice.

Samples

Learn to use the "pi-cli" framework by exploring our working samples on [Github](#). Just clone the Visual Studio solution, update the configuration options with your license key, and start exploring!

GitHub CLI

Build the command line terminals similar to [GitHub Cli](#). Start by cloning the repository [github-style-cli](#).

.NET CLI

Build the command line terminals similar to [.NET Cli](#). Start by cloning the repository [dotnet-style-cli](#).

Traditional CLI

Build the command line terminal in good old key=value arguments. Start by cloning the repository [traditional-keyvalue-cli](#).

Note: This is a **preview** release. It is also subject to design changes without any advance notice.

perpetualintelligence/cli

This repository contains the cross-platform "pi-cli" framework. We build the following NuGet packages from this repository.

PerpetualIntelligence.Cli v2.4.1-preview.220421591

We track the [issues](#) and [tasks](#) here. We make our best effort to respond to issues in a timely fashion. You can read more about our procedures for classifying and resolving issues in our [Issues policy](#) topic.

We welcome contributions to help us improve and complete the docs.

This project has adopted the code of conduct defined by the Contributor Covenant to clarify expected behavior in our community. See the [Code of Conduct](#).

Introduction

status in-progress

protocols

release preview

Note: This is a ***preview*** release. It is also subject to design changes without any advance notice.

This repository contains the cross-platform protocols and standards abstraction for Perpetual Intelligence L.L.C. managed services, frameworks, and tools. We build the following NuGet packages from this repository.

PerpetualIntelligence.Shared v2.4.1-preview.2204201641

PerpetualIntelligence.Test v2.4.1-preview.2204201641

PerpetualIntelligence.Protocols v2.4.1-preview.2204201641

PerpetualIntelligence.Protocols.Defaults v2.4.1-preview.2204201641




We track the [issues](#) and [tasks](#) here. We make our best effort to respond to issues in a timely fashion. You can read more about our procedures for classifying and resolving issues in our [Issues policy](#) topic.

We welcome contributions to help us improve and complete the docs.

This project has adopted the code of conduct defined by the Contributor Covenant to clarify expected behavior in our community. See the [Code of Conduct](#).

Featured Samples

Browse our code samples and tutorials on [Github](#).

	TUTORIAL	DESCRIPTION	SERVICE
	github-style-cli	Learn to build modern CLI terminals similar to Github CLI	pi-cli
	dotnet-style-cli	Learn to build modern CLI terminals similar to .NET Core CLI	pi-cli
	keyvalue-cli	Learn to build CLI terminals with good old traditional commands and key=value arguments.	pi-cli

Namespace PerpetualIntelligence.Cli

Classes

Errors

The errors for the Perpetual Intelligence's `pi-cli` framework.

Terminal

Terminals, also known as command lines, consoles, or CLI applications, allow organizations and users to accomplish and automate tasks on a computer without using a graphical user interface. If a CLI terminal supports user interaction, the UX is the terminal.

Class Errors

The errors for the Perpetual Intelligence's `pi-cli` framework.

Inheritance

System.Object

Errors

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public static class Errors
```

Fields

DuplicateArgument

The argument is not already added to the command.

Declaration

```
public const string DuplicateArgument = "duplicate_argument"
```

Field Value

TYPE	DESCRIPTION
System.String	

InvalidArgument

The argument is not valid.

Declaration

```
public const string InvalidArgument = "invalid_argument"
```

Field Value

TYPE	DESCRIPTION
System.String	

InvalidCommand

The command is not valid.

Declaration

```
public const string InvalidCommand = "invalid_command"
```

Field Value

TYPE	DESCRIPTION
System.String	

InvalidConfiguration

The configuration is not valid.

Declaration

```
public const string InvalidConfiguration = "invalid_configuration"
```

Field Value

TYPE	DESCRIPTION
System.String	

InvalidLicense

The license is not valid.

Declaration

```
public const string InvalidLicense = "invalid_license"
```

Field Value

TYPE	DESCRIPTION
System.String	

InvalidRequest

The request is not valid.

Declaration

```
public const string InvalidRequest = "invalid_request"
```

Field Value

TYPE	DESCRIPTION
System.String	

MissingArgument

The argument is missing.

Declaration

```
public const string MissingArgument = "missing_argument"
```

Field Value

TYPE	DESCRIPTION
System.String	

MissingClaim

The claim is missing.

Declaration

```
public const string MissingClaim = "missing_claim"
```

Field Value

TYPE	DESCRIPTION
System.String	

RequestCanceled

The request is canceled.

Declaration

```
public const string RequestCanceled = "request_canceled"
```

Field Value

TYPE	DESCRIPTION
System.String	

ServerError

The server error.

Declaration

```
public const string ServerError = "server_error"
```

Field Value

TYPE	DESCRIPTION
System.String	

UnauthorizedAccess

The access is not authorized.

Declaration

```
public const string UnauthorizedAccess = "unauthorized_access"
```

Field Value

TYPE	DESCRIPTION
System.String	

UnsupportedArgument

The argument is not supported.

Declaration

```
public const string UnsupportedArgument = "unsupported_argument"
```

Field Value

TYPE	DESCRIPTION
System.String	

UnsupportedCommand

The command is not supported.

Declaration

```
public const string UnsupportedCommand = "unsupported_command"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class Terminal

Terminals, also known as command lines, consoles, or CLI applications, allow organizations and users to accomplish and automate tasks on a computer without using a graphical user interface. If a CLI terminal supports user interaction, the UX is the terminal.

Inheritance

System.Object
Terminal

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class Terminal
```

Properties

AuthorizedApplicationId

The authorized application id for this terminal.

Declaration

```
[JsonPropertyName("authorized_application_id")]  
public string AuthorizedApplicationId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Cli.Authentication

Classes

[AuthenticationDelegateHandler](#)

Submits an authentication request asynchronously.

[ClientCrossPlatformNoTokenCache](#)

The [IClientCrossPlatformTokenCache](#) that represents no caching.

[MsalInteractiveAuthenticationProvider](#)

The `MSAL` authentication provider.

[MsalPublicClientApplicationFactory](#)

The [IMsalPublicClientApplicationFactory](#) to create `Microsoft.Identity.Client.IPublicClientApplication`.

[MsalPublicClientApplicationFactoryContext](#)

The [IMsalPublicClientApplicationFactory](#) context.

[MsalPublicClientApplicationFactoryResult](#)

The [IMsalPublicClientApplicationFactory](#) result.

Interfaces

[IClientCrossPlatformTokenCache](#)

An abstraction of a client's cross platform token cache. Client applications (desktop and mobile apps) should try to get a token from the cache before acquiring a token by another method.

[IMsalPublicClientApplicationFactory](#)

An abstraction to create `Microsoft.Identity.Client.IClientApplicationBase`.

Class AuthenticationDelegateHandler

Submits an authentication request asynchronously.

Inheritance

System.Object
System.Net.Http.HttpMessageHandler
System.Net.Http.DelegatingHandler
AuthenticationDelegateHandler

Implements

System.IDisposable

Inherited Members

System.Net.Http.DelegatingHandler.Dispose(System.Boolean)
System.Net.Http.DelegatingHandler.InnerHandler
System.Net.Http.HttpMessageHandler.Dispose()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class AuthenticationDelegateHandler : DelegatingHandler, IDisposable
```

Constructors

AuthenticationDelegateHandler(IAuthenticationProvider)

Initializes a new instance.

Declaration

```
public AuthenticationDelegateHandler(IAuthenticationProvider authenticationProvider)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Graph.IAuthenticationProvider	authenticationProvider	

Methods

SendAsync(HttpRequestMessage, CancellationToken)

Sends a request asynchronously.

Declaration

```
protected override async Task<HttpResponseMessage> SendAsync(HttpRequestMessage request, CancellationToken cancellationToken)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpRequestMessage	request	
System.Threading.CancellationToken	cancellationToken	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< HttpResponseMessage >	

Overrides

System.Net.Http.DelegatingHandler.SendAsync(System.Net.Http.HttpRequestMessage, System.Threading.CancellationToken)

Implements

System.IDisposable

Class ClientCrossPlatformNoTokenCache

The [IClientCrossPlatformTokenCache](#) that represents no caching.

Inheritance

System.Object
ClientCrossPlatformNoTokenCache

Implements

[IClientCrossPlatformTokenCache](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ClientCrossPlatformNoTokenCache : IClientCrossPlatformTokenCache
```

Methods

RegisterCacheAsync(IClientApplicationBase)

This method does nothing and it is provided for infrastructure consistency.

Declaration

```
public Task RegisterCacheAsync(IClientApplicationBase clientApplicationBase)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Identity.Client.IClientApplicationBase	clientApplicationBase	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Implements

[IClientCrossPlatformTokenCache](#)

Interface IClientCrossPlatformTokenCache

An abstraction of a client's cross platform token cache. Client applications (desktop and mobile apps) should try to get a token from the cache before acquiring a token by another method.

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IClientCrossPlatformTokenCache
```

Methods

RegisterCacheAsync(IClientApplicationBase)

Registers the cache with the client application.

Declaration

```
Task RegisterCacheAsync(IClientApplicationBase clientApplicationBase)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Identity.Client.IClientApplicationBase	clientApplicationBase	The client application.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

See Also

- Microsoft.Identity.Client.Extensions.Msal.MsalCacheHelper
- <https://docs.microsoft.com/en-us/azure/active-directory/develop/msal-net-token-cache-serialization?tabs=desktop>
- <https://github.com/AzureAD/microsoft-authentication-extensions-for-dotnet/wiki/Cross-platform-Token-Cache>
- <https://github.com/AzureAD/microsoft-authentication-extensions-for-dotnet/blob/master/sample/ManualTestApp/Program.cs>

Interface IMsalPublicClientApplicationFactory

An abstraction to create Microsoft.Identity.Client.IClientApplicationBase.

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IMsalPublicClientApplicationFactory
```

Methods

CreateAsync(MsalPublicClientApplicationFactoryContext)

Creates an instance of Microsoft.Identity.Client.IClientApplicationBase asynchronously.

Declaration

```
Task<MsalPublicClientApplicationFactoryResult> CreateAsync(MsalPublicClientApplicationFactoryContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
MsalPublicClientApplicationFactoryContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< MsalPublicClientApplicationFactoryResult >	An instance of Microsoft.Identity.Client.IClientApplicationBase.

Class MsalInteractiveAuthenticationProvider

The `MSAL` authentication provider.

Inheritance

System.Object
MsalInteractiveAuthenticationProvider

Implements

Microsoft.Graph.IAuthenticationProvider

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class MsalInteractiveAuthenticationProvider : IAuthenticationProvider
```

Constructors

MsalInteractiveAuthenticationProvider(IMsalPublicClientApplicationFactory, CliOptions)

Initializes a new instance.

Declaration

```
public MsalInteractiveAuthenticationProvider(IMsalPublicClientApplicationFactory clientApplicationFactory, CliOptions cliOptions)
```

Parameters

TYPE	NAME	DESCRIPTION
IMsalPublicClientApplicationFactory	clientApplicationFactory	
CliOptions	cliOptions	

Methods

AuthenticateRequestAsync(HttpRequestMessage)

Authenticates the request.

Declaration

```
public async Task AuthenticateRequestAsync(HttpRequestMessage request)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Net.Http.HttpRequestMessage	request	The request to authenticate.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

GetTokenAsync()

Gets an `access_token` to run a protected command.

Declaration

```
public async Task<string> GetTokenAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

Implements

Microsoft.Graph.IAuthenticationProvider

Class MsalPublicClientApplicationFactory

The [IMsalPublicClientApplicationFactory](#) to create Microsoft.Identity.Client.IPublicClientApplication.

Inheritance

System.Object
MsalPublicClientApplicationFactory

Implements

[IMsalPublicClientApplicationFactory](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class MsalPublicClientApplicationFactory : IMsalPublicClientApplicationFactory
```

Constructors

MsalPublicClientApplicationFactory(IClientCrossPlatformTokenCache)

Initialize a new instance.

Declaration

```
public MsalPublicClientApplicationFactory(IClientCrossPlatformTokenCache clientCrossPlatformTokenCache)
```

Parameters

TYPE	NAME	DESCRIPTION
IClientCrossPlatformTokenCache	clientCrossPlatformTokenCache	

Methods

CreateAsync(MsalPublicClientApplicationFactoryContext)

Create a new Microsoft.Identity.Client.IPublicClientApplication for authentication.

Declaration

```
public Task<MsalPublicClientApplicationFactoryResult> CreateAsync(MsalPublicClientApplicationFactoryContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
MsalPublicClientApplicationFactoryContext	context	The authentication factory context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< MsalPublicClientApplicationFactoryResult >	

Exceptions

TYPE	CONDITION
System.NotImplementedException	

Implements

[IMsalPublicClientApplicationFactory](#)

Class MsalPublicClientApplicationFactoryContext

The [IMsalPublicClientApplicationFactory](#) context.

Inheritance

System.Object
MsalPublicClientApplicationFactoryContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class MsalPublicClientApplicationFactoryContext
```

Constructors

MsalPublicClientApplicationFactoryContext(AuthenticationOptions)

Initialize a new instance.

Declaration

```
public MsalPublicClientApplicationFactoryContext(AuthenticationOptions authenticationOptions)
```

Parameters

TYPE	NAME	DESCRIPTION
AuthenticationOptions	authenticationOptions	The authentication options.

Properties

AuthenticationOptions

The authentication options.

Declaration

```
public AuthenticationOptions AuthenticationOptions { get; }
```

Property Value

TYPE	DESCRIPTION
AuthenticationOptions	

Class MsalPublicClientApplicationFactoryResult

The [IMsalPublicClientApplicationFactory](#) result.

Inheritance

System.Object
MsalPublicClientApplicationFactoryResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Authentication](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class MsalPublicClientApplicationFactoryResult
```

Constructors

MsalPublicClientApplicationFactoryResult(IClientApplicationBase)

Initialize a new instance.

Declaration

```
public MsalPublicClientApplicationFactoryResult(IClientApplicationBase application)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Identity.Client.IClientApplicationBase	application	

Properties

Application

The client application.

Declaration

```
public IClientApplicationBase Application { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Identity.Client.IClientApplicationBase	

Methods

As<TApp>()

Casts the [Application](#) to the specified Microsoft.Identity.Client.IClientApplicationBase type.

Declaration

```
public TApp As<TApp>()  
    where TApp : IClientApplicationBase
```

Returns

TYPE	DESCRIPTION
TApp	

Type Parameters

NAME	DESCRIPTION
TApp	The client application type.

Namespace PerpetualIntelligence.Cli.Commands

Classes

Argument

The [Argument](#) class is a runtime validated representation of an actual command argument and its value passed by a user or an application.

ArgumentDescriptor

The [ArgumentDescriptor](#) class defines the command argument identity, data type, and data validation behavior. We also refer to arguments as command options or command flags.

ArgumentDescriptors

The ordered [ArgumentDescriptor](#) collection.

Arguments

The ordered [Argument](#) keyed collection.

ArgumentString

An immutable argument string extracted from the [CommandString](#).

ArgumentStrings

The ordered [ArgumentString](#) collection.

Command

An immutable `pi-cli` command. A command is a specific action or a set of actions that a user or an application requests the underlying system to perform. It can be a simple action such as invoking a system method or an OS command or representing a complex operation that calls a set of protected APIs over the internal or external network. A command can virtually do anything in the context of your application or service.

CommandDescriptor

The [CommandDescriptor](#) defines the command identity and its supported arguments that an end-user or an application can use. You can also describe the command behavior, such as whether the command is a root, grouped, or subcommand.

CommandString

An immutable unicode textual form representing the command and its arguments or options that a user or an application wants to execute.

Class Argument

The [Argument](#) class is a runtime validated representation of an actual command argument and its value passed by a user or an application.

Inheritance

System.Object
Argument

Implements

System.IEquatable<[Argument](#)>

Inherited Members

System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class Argument : IEquatable<Argument>
```

Remarks

An argument id is always unique within a command. By design [Argument](#) implements the default equality System.IEquatable<T> and [GetHashCode\(\)](#) using [Id](#) property. Thus, two arguments with the same id are equal irrespective of other property values. This is done to improve performance during lookup and avoid multiple arguments with same identifiers.

The arguments can have same ids across multiple commands. Each [Command](#) has [ArgumentDescriptors](#) collection that contains arguments with unique ids.

Constructors

[Argument](#)(ArgumentDescriptor, Object)

Initialize a new instance..

Declaration

```
public Argument(ArgumentDescriptor argumentDescriptor, object value)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentDescriptor	argumentDescriptor	The argument descriptor.
System.Object	value	The argument value.

[Argument](#)(String, Object, DataType)

Initialize a new instance..

Declaration

```
public Argument(string id, object value, DataType dataType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The argument id.
System.Object	value	The argument value.
System.ComponentModel.DataAnnotations.DataType	dataType	The argument data type.

Argument(String, Object, String)

Initialize a new instance..

Declaration

```
public Argument(string id, object value, string customDataType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The argument id.
System.Object	value	The argument value.
System.String	customDataType	The argument custom data type.

Properties

Alias

The argument alias.

Declaration

```
public string Alias { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

CustomDataType

The argument custom data type.

Declaration

```
[JsonPropertyName("custom_data_type")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public string CustomDataType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

This custom data type is used only if the [DataType](#) property is set to `System.ComponentModel.DataAnnotations.DataType.Custom`.

DataType

The argument data type. Defaults to `System.ComponentModel.DataAnnotations.DataType.Text`.

Declaration

```
[JsonPropertyName("data_type")]
public DataType DataType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.DataType	

Description

The argument description.

Declaration

```
[JsonPropertyName("description")]
public string Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Id

The argument id.

Declaration

```
[JsonPropertyName("id")]
public string Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The argument id is unique with in a command.

Properties

The argument custom properties.

Declaration

```
[JsonPropertyName("properties")]  
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]  
public Dictionary<string, object> Properties { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Object>	

Value

The argument value.

Declaration

```
[JsonPropertyName("value")]  
public object Value { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

Methods

Equals(Argument)

Declaration

```
public bool Equals(Argument other)
```

Parameters

TYPE	NAME	DESCRIPTION
Argument	other	

Returns

TYPE	DESCRIPTION
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

System.Object.Equals(System.Object)

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

TYPE	DESCRIPTION
System.Int32	

Overrides

System.Object.GetHashCode()

Operators

Equality(Argument, Argument)

Declaration

```
public static bool operator ==(Argument left, Argument right)
```

Parameters

TYPE	NAME	DESCRIPTION
Argument	left	
Argument	right	

Returns

TYPE	DESCRIPTION
System.Boolean	

Inequality(Argument, Argument)

Declaration

```
public static bool operator !=(Argument left, Argument right)
```

Parameters

TYPE	NAME	DESCRIPTION
Argument	left	
Argument	right	

Returns

TYPE	DESCRIPTION
System.Boolean	

Implements

System.IEquatable<T>

See Also

[Command](#)

Class ArgumentDescriptor

The [ArgumentDescriptor](#) class defines the command argument identity, data type, and data validation behavior. We also refer to arguments as command options or command flags.

Inheritance

System.Object
ArgumentDescriptor

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class ArgumentDescriptor
```

Constructors

ArgumentDescriptor(String, DataType, Boolean, String, ValidationAttribute[], Object)

Initializes a new instance.

Declaration

```
public ArgumentDescriptor(string id, DataType dataType, bool required = false, string description = null, ValidationAttribute[] validationAttributes = null, object defaultValue = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The argument id.
System.ComponentModel.DataAnnotations.DataType	dataType	The argument data type.
System.Boolean	required	The argument is required.
System.String	description	The argument description.
System.ComponentModel.DataAnnotations.ValidationAttribute[]	validationAttributes	The data validation attributes.
System.Object	defaultValue	The argument default value.

ArgumentDescriptor(String, String, Boolean, String, ValidationAttribute[], Object)

Initializes a new instance.

Declaration

```
public ArgumentDescriptor(string id, string customDataType, bool required = false, string description = null, ValidationAttribute[] validationAttributes = null, object defaultValue = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The argument id.
System.String	customDataType	The argument custom data type.
System.Boolean	required	The argument is required.
System.String	description	The argument description.
System.ComponentModel.DataAnnotations.ValidationAttribute[]	validationAttributes	The data validation attributes.
System.Object	defaultValue	The argument default value.

Properties

Alias

The argument alias.

Declaration

```
public string Alias { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The argument alias is unique within a command. Argument alias supports the legacy apps that identified a command argument with an id and an alias string. For modern console apps, we recommend using just an argument identifier. The core data model is optimized to work with argument id. In general, an app should not identify the same argument with multiple strings. Using alias will degrade the performance.

CustomDataType

The argument custom data type.

Declaration

```
public string CustomDataType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

This custom data type is used only if the [DataType](#) property is set to `System.ComponentModel.DataAnnotations.DataType.Custom`.

DataType

The argument data type.

Declaration

```
public DataType DataType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.DataType	

DefaultValue

The default argument value. `null` means the argument does not support a default value.

Declaration

```
public object DefaultValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

Remarks

If [DefaultValue](#) is set to a non `null` value, then the argument will have [DefaultValue](#), if a user or an app does not specify any value.

Description

The argument description.

Declaration

```
public string Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The argument id is unique across all commands.

Disabled

Determines whether the argument is disabled

Declaration

```
public bool? Disabled { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Id

The argument id.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The argument id is unique within a command.

IsRequired

Determines is the argument is required.

Declaration

```
public bool IsRequired { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Obsolete

Determines whether the argument is obsolete.

Declaration

```
public bool? Obsolete { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Properties

The custom properties.

Declaration

```
public Dictionary<string, object> Properties { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Object>	

ValidationAttributes

The data annotation validation attributes to check the argument value.

Declaration

```
public IEnumerable<ValidationAttribute> ValidationAttributes { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<System.ComponentModel.DataAnnotations.ValidationAttribute>	

See Also

- [Argument](#)
- [CommandString](#)
- [CommandDescriptor](#)

Class ArgumentDescriptors

The ordered [ArgumentDescriptor](#) collection.

Inheritance
System.Object
System.Collections.ObjectModel.Collection< ArgumentDescriptor >
System.Collections.ObjectModel.KeyedCollection<System.String, ArgumentDescriptor >
ArgumentDescriptors
Implements
System.Collections.Generic.IList< ArgumentDescriptor >
System.Collections.Generic ICollection< ArgumentDescriptor >
System.Collections.Generic.IReadOnlyList< ArgumentDescriptor >
System.Collections.Generic.IReadOnlyCollection< ArgumentDescriptor >
System.Collections.Generic.IEnumerable< ArgumentDescriptor >
System.Collections.IList
System.Collections.ICollection
System.Collections.IEnumerable
Inherited Members
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.ChangeItemKey(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor, System.String)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.ClearItems()
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Contains(System.String)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.GetKeyForItem(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.InsertItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Remove(System.String)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.RemoveItem(System.Int32)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.SetItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Comparer
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Dictionary
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Item[System.String]
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.get_Item(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.set_Item(System.Int32, System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Add(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Clear()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.ClearItems()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Contains(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.CopyTo(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor[], System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.GetEnumerator()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.IndexOf(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Insert(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.InsertItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Remove(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.RemoveAt(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.RemoveItem(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.SetItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.ICollection.CopyTo(System.Array, System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IEnumerable.GetEnumerator()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.Add(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.Contains(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.IndexOf(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.Insert(System.Int32, System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.Remove(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Count
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.Items
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.Generic.ICollection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.IsReadOnly
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.ICollection.IsSynchronized
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.ICollection.SyncRoot
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.IsFixedSize
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.IsReadOnly
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.System.Collections.IList.Item[System.Int32]
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()
Namespace: PerpetualIntelligence.Cli.Commands
Assembly: PerpetualIntelligence.Cli.dll
Syntax
<pre>public sealed class ArgumentDescriptors : KeyedCollection<string, ArgumentDescriptor>, IList<ArgumentDescriptor>, ICollection<ArgumentDescriptor>, IReadOnlyList<ArgumentDescriptor>, IReadOnlyCollection<ArgumentDescriptor>, IEnumerable<ArgumentDescriptor>, IList, ICollection, IEnumerable</pre>

Remarks

The argument descriptor collection comparer is System.StringComparer.Ordinal and it determines whether two **ld** strings are equal. Every argument descriptor in the collection must have unique id.

Constructors

ArgumentDescriptors(IStringComparer)

Initializes a new instance.

Declaration

```
public ArgumentDescriptors(IStringComparer stringComparer)
```

Parameters

TYPE	NAME	DESCRIPTION
IStringComparer	stringComparer	

ArgumentDescriptors(IStringComparer, IEnumerable<ArgumentDescriptor>)

Initializes a new instance with the specified argument descriptors.

Declaration

```
public ArgumentDescriptors(IStringComparer stringComparer, IEnumerable<ArgumentDescriptor> collection)
```

Parameters

TYPE	NAME	DESCRIPTION
IStringComparer	stringComparer	The string comparer.
System.Collections.Generic.IEnumerable< ArgumentDescriptor >	collection	The argument descriptors.

Properties

Item[Int32]

Gets or sets an [ArgumentDescriptor](#) instance at the specified index.

Declaration

```
public ArgumentDescriptor this[int index] { get; set; }
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	index	The zero based index.

Property Value

TYPE	DESCRIPTION
ArgumentDescriptor	

Item[String]

Gets an [ArgumentDescriptor](#) instance with the specified id.

Declaration

```
public ArgumentDescriptor this[string id] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The argument id.

Property Value

TYPE	DESCRIPTION
ArgumentDescriptor	ArgumentDescriptor instance if found.

Exceptions

TYPE	CONDITION
ErrorException	If ArgumentDescriptor instance with specified id is not found.

Item[String, Nullable<Boolean>, Nullable<Boolean>]

Gets an [ArgumentDescriptor](#) instance with the specified id.

Declaration

```
public ArgumentDescriptor this[string idOrAlias, bool? alias = null, bool? ifNotThen = null ] { get; }
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	idOrAlias	The argument id or its alias.

TYPE	NAME	DESCRIPTION
System.Nullable<System.Boolean>	alias	<code>true</code> to find the argument by its alias, <code>false</code> to find by its identifier.
System.Nullable<System.Boolean>	ifNotThen	<code>true</code> to find the argument by its id. If not found, then attempt to find it by its alias. <code>false</code> to find by id or alias but not both.

Property Value

TYPE	DESCRIPTION
ArgumentDescriptor	ArgumentDescriptor instance if found.

Remarks

We recommend to use [Item\(String\)](#) to get an argument. Using alias will degrade the application's performance.

Exceptions

TYPE	CONDITION
ErrorException	If ArgumentDescriptor instance with specified id is not found.

StringComparer

The string comparer.

Declaration

```
public IStringComparer StringComparer { get; }
```

Property Value

TYPE	DESCRIPTION
IStringComparer	

Methods

GetKeyForItem(ArgumentDescriptor)

Declaration

```
protected override string GetKeyForItem(ArgumentDescriptor item)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentDescriptor	item	

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.ArgumentDescriptor>.GetKeyForItem(PerpetualIntelligence.Cli.Commands.ArgumentDescriptor)

Implements

- System.Collections.Generic.IList<T>
- System.Collections.Generic.ICollection<T>
- System.Collections.Generic.IReadOnlyList<T>
- System.Collections.Generic.IReadOnlyCollection<T>
- System.Collections.Generic.IEnumerable<T>
- System.Collections.IList
- System.Collections.ICollection
- System.Collections.IEnumerable

Extension Methods

- [IEnumerableExtensions.HasDuplicates<T, TProp>\(IEnumerable<T>, Func<T, TProp>\)](#)
- [IEnumerableExtensions.IsNullOrEmpty<T>\(IEnumerable<T>\)](#)

Class Arguments

The ordered [Argument](#) keyed collection.

Inheritance

System.Object
System.Collections.ObjectModel.Collection<[Argument](#)>
System.Collections.ObjectModel.KeyedCollection<System.String, [Argument](#)>
Arguments

Implements

System.Collections.Generic.ICollection<[Argument](#)>
System.Collections.Generic.ICollection<[Argument](#)>
System.Collections.Generic.IReadOnlyList<[Argument](#)>
System.Collections.Generic.IReadOnlyCollection<[Argument](#)>
System.Collections.Generic.IEnumerable<[Argument](#)>
System.Collections.ICollection
System.Collections.ICollection
System.Collections.IEnumerable

Inherited Members

System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.ChangeItemKey(PerpetualIntelligence.Cli.Commands.Argument, System.String)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.ClearItems()
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.Contains(System.String)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.GetKeyForItem(PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.InsertItem(System.Int32, PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.Remove(System.String)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.RemoveItem(System.Int32)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.SetItem(System.Int32, PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.Comparer
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.Dictionary
System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.Item[System.String]
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.get_Item(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.set_Item(System.Int32, System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Add(PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Clear()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.ClearItems()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Contains(PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.CopyTo(PerpetualIntelligence.Cli.Commands.Argument[], System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.GetEnumerator()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.IndexOf(PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Insert(System.Int32, PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.InsertItem(System.Int32, PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Remove(PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.RemoveAt(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.RemoveItem(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.SetItem(System.Int32, PerpetualIntelligence.Cli.Commands.Argument)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.CopyTo(System.Array, System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.IEnumerable.GetEnumerator()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.Add(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.Contains(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.IndexOf(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.Insert(System.Int32, System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.Remove(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Count
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Item[System.Int32]
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.Items
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.Generic.ICollection<PerpetualIntelligence.Cli.Commands.Argument>.IsReadOnly
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.IsSynchronized
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.SyncRoot
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.IsFixedSize
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.IsReadOnly
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.Argument>.System.Collections.ICollection.Item[System.Int32]
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class Arguments : KeyedCollection<string, Argument>, ICollection<Argument>, IReadOnlyList<Argument>, IReadOnlyCollection<Argument>, IEnumerable<Argument>, IList, ICollection, IEnumerable
```

Remarks

The argument collection comparer is System.StringComparer.Ordinal and it determines whether two **Id** strings are equal. Every argument in the collection must have unique id.

Constructors

Arguments(IStringComparer)

Initializes a new instance.

Declaration

```
public Arguments(IStringComparer stringComparer)
```

Parameters

TYPE	NAME	DESCRIPTION
IStringComparer	stringComparer	

Properties

StringComparer

The string comparer.

Declaration

```
public IStringComparer StringComparer { get; }
```

Property Value

TYPE	DESCRIPTION
IStringComparer	

Methods

GetKeyForItem(Argument)

Declaration

```
protected override string GetKeyForItem(Argument item)
```

Parameters

TYPE	NAME	DESCRIPTION
Argument	item	

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Collections.ObjectModel.KeyedCollection<System.String, PerpetualIntelligence.Cli.Commands.Argument>.GetKeyForItem(PerpetualIntelligence.Cli.Commands.Argument)

GetValue<TValue>(String)

Gets the argument value by its id.

Declaration

```
public TValue GetValue<TValue>(string argId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	argId	The argument identifier or the alias.

Returns

TYPE	DESCRIPTION
TValue	

Type Parameters

NAME	DESCRIPTION
TValue	

GetValue<TValue>(String, Nullable<Boolean>)

Gets the argument value by its id or alias.

Declaration

```
public TValue GetValue<TValue>(string argIdOrAlias, bool? alias = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	argIdOrAlias	The argument identifier or the alias.
System.Nullable<System.Boolean>	alias	<code>true</code> to find the argument by alias, <code>false</code> to find by its identifier.

Returns

TYPE	DESCRIPTION
TValue	

Type Parameters

NAME	DESCRIPTION
TValue	The value type.

Remarks

We recommend to use `GetValue<TValue>(String)` to get an argument value. Using alias will degrade the application's performance.

Implements

- System.Collections.Generic.IList<T>
- System.Collections.Generic ICollection<T>
- System.Collections.Generic.IReadOnlyList<T>
- System.Collections.Generic.IReadOnlyCollection<T>
- System.Collections.Generic.IEnumerable<T>
- System.Collections.IList
- System.Collections.ICollection
- System.Collections.IEnumerable

Extension Methods

- [IEnumerableExtensions.HasDuplicates<T, TProp>\(IEnumerable<T>, Func<T, TProp>\)](#)
- [IEnumerableExtensions.IsNullOrEmpty<T>\(IEnumerable<T>\)](#)

Class ArgumentString

An immutable argument string extracted from the [CommandString](#).

Inheritance

System.Object
ArgumentString

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [PerpetualIntelligence.Cli.Commands](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class ArgumentString
```

Constructors

ArgumentString(String)

Initialize a new instance.

Declaration

```
public ArgumentString(string raw)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	raw	The raw argument string.

ArgumentString(String, Boolean, Int32)

Initialize a new instance.

Declaration

```
public ArgumentString(string raw, bool aliasPrefix, int position)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	raw	The raw argument string.
System.Boolean	aliasPrefix	<code>true</code> if the argument string has an alias prefix, otherwise <code>false</code> .

TYPE	NAME	DESCRIPTION
System.Int32	position	The zero based position or index of the argument string with in a command string.

Properties

AliasPrefix

true if the argument string has an alias prefix, otherwise false.

Declaration

```
public bool AliasPrefix { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Position

The zero based position or index of the argument string with in a command string. ///

Declaration

```
public int Position { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Raw

The argument string.

Declaration

```
public string Raw { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ToString()

The string representation of [ArgumentString](#).

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Class ArgumentStrings

The ordered [ArgumentString](#) collection.

Inheritance
System.Object
System.Collections.ObjectModel.Collection< ArgumentString >
System.Collections.ObjectModel.KeyedCollection<System.Int32, ArgumentString >
ArgumentStrings
Implements
System.Collections.Generic.IList< ArgumentString >
System.Collections.Generic.ICollection< ArgumentString >
System.Collections.Generic.IReadOnlyList< ArgumentString >
System.Collections.Generic.IReadOnlyCollection< ArgumentString >
System.Collections.Generic.IEnumerable< ArgumentString >
System.Collections.IList
System.Collections.ICollection
System.Collections.IEnumerable
Inherited Members
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.ChangeItemKey(PerpetualIntelligence.Cli.Commands.ArgumentString, System.Int32)
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.ClearItems()
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.Contains(System.Int32)
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.GetKeyForItem(PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.InsertItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.Remove(System.Int32)
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.RemoveItem(System.Int32)
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.SetItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.Comparer
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.Dictionary
System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.Item[System.Int32]
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.get_Item(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.set_Item(System.Int32, System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Add(PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Clear()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.ClearItems()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Contains(PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.CopyTo(PerpetualIntelligence.Cli.Commands.ArgumentString[], System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.GetEnumerator()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.IndexOf(PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Insert(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.InsertItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Remove(PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.RemoveAt(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.RemoveItem(System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.SetItem(System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.ICollection.CopyTo(System.Array, System.Int32)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IEnumerable.GetEnumerator()
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.Add(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.Contains(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.IndexOf(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.Insert(System.Int32, System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.Remove(System.Object)
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Count
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Item[System.Int32]
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.Items
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.Generic.ICollection<PerpetualIntelligence.Cli.Commands.ArgumentString>.IsReadOnly
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.ICollection.IsSynchronized
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.ICollection.SyncRoot
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.IsFixedSize
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.IsReadOnly
System.Collections.ObjectModel.Collection<PerpetualIntelligence.Cli.Commands.ArgumentString>.System.Collections.IList.Item[System.Int32]
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()
Namespace: PerpetualIntelligence.Cli.Commands
Assembly: PerpetualIntelligence.Cli.dll
Syntax
<pre>public sealed class ArgumentStrings : KeyedCollection<int, ArgumentString>, IList<ArgumentString>, ICollection<ArgumentString>, IReadOnlyList<ArgumentString>, IReadOnlyCollection<ArgumentString>, IEnumerable<ArgumentString>, IList, ICollection, IEnumerable</pre>

Remarks

The [ArgumentStrings](#) collection determines the [Position](#) of each [ArgumentString](#) with in a [CommandString](#).

Constructors

ArgumentStrings()

Initializes a new instance.

Declaration

```
public ArgumentStrings()
```

Methods

GetKeyForItem(ArgumentString)

Gets the [Position](#) as a key for the item.

Declaration

```
protected override int GetKeyForItem(ArgumentString item)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentString	item	The item to check.

Returns

TYPE	DESCRIPTION
System.Int32	The Position as a key for the item.

Overrides

System.Collections.ObjectModel.KeyedCollection<System.Int32, PerpetualIntelligence.Cli.Commands.ArgumentString>.GetKeyForItem(PerpetualIntelligence.Cli.Commands.ArgumentString)

Implements

- System.Collections.Generic.IList<T>
- System.Collections.Generic ICollection<T>
- System.Collections.Generic.IReadOnlyList<T>
- System.Collections.Generic.IReadOnlyCollection<T>
- System.Collections.Generic.IEnumerable<T>
- System.Collections.IList
- System.Collections.ICollection
- System.Collections.IEnumerable

Extension Methods

- [IEnumerableExtensions.HasDuplicates<T, TProp>\(IEnumerable<T>, Func<T, TProp>\)](#)
- [IEnumerableExtensions.IsNullOrEmpty<T>\(IEnumerable<T>\)](#)

Class Command

An immutable `pi-cli` command. A command is a specific action or a set of actions that a user or an application requests the underlying system to perform. It can be a simple action such as invoking a system method or an OS command or representing a complex operation that calls a set of protected APIs over the internal or external network. A command can virtually do anything in the context of your application or service.

Inheritance

System.Object
Command

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class Command
```

Constructors

Command(CommandDescriptor, Arguments, Dictionary<String, Object>)

Initializes a new instance.

Declaration

```
public Command(CommandDescriptor commandDescriptor, Arguments arguments = null, Dictionary<string, object> properties = null)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandDescriptor	commandDescriptor	
Arguments	arguments	
System.Collections.Generic.Dictionary<System.String, System.Object>	properties	

Command(String, String, String, Arguments, Dictionary<String, Object>)

Initialize a new instance.

Declaration

```
public Command(string id, string name, string description, Arguments arguments = null, Dictionary<string, object> properties = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The command id.
System.String	name	The command name.
System.String	description	The command description.
Arguments	arguments	The command arguments.
System.Collections.Generic.Dictionary<System.String, System.Object>	properties	The command properties.

Properties

Arguments

The command arguments.

Declaration

```
public Arguments Arguments { get; }
```

Property Value

TYPE	DESCRIPTION
Arguments	

Description

The command description.

Declaration

```
public string Description { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Id

The command id unique.

Declaration

```
public string Id { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Name

The command name.

Declaration

```
public string Name { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Properties

The command custom properties.

Declaration

```
public Dictionary<string, object> Properties { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Object>	

Methods

GetOptionalArgumentValue<TValue>(String)

Gets the optional argument value for the specified identifier.

Declaration

```
public TValue GetOptionalArgumentValue<TValue>(string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	

Returns

TYPE	DESCRIPTION
TValue	The optional argument value.

Type Parameters

NAME	DESCRIPTION
TValue	The type of value.

GetRequiredArgumentValue<TValue>(String)

Gets the required argument value for the specified identifier.

Declaration

```
public TValue GetRequiredArgumentValue<TValue>(string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	

Returns

TYPE	DESCRIPTION
TValue	The argument value.

Type Parameters

NAME	DESCRIPTION
TValue	The type of value.

Exceptions

TYPE	CONDITION
ErrorException	If the argument is not supported.

TryGetArgument(String, out Argument)

Attempts to find an argument.

Declaration

```
[WriteUnitTest]  
public bool TryGetArgument(string id, out Argument argument)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The argument identifier.

TYPE	NAME	DESCRIPTION
Argument	argument	The argument if found in the collection.

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if an argument exist in the collection, otherwise <code>false</code> .

Extension Methods

[CommandExtensions.ReadAnswerAsync\(Command, String, String\[\]\)](#)

See Also

[CommandDescriptor](#)

[Argument](#)

[Arguments](#)

Class CommandDescriptor

The [CommandDescriptor](#) defines the command identity and its supported arguments that an end-user or an application can use. You can also describe the command behavior, such as whether the command is a root, grouped, or subcommand.

Inheritance

System.Object
CommandDescriptor

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class CommandDescriptor
```

Constructors

CommandDescriptor(String, String, String, String, ArgumentDescriptors, Dictionary<String, Object>, String)

Initializes a new instance.

Declaration

```
public CommandDescriptor(string id, string name, string prefix, string description, ArgumentDescriptors argumentDescriptors = null, Dictionary<string, object> properties = null, string defaultArgument = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The command id.
System.String	name	The command name.
System.String	prefix	The command prefix to map the command string.
System.String	description	The command description.
ArgumentDescriptors	argumentDescriptors	The command argument descriptors.

TYPE	NAME	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Object>	properties	The custom properties.
System.String	defaultArgument	The default argument.

Properties

ArgumentDescriptors

The command argument descriptors.

Declaration

```
public ArgumentDescriptors ArgumentDescriptors { get; }
```

Property Value

TYPE	DESCRIPTION
ArgumentDescriptors	

DefaultArgument

The default argument. `null` means the command does not support a default argument.

Declaration

```
public string DefaultArgument { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

[DefaultArgument](#) is not the default argument value (see [DefaultValue](#)), it is the default argument identifier (see [Id](#)) whose value is populated automatically based on the [CommandString](#). If [DefaultArgument](#) is set to a non `null` value, then the [ICommandExtractor](#) will attempt to extract the value from the [CommandString](#) and put it in an [Argument](#) identified by [DefaultArgument](#).

See Also

[DefaultValue](#)

Description

The command description.

Declaration

```
public string Description { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Id

The command id.

Declaration

```
public string Id { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The command id is unique across all command group.

IsGroup

Returns `true` if this descriptor represents a command group; otherwise, `false`.

Declaration

```
public bool IsGroup { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsProtected

Returns `true` if this descriptor represents a protected command; otherwise, `false`.

Declaration

```
public bool IsProtected { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

IsRoot

Returns `true` if this descriptor represents a command root; otherwise, `false`.

Declaration

```
public bool IsRoot { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Name

The command name.

Declaration

```
public string Name { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The command name is unique within a command group.

Prefix

The prefix to match the command string.

Declaration

```
public string Prefix { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Properties

The custom properties.

Declaration

```
public Dictionary<string, object> Properties { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Object>	

Methods

TryGetArgumentDescriptor(String, out ArgumentDescriptor)

Attempts to find an argument descriptor.

Declaration

```
public bool TryGetArgumentDescriptor(string argId, out ArgumentDescriptor argumentDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	argId	The argument descriptor identifier.
ArgumentDescriptor	argumentDescriptor	The argument descriptor if found.

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if an argument descriptor exist in the collection, otherwise <code>false</code> .

See Also

[Command](#)

[ArgumentDescriptor](#)

[AddDescriptor<TRunner, TChecker>](#)([ICliBuilder](#), [CommandDescriptor](#), System.Boolean, System.Boolean, System.Boolean)

Class CommandString

An immutable unicode textual form representing the command and its arguments or options that a user or an application wants to execute.

Inheritance

System.Object
CommandString

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class CommandString
```

Constructors

CommandString(String)

Initialize a new instance.

Declaration

```
[JsonConstructor]  
public CommandString(string raw)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	raw	

Properties

Raw

The command string raw value.

Declaration

```
[JsonPropertyName("raw")]  
public string Raw { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Cli.Commands.Checkers

Classes

[ArgumentChecker](#)

The argument checker.

[ArgumentCheckerContext](#)

The argument checker context.

[ArgumentCheckerResult](#)

The argument checker result.

[CommandChecker](#)

The command checker.

[CommandCheckerContext](#)

The command checker context.

[CommandCheckerResult](#)

The command checker result.

Interfaces

[IArgumentChecker](#)

An abstraction to check an [Argument](#).

[ICommandChecker](#)

An abstraction to check a [Command](#).

Class ArgumentChecker

The argument checker.

Inheritance

System.Object
ArgumentChecker

Implements

[IArgumentChecker](#)
[IChecker](#)<[ArgumentCheckerContext](#), [ArgumentCheckerResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ArgumentChecker : IArgumentChecker, IChecker<ArgumentCheckerContext, ArgumentCheckerResult>
```

Remarks

The [ArgumentChecker](#) uses the System.ComponentModel.DataAnnotations.ValidationAttribute to check an argument value.

Constructors

ArgumentChecker(IArgumentDataTypeMapper, CliOptions, ILogger<ArgumentChecker>)

Initialize a new instance.

Declaration

```
public ArgumentChecker(IArgumentDataTypeMapper mapper, CliOptions options, ILogger<ArgumentChecker> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
IArgumentDataTypeMapper	mapper	The argument data-type mapper.
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< ArgumentChecker >	logger	The logger.

Methods

CheckAsync(ArgumentCheckerContext)

Checks the context asynchronously.

Declaration

```
public async Task<ArgumentCheckerResult> CheckAsync(ArgumentCheckerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentCheckerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< ArgumentCheckerResult >	The checker result.

StrictTypeCheckingAsync(ArgumentCheckerContext, ArgumentDataTypeMapperResult)

Checks the argument value compatibility.

Declaration

```
protected Task<ArgumentCheckerResult> StrictTypeCheckingAsync(ArgumentCheckerContext context, ArgumentDataTypeMapperResult mapperResult)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentCheckerContext	context	
ArgumentDataTypeMapperResult	mapperResult	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< ArgumentCheckerResult >	

Implements

- [IArgumentChecker](#)
- [IChecker<TContext, TResult>](#)

Class ArgumentCheckerContext

The argument checker context.

Inheritance

System.Object
ArgumentCheckerContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ArgumentCheckerContext
```

Constructors

ArgumentCheckerContext(ArgumentDescriptor, Argument)

Initialize a new instance.

Declaration

```
public ArgumentCheckerContext(ArgumentDescriptor argumentDescriptor, Argument argument)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentDescriptor	argumentDescriptor	The argument descriptor.
Argument	argument	The argument.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

Argument

The argument to check.

Declaration

```
public Argument Argument { get; set; }
```

Property Value

TYPE	DESCRIPTION
Argument	

ArgumentDescriptor

The argument descriptor.

Declaration

```
public ArgumentDescriptor ArgumentDescriptor { get; }
```

Property Value

TYPE	DESCRIPTION
ArgumentDescriptor	

See Also

- [IArgumentChecker](#)
- [ArgumentCheckerResult](#)

Class ArgumentCheckerResult

The argument checker result.

Inheritance

System.Object
ArgumentCheckerResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ArgumentCheckerResult
```

Constructors

ArgumentCheckerResult(Type)

Initialize a new instance.

Declaration

```
public ArgumentCheckerResult(Type mappedType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	mappedType	The mapped type to validate the argument.

Properties

MappedType

The mapped system type.

Declaration

```
public Type MappedType { get; }
```

Property Value

TYPE	DESCRIPTION
System.Type	

See Also

[IArgumentChecker](#)
[ArgumentCheckerContext](#)

Class CommandChecker

The command checker.

Inheritance

System.Object
CommandChecker

Implements

[ICommandChecker](#)
[IChecker](#)<[CommandCheckerContext](#), [CommandCheckerResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandChecker : ICommandChecker, IChecker<CommandCheckerContext, CommandCheckerResult>
```

Constructors

CommandChecker(IArgumentChecker, CliOptions, ILogger<CommandChecker>)

Initialize a new instance.

Declaration

```
public CommandChecker(IArgumentChecker argumentChecker, CliOptions options, ILogger<CommandChecker> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
IArgumentChecker	argumentChecker	The argument checker.
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< CommandChecker >	logger	The logger.

Methods

CheckAsync(CommandCheckerContext)

Checks the context asynchronously.

Declaration

```
public virtual async Task<CommandCheckerResult> CheckAsync(CommandCheckerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandCheckerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandCheckerResult >	The checker result.

Implements

- [ICommandChecker](#)
- [IChecker<TContext, TResult>](#)

Class CommandCheckerContext

The command checker context.

Inheritance

System.Object
CommandCheckerContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandCheckerContext
```

Constructors

CommandCheckerContext(CommandDescriptor, Command)

Initialize a new instance.

Declaration

```
public CommandCheckerContext(CommandDescriptor commandDescriptor, Command command)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandDescriptor	commandDescriptor	The command descriptor.
Command	command	The command to check.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

Command

The command to check.

Declaration

```
public Command Command { get; set; }
```

Property Value

TYPE	DESCRIPTION
Command	

CommandDescriptor

The command descriptor.

Declaration

```
public CommandDescriptor CommandDescriptor { get; }
```

Property Value

TYPE	DESCRIPTION
CommandDescriptor	

Class CommandCheckerResult

The command checker result.

Inheritance

System.Object

CommandCheckerResult

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandCheckerResult
```

Interface IArgumentChecker

An abstraction to check an [Argument](#).

Inherited Members

[IChecker<ArgumentCheckerContext, ArgumentCheckerResult>.CheckAsync\(ArgumentCheckerContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IArgumentChecker : IChecker<ArgumentCheckerContext, ArgumentCheckerResult>
```

Interface ICommandChecker

An abstraction to check a [Command](#).

Inherited Members

[IChecker<CommandCheckerContext, CommandCheckerResult>.CheckAsync\(CommandCheckerContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Checkers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface ICommandChecker : IChecker<CommandCheckerContext, CommandCheckerResult>
```

Namespace PerpetualIntelligence.Cli.Commands.Comparers

Classes

[StringComparisonComparer](#)

The default System.StringComparison based [IStringComparer](#) to compare the `pi-cli` command strings, argument strings and argument values.

Class StringComparisonComparer

The default System.StringComparison based [IStringComparer](#) to compare the `pi-cli` command strings, argument strings and argument values.

Inheritance

System.Object
StringComparisonComparer

Implements

[IStringComparer](#)
System.Collections.Generic.IEqualityComparer<System.String>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Comparers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class StringComparisonComparer : IStringComparer, IEqualityComparer<string>
```

Constructors

StringComparisonComparer(StringComparison)

Initialize a new instance.

Declaration

```
public StringComparisonComparer(StringComparison comparison)
```

Parameters

TYPE	NAME	DESCRIPTION
System.StringComparison	comparison	

Properties

Comparison

Gets the System.StringComparison value to compare the two strings.

Declaration

```
public StringComparison Comparison { get; }
```

Property Value

TYPE	DESCRIPTION
System.StringComparison	

Methods

Equals(String, String)

Determines whether the two System.String objects are equal.

Declaration

```
public bool Equals(string x, string y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	x	The first string to compare.
System.String	y	The second string to compare.

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the two strings are equal; otherwise, <code>false</code> .

GetHashCode(String)

Returns a hash code for the string.

Declaration

```
public int GetHashCode(string obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	obj	The string.

Returns

TYPE	DESCRIPTION
System.Int32	The hash code.

New(StringComparison)

Creates a new instance with specified System.StringComparison.

Declaration

```
public static IStringComparer New(StringComparison comparison)
```

Parameters

TYPE	NAME	DESCRIPTION
System.StringComparison	comparison	The System.StringComparison to use.

Returns

TYPE	DESCRIPTION
IStringComparer	A new StringComparisonComparer instance.

Remarks

[New\(StringComparison\)](#) method is useful to create a new instance where dependency injection is not available to inject the registered [IStringComparer](#).

Implements

[IStringComparer](#)
System.Collections.Generic.IEqualityComparer<T>

Namespace PerpetualIntelligence.Cli.Commands.Extractors

Classes

[ArgumentExtractor](#)

The default [IArgumentExtractor](#).

[ArgumentExtractorContext](#)

The argument extractor context.

[ArgumentExtractorResult](#)

The argument extractor result.

[CommandExtractor](#)

The default [ICommandExtractor](#).

[CommandExtractorContext](#)

The command extractor context.

[CommandExtractorResult](#)

The command extractor result.

Interfaces

[IArgumentExtractor](#)

An abstraction to extract an argument.

[ICommandExtractor](#)

An abstraction to extract a command.

Class ArgumentExtractor

The default [IArgumentExtractor](#).

Inheritance

System.Object
ArgumentExtractor

Implements

[IArgumentExtractor](#)
[IExtractor](#)<[ArgumentExtractorContext](#), [ArgumentExtractorResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
[WriteDocumentation]
public class ArgumentExtractor : IArgumentExtractor, IExtractor<ArgumentExtractorContext,
ArgumentExtractorResult>
```

Remarks

The syntax for a separator based argument is {arg}={value} for e.g. name=oneimlx. The syntax has 4 parts:

- 1. - is an argument prefix. You can configure it via [ArgumentPrefix](#)
- 2. {arg} is an argument id. For e.g. name
- 3. = is an argument separator. You can configure it via [ArgumentSeparator](#)
- 4. {value} is an argument value. For e.g. oneimlx

Constructors

ArgumentExtractor(IStringComparer, CliOptions, ILogger<ArgumentExtractor>)

Initialize a new instance.

Declaration

```
public ArgumentExtractor(IStringComparer stringComparer, CliOptions options, ILogger<ArgumentExtractor>
logger)
```

Parameters

TYPE	NAME	DESCRIPTION
IStringComparer	stringComparer	The string comparer.
CliOptions	options	The configuration options.

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger< ArgumentExtractor >	logger	The logger.

Properties

ArgumentAliasNoValueRegexPattern

Gets the REGEX pattern to match the argument alias with no value.

Declaration

```
protected virtual string ArgumentAliasNoValueRegexPattern { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

Lets assume that you have configured [Separator](#) as a single whitespace string ' ', and [ArgumentAliasPrefix](#) as a single dash character string '-'.

The default implementation for [ArgumentAliasNoValueRegexPattern](#) will match using the following criteria:

- 1. '^ *(-)(.+)\$'Default example REGEX pattern
- 2. '^'Matches the beginning of the string
- 3. '* 'Matches 0 or more [Separator](#)
- 4. '(-)+'Create a new capture group and matches 1 or more [ArgumentAliasPrefix](#)
- 5. '(.+?)'Create a new capture group and matches characters (1 or more) except line breaks
- 6. '\$'Matches the end of the string

ArgumentAliasValueRegexPattern

Gets the REGEX pattern to match the argument id and value using [ArgumentPrefix](#).

Declaration

```
protected virtual string ArgumentAliasValueRegexPattern { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ArgumentIdNoValueRegexPattern

Gets the REGEX pattern to match the argument alias and value using [ArgumentAliasPrefix](#).

Declaration

```
protected virtual string ArgumentIdNoValueRegexPattern { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ArgumentIdValueRegexPattern

Gets the REGEX pattern to match the argument alias and value using [ArgumentAliasPrefix](#).

Declaration

```
protected virtual string ArgumentIdValueRegexPattern { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ArgumentValueWithinRegexPattern

Gets the REGEX pattern to match the argument alias and value using [ArgumentValueWithin](#).

Declaration

```
protected virtual string ArgumentValueWithinRegexPattern { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

ExtractAsync(ArgumentExtractorContext)

Extracts from a context asynchronously.

Declaration

```
public Task<ArgumentExtractorResult> ExtractAsync(ArgumentExtractorContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentExtractorContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< ArgumentExtractorResult >	The extractor result.

Implements

- [IArgumentExtractor](#)
- [IExtractor<TContext, TResult>](#)

See Also

[ArgumentPrefix](#)

[ArgumentSeparator](#)

Class ArgumentExtractorContext

The argument extractor context.

Inheritance

System.Object
ArgumentExtractorContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ArgumentExtractorContext
```

Constructors

ArgumentExtractorContext(ArgumentString, CommandDescriptor)

Initialize a new instance.

Declaration

```
public ArgumentExtractorContext(ArgumentString argumentString, CommandDescriptor commandDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentString	argumentString	The argument string.
CommandDescriptor	commandDescriptor	The command descriptor.

Properties

ArgumentString

The argument string.

Declaration

```
public ArgumentString ArgumentString { get; set; }
```

Property Value

TYPE	DESCRIPTION
ArgumentString	

CommandDescriptor

The command descriptor.

Declaration

```
public CommandDescriptor CommandDescriptor { get; set; }
```

Property Value

TYPE	DESCRIPTION
CommandDescriptor	

Class ArgumentExtractorResult

The argument extractor result.

Inheritance

System.Object
ArgumentExtractorResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ArgumentExtractorResult
```

Constructors

ArgumentExtractorResult(Argument)

Initialize a new instance.

Declaration

```
public ArgumentExtractorResult(Argument argument)
```

Parameters

TYPE	NAME	DESCRIPTION
Argument	argument	

Properties

Argument

The extracted argument.

Declaration

```
public Argument Argument { get; set; }
```

Property Value

TYPE	DESCRIPTION
Argument	

Class CommandExtractor

The default [ICommandExtractor](#).

Inheritance

System.Object
CommandExtractor

Implements

[ICommandExtractor](#)
[IExtractor](#)<[CommandExtractorContext](#), [CommandExtractorResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandExtractor : ICommandExtractor, IExtractor<CommandExtractorContext, CommandExtractorResult>
```

Constructors

CommandExtractor(ICommandDescriptorStore, IArgumentExtractor, IStringComparer, CliOptions, ILogger<CommandExtractor>, IDefaultArgumentProvider, IDefaultArgumentValueProvider)

Initialize a new instance.

Declaration

```
public CommandExtractor(ICommandDescriptorStore commandStore, IArgumentExtractor argumentExtractor, IStringComparer stringComparer, CliOptions options, ILogger<CommandExtractor> logger, IDefaultArgumentProvider defaultArgumentProvider = null, IDefaultArgumentValueProvider defaultArgumentValueProvider = null)
```

Parameters

TYPE	NAME	DESCRIPTION
ICommandDescriptorStore	commandStore	The command descriptor store.
IArgumentExtractor	argumentExtractor	The argument extractor.
IStringComparer	stringComparer	The string comparer.
CliOptions	options	The configuration options.

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger< CommandExtractor >	logger	The logger.
IDefaultArgumentProvider	defaultArgumentProvider	The optional default argument provider.
IDefaultArgumentValueProvider	defaultArgumentValueProvider	The optional argument default value provider.

Methods

ExtractAsync(CommandExtractorContext)

Extracts from a context asynchronously.

Declaration

```
public async Task<CommandExtractorResult> ExtractAsync(CommandExtractorContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandExtractorContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandExtractorResult >	The extractor result.

Implements

- [ICommandExtractor](#)
- [IExtractor<TContext, TResult>](#)

See Also

[Separator](#)

Class CommandExtractorContext

The command extractor context.

Inheritance

System.Object
CommandExtractorContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandExtractorContext
```

Constructors

CommandExtractorContext(CommandString)

Initialize a new instance.

Declaration

```
public CommandExtractorContext(CommandString commandString)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandString	commandString	The command string.

Properties

CommandString

The command string.

Declaration

```
public CommandString CommandString { get; set; }
```

Property Value

TYPE	DESCRIPTION
CommandString	

Class CommandExtractorResult

The command extractor result.

Inheritance

System.Object
CommandExtractorResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandExtractorResult
```

Constructors

CommandExtractorResult(Command, CommandDescriptor)

Initialize a new instance.

Declaration

```
public CommandExtractorResult(Command command, CommandDescriptor commandDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
Command	command	The extracted command.
CommandDescriptor	commandDescriptor	The extracted command descriptor.

Properties

Command

The extracted command.

Declaration

```
public Command Command { get; }
```

Property Value

TYPE	DESCRIPTION
Command	

CommandDescriptor

The extracted command descriptor.

Declaration

```
public CommandDescriptor CommandDescriptor { get; }
```

Property Value

TYPE	DESCRIPTION
CommandDescriptor	

Interface IArgumentExtractor

An abstraction to extract an argument.

Inherited Members

[IExtractor<ArgumentExtractorContext, ArgumentExtractorResult>.ExtractAsync\(ArgumentExtractorContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IArgumentExtractor : IExtractor<ArgumentExtractorContext, ArgumentExtractorResult>
```

Interface ICommandExtractor

An abstraction to extract a command.

Inherited Members

[IExtractor<CommandExtractorContext, CommandExtractorResult>.ExtractAsync\(CommandExtractorContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Extractors](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface ICommandExtractor : IExtractor<CommandExtractorContext, CommandExtractorResult>
```

Namespace PerpetualIntelligence.Cli.Commands.Handlers

Classes

[CommandHandler](#)

The default handler to handle a `pi-cli` command request routed from a [CommandRouter](#).

[CommandHandlerContext](#)

The command handler context.

[CommandHandlerResult](#)

The command handler result.

Interfaces

[ICommandHandler](#)

An abstraction to handle a `pi-cli` command request routed from a [ICommandRouter](#).

Class CommandHandler

The default handler to handle a `pi-cli` command request routed from a [CommandRouter](#).

Inheritance

System.Object
CommandHandler

Implements

[ICommandHandler](#)
[IHandler](#)<[CommandHandlerContext](#), [CommandHandlerResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Handlers](#)

Assembly: [PerpetualIntelligence.Cli.dll](#)

Syntax

```
public sealed class CommandHandler : ICommandHandler, IHandler<CommandHandlerContext, CommandHandlerResult>
```

Constructors

CommandHandler(IServiceProvider, ILicenseChecker, CliOptions, ILogger<CommandHandler>)

Initialize a news instance.

Declaration

```
public CommandHandler(IServiceProvider services, ILicenseChecker licenseChecker, CliOptions options, ILogger<CommandHandler> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IServiceProvider	services	
ILicenseChecker	licenseChecker	
CliOptions	options	
Microsoft.Extensions.Logging.ILogger< CommandHandler >	logger	

Methods

HandleAsync(CommandHandlerContext)

Handles the context.

Declaration

```
public async Task<CommandHandlerResult> HandleAsync(CommandHandlerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandHandlerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandHandlerResult >	The handler result.

Implements

- [ICommandHandler](#)
- [IHandler<TContext, TResult>](#)

Class CommandHandlerContext

The command handler context.

Inheritance

System.Object
CommandHandlerContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Handlers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandHandlerContext
```

Constructors

CommandHandlerContext(CommandDescriptor, Command, License)

Initialize a new instance.

Declaration

```
public CommandHandlerContext(CommandDescriptor commandDescriptor, Command command, License license)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandDescriptor	commandDescriptor	The command descriptor to handle.
Command	command	The command handle.
License	license	The extracted license.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

Command

The command to handle.

Declaration

```
public Command Command { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
Command	

CommandDescriptor

The command descriptor to handle.

Declaration

```
public CommandDescriptor CommandDescriptor { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
CommandDescriptor	

License

The extracted licenses.

Declaration

```
public License License { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
License	

Class CommandHandlerResult

The command handler result.

Inheritance

System.Object

CommandHandlerResult

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Handlers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandHandlerResult
```


Interface ICommandHandler

An abstraction to handle a `pi-cli` command request routed from a [ICommandRouter](#).

Inherited Members

[IHandler<CommandHandlerContext, CommandHandlerResult>](#).HandleAsync(CommandHandlerContext)

Namespace: [PerpetualIntelligence.Cli.Commands.Handlers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface ICommandHandler : IHandler<CommandHandlerContext, CommandHandlerResult>
```

Namespace PerpetualIntelligence.Cli.Commands.Mappers

Classes

[ArgumentDataTypeMapperContext](#)

The argument data-type mapper context.

[ArgumentDataTypeMapperResult](#)

The argument data-type mapper result.

[DataAnnotationsArgumentDataTypeMapper](#)

The argument data type mapper using System.ComponentModel.DataAnnotations.

Interfaces

[IArgumentDataTypeMapper](#)

An abstraction to map an [DataType](#) to System.Type.

Class ArgumentDataTypeMapperContext

The argument data-type mapper context.

Inheritance

System.Object
ArgumentDataTypeMapperContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Mappers](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ArgumentDataTypeMapperContext
```

Constructors

ArgumentDataTypeMapperContext(Argument)

Initialize a new instance.

Declaration

```
public ArgumentDataTypeMapperContext(Argument argument)
```

Parameters

TYPE	NAME	DESCRIPTION
Argument	argument	The argument.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

Argument

The argument to map.

Declaration

```
public Argument Argument { get; set; }
```

Property Value

TYPE	DESCRIPTION
Argument	

Class ArgumentDataTypeMapperResult

The argument data-type mapper result.

Inheritance

System.Object
ArgumentDataTypeMapperResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Mappers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ArgumentDataTypeMapperResult
```

Constructors

ArgumentDataTypeMapperResult(Type)

Initialize a new instance.

Declaration

```
public ArgumentDataTypeMapperResult(Type mappedType)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	mappedType	The mapped type.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	Null mapped type.

Properties

MappedType

The mapped system type.

Declaration

```
public Type MappedType { get; }
```

Property Value

TYPE	DESCRIPTION
System.Type	

Class DataAnnotationsArgumentDataTypeMapper

The argument data type mapper using System.ComponentModel.DataAnnotations.

Inheritance

System.Object
DataAnnotationsArgumentDataTypeMapper

Implements

IArgumentDataTypeMapper
IMapper<ArgumentDataTypeMapperContext, ArgumentDataTypeMapperResult>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Mappers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class DataAnnotationsArgumentDataTypeMapper : IArgumentDataTypeMapper,
IMapper<ArgumentDataTypeMapperContext, ArgumentDataTypeMapperResult>
```

Constructors

DataAnnotationsArgumentDataTypeMapper(CliOptions, ILogger<DataAnnotationsArgumentDataTypeMapper>)

Initialize a new instance.

Declaration

```
public DataAnnotationsArgumentDataTypeMapper(CliOptions options,
ILogger<DataAnnotationsArgumentDataTypeMapper> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< DataAnnotationsArgumentDataTypeMapper >	logger	The logger.

Methods

MapAsync(ArgumentDataTypeMapperContext)

Maps asynchronously.

Declaration

```
public Task<ArgumentDataTypeMapperResult> MapAsync(ArgumentDataTypeMapperContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentDataTypeMapperContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< ArgumentDataTypeMapperResult >	

Implements

[IArgumentDataTypeMapper](#)
[IMapper<TContext, TResult>](#)

Interface IArgumentDataTypeMapper

An abstraction to map an [DataType](#) to System.Type.

Inherited Members

[IMapper<ArgumentDataTypeMapperContext, ArgumentDataTypeMapperResult>.MapAsync\(ArgumentDataTypeMapperContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Mappers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IArgumentDataTypeMapper : IMapper<ArgumentDataTypeMapperContext,
ArgumentDataTypeMapperResult>
```

Namespace PerpetualIntelligence.Cli.Commands.Providers

Classes

[DefaultArgumentProvider](#)

The default argument provider.

[DefaultArgumentProviderContext](#)

The argument default provider context.

[DefaultArgumentProviderResult](#)

The argument default provider result.

[DefaultArgumentValueProvider](#)

The default argument default value provider.

[DefaultArgumentValueProviderContext](#)

The argument default value provider context.

[DefaultArgumentValueProviderResult](#)

The argument default value provider result.

Interfaces

[IDefaultArgumentProvider](#)

An abstraction to provide an argument's default value.

[IDefaultArgumentValueProvider](#)

An abstraction to provide an argument's default value.

Class DefaultArgumentProvider

The default argument provider.

Inheritance

System.Object
DefaultArgumentProvider

Implements

[IDefaultArgumentProvider](#)
[IProvider](#)<[DefaultArgumentProviderContext](#), [DefaultArgumentProviderResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class DefaultArgumentProvider : IDefaultArgumentProvider, IProvider<DefaultArgumentProviderContext, DefaultArgumentProviderResult>
```

Constructors

DefaultArgumentProvider(CliOptions, ILogger<DefaultArgumentProvider>)

Initialize a new instance.

Declaration

```
public DefaultArgumentProvider(CliOptions options, ILogger<DefaultArgumentProvider> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< DefaultArgumentProvider >	logger	The logger.

Methods

ProvideAsync(DefaultArgumentProviderContext)

Provides default values for all the command arguments.

Declaration

```
public Task<DefaultArgumentProviderResult> ProvideAsync(DefaultArgumentProviderContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
DefaultArgumentProviderContext	context	The argument default value provider context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< DefaultArgumentProviderResult >	The DefaultArgumentValueProviderResult instance that contains the default values.

Exceptions

TYPE	CONDITION
ErrorException	

Implements

[IDefaultArgumentProvider](#)
[IProvider<TContext, TResult>](#)

See Also

[DefaultArgument](#)

Class DefaultArgumentProviderContext

The argument default provider context.

Inheritance

System.Object
DefaultArgumentProviderContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class DefaultArgumentProviderContext
```

Constructors

DefaultArgumentProviderContext(CommandDescriptor)

Initialize a new instance.

Declaration

```
public DefaultArgumentProviderContext(CommandDescriptor commandDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandDescriptor	commandDescriptor	The command descriptor.

Properties

CommandDescriptor

The command descriptor.

Declaration

```
public CommandDescriptor CommandDescriptor { get; set; }
```

Property Value

TYPE	DESCRIPTION
CommandDescriptor	

Class DefaultArgumentProviderResult

The argument default provider result.

Inheritance

System.Object
DefaultArgumentProviderResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class DefaultArgumentProviderResult
```

Constructors

DefaultArgumentProviderResult(ArgumentDescriptor)

Initialize a new instance.

Declaration

```
public DefaultArgumentProviderResult(ArgumentDescriptor defaultArgumentDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentDescriptor	defaultArgumentDescriptor	The default argument descriptor.

Properties

DefaultArgumentDescriptor

The default argument descriptor.

Declaration

```
public ArgumentDescriptor DefaultArgumentDescriptor { get; }
```

Property Value

TYPE	DESCRIPTION
ArgumentDescriptor	

Class DefaultArgumentValueProvider

The default argument default value provider.

Inheritance

System.Object
DefaultArgumentValueProvider

Implements

IDefaultArgumentValueProvider
IProvider<DefaultArgumentValueProviderContext, DefaultArgumentValueProviderResult>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class DefaultArgumentValueProvider : IDefaultArgumentValueProvider,
    IProvider<DefaultArgumentValueProviderContext, DefaultArgumentValueProviderResult>
```

Constructors

DefaultArgumentValueProvider(IStringComparer)

Initialize a new instance.

Declaration

```
public DefaultArgumentValueProvider(IStringComparer stringComparer)
```

Parameters

TYPE	NAME	DESCRIPTION
IStringComparer	stringComparer	The string comparer.

Methods

ProvideAsync(DefaultArgumentValueProviderContext)

Provides default values for all the command arguments.

Declaration

```
public Task<DefaultArgumentValueProviderResult> ProvideAsync(DefaultArgumentValueProviderContext context)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
DefaultArgumentValueProviderContext	context	The argument default value provider context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< DefaultArgumentValueProviderResult >	The DefaultArgumentValueProviderResult instance that contains the default values.

Exceptions

TYPE	CONDITION
ErrorException	

Implements

[IDefaultArgumentValueProvider](#)
[IProvider<TContext, TResult>](#)

Class DefaultArgumentValueProviderContext

The argument default value provider context.

Inheritance

System.Object
DefaultArgumentValueProviderContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class DefaultArgumentValueProviderContext
```

Constructors

DefaultArgumentValueProviderContext(CommandDescriptor)

Initialize a new instance.

Declaration

```
public DefaultArgumentValueProviderContext(CommandDescriptor commandDescriptor)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandDescriptor	commandDescriptor	The command descriptor.

Properties

CommandDescriptor

The command descriptor.

Declaration

```
public CommandDescriptor CommandDescriptor { get; set; }
```

Property Value

TYPE	DESCRIPTION
CommandDescriptor	

Class DefaultArgumentValueProviderResult

The argument default value provider result.

Inheritance

System.Object
DefaultArgumentValueProviderResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class DefaultArgumentValueProviderResult
```

Constructors

DefaultArgumentValueProviderResult(ArgumentDescriptors)

Initialize a new instance.

Declaration

```
public DefaultArgumentValueProviderResult(ArgumentDescriptors defaultValueArgumentDescriptors)
```

Parameters

TYPE	NAME	DESCRIPTION
ArgumentDescriptors	defaultValueArgumentDescriptors	The default value argument descriptors.

Properties

DefaultValueArgumentDescriptors

The default value argument descriptors.

Declaration

```
public ArgumentDescriptors DefaultValueArgumentDescriptors { get; }
```

Property Value

TYPE	DESCRIPTION
ArgumentDescriptors	

Interface IDefaultArgumentProvider

An abstraction to provide an argument's default value.

Inherited Members

[IProvider<DefaultArgumentProviderContext, DefaultArgumentProviderResult>.ProvideAsync\(DefaultArgumentProviderContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IDefaultArgumentProvider : IProvider<DefaultArgumentProviderContext,  
DefaultArgumentProviderResult>
```

Interface IDefaultArgumentValueProvider

An abstraction to provide an argument's default value.

Inherited Members

[IProvider<DefaultArgumentValueProviderContext, DefaultArgumentValueProviderResult>.ProvideAsync\(DefaultArgumentValueProviderContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Providers](#)

Assembly: [PerpetualIntelligence.Cli.dll](#)

Syntax

```
public interface IDefaultArgumentValueProvider : IProvider<DefaultArgumentValueProviderContext, DefaultArgumentValueProviderResult>
```

Namespace PerpetualIntelligence.Cli.Commands.Publishers

Classes

[ErrorPublisher](#)

The default [IErrorPublisher](#) to publish an [Error](#).

[ErrorPublisherContext](#)

The [IExceptionPublisher](#) context.

[ExceptionPublisher](#)

The default [IExceptionPublisher](#) to publish an System.Exception.

[ExceptionPublisherContext](#)

The [IExceptionPublisher](#) context.

Interfaces

[IErrorPublisher](#)

An abstraction to publish [Error](#).

[IExceptionPublisher](#)

An abstraction to publish System.Exception.

Class ErrorPublisher

The default [IErrorPublisher](#) to publish an [Error](#).

Inheritance

System.Object
ErrorPublisher

Implements

[IErrorPublisher](#)
[IPublisherNoResult<ErrorPublisherContext>](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Publishers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ErrorPublisher : IErrorPublisher, IPublisherNoResult<ErrorPublisherContext>
```

Constructors

ErrorPublisher(CliOptions, ILogger<ExceptionPublisher>)

Initialize a new instance.

Declaration

```
public ErrorPublisher(CliOptions options, ILogger<ExceptionPublisher> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< ExceptionPublisher >	logger	The logger.

Methods

PublishAsync(ErrorPublisherContext)

Publish the [Error](#) asynchronously

Declaration

```
public Task PublishAsync(ErrorPublisherContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
ErrorPublisherContext	context	The error to publish.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	The string representation.

Implements

- [IPublisher](#)
- [IPublisherNoResult<TContext>](#)

Class ErrorPublisherContext

The [IExceptionPublisher](#) context.

Inheritance

System.Object
ErrorPublisherContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Publishers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ErrorPublisherContext
```

Constructors

ErrorPublisherContext(Error)

Initialize a new instance.

Declaration

```
public ErrorPublisherContext(Error error)
```

Parameters

TYPE	NAME	DESCRIPTION
Error	error	The error.

ErrorPublisherContext(Error, String)

Initialize a new instance.

Declaration

```
public ErrorPublisherContext(Error error, string rawCommandString)
```

Parameters

TYPE	NAME	DESCRIPTION
Error	error	The error.
System.String	rawCommandString	The raw command string.

Properties

Error

The error.

Declaration

```
public Error Error { get; set; }
```

Property Value

TYPE	DESCRIPTION
Error	

RawCommandString

The raw command string.

Declaration

```
public string RawCommandString { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class ExceptionPublisher

The default [IExceptionPublisher](#) to publish an System.Exception.

Inheritance

System.Object
ExceptionPublisher

Implements

[IExceptionPublisher](#)
[IPublisherNoResult<ExceptionPublisherContext>](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Publishers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ExceptionPublisher : IExceptionPublisher, IPublisherNoResult<ExceptionPublisherContext>
```

Constructors

ExceptionPublisher(CliOptions, ILogger<ExceptionPublisher>)

Initialize a new instance.

Declaration

```
public ExceptionPublisher(CliOptions options, ILogger<ExceptionPublisher> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< ExceptionPublisher >	logger	The logger.

Methods

PublishAsync(ExceptionPublisherContext)

Publish the System.Exception asynchronously to the logger.

Declaration

```
public Task PublishAsync(ExceptionPublisherContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
ExceptionPublisherContext	context	The error to publish.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	The string representation.

Implements

- [IExceptionPublisher](#)
- [IPublisherNoResult<TContext>](#)

Class ExceptionPublisherContext

The [IExceptionPublisher](#) context.

Inheritance

System.Object
ExceptionPublisherContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Publishers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ExceptionPublisherContext
```

Constructors

ExceptionPublisherContext(String, Exception)

Initialize a new instance.

Declaration

```
public ExceptionPublisherContext(string rawCommandString, Exception exception)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	rawCommandString	The raw command string.
System.Exception	exception	The exception.

Properties

Exception

The exception.

Declaration

```
public Exception Exception { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Exception	

RawCommandString

The command string.

Declaration

```
public string RawCommandString { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Interface IErrorPublisher

An abstraction to publish [Error](#).

Inherited Members

[IPublisherNoResult<ErrorPublisherContext>.PublishAsync\(ErrorPublisherContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Publishers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IErrorPublisher : IPublisherNoResult<ErrorPublisherContext>
```

Interface IExceptionPublisher

An abstraction to publish System.Exception.

Inherited Members

[IPublisherNoResult<ExceptionPublisherContext>.PublishAsync\(ExceptionPublisherContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Publishers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface IExceptionPublisher : IPublisherNoResult<ExceptionPublisherContext>
```

Namespace PerpetualIntelligence.Cli.Commands.Routers

Classes

[CommandRouter](#)

The default [ICommandRouter](#).

[CommandRouterContext](#)

The `pi-cli` generic command router context.

[CommandRouterResult](#)

The command router context.

Interfaces

[ICommandRouter](#)

An abstraction of a command router.

Class CommandRouter

The default [ICommandRouter](#).

Inheritance

System.Object
CommandRouter

Implements

[ICommandRouter](#)
[IRouter](#)<[CommandRouterContext](#), [CommandRouterResult](#), [ICommandHandler](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Routers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class CommandRouter : ICommandRouter, IRouter<CommandRouterContext, CommandRouterResult, ICommandHandler>
```

Constructors

CommandRouter(ILicenseExtractor, ICommandExtractor, ICommandHandler)

Initializes a new instance.

Declaration

```
public CommandRouter(ILicenseExtractor licenseExtractor, ICommandExtractor commandExtractor, ICommandHandler commandHandler)
```

Parameters

TYPE	NAME	DESCRIPTION
ILicenseExtractor	licenseExtractor	The license extractor.
ICommandExtractor	commandExtractor	The command extractor.
ICommandHandler	commandHandler	The command handler.

Methods

RouteAsync(CommandRouterContext)

Routes the command request to the registered handler.

Declaration

```
public async Task<CommandRouterResult> RouteAsync(CommandRouterContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandRouterContext	context	The router context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandRouterResult >	The CommandRouterResult instance.

TryFindHandlerAsync(CommandRouterContext)

Finds a matching route handler for a request asynchronously.

Declaration

```
public Task<TryResultOrError<ICommandHandler>> TryFindHandlerAsync(CommandRouterContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandRouterContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < ICommandHandler >>	The request handler.

Implements

[ICommandRouter](#)

[IRouter](#)<[TContext](#), [TResult](#), [THandler](#)>

Class CommandRouterContext

The `pi-cli` generic command router context.

Inheritance

System.Object
CommandRouterContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Routers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandRouterContext
```

Constructors

CommandRouterContext(String, Nullable<CancellationToken>)

The command string.

Declaration

```
public CommandRouterContext(string rawCommandString, CancellationTokens? cancellationToken = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	rawCommandString	The raw command string.
System.Nullable<System.Threading.CancellationTokens>	cancellationToken	The cancellation token.

Properties

CancellationToken

The cancellation token.

Declaration

```
public CancellationTokens? CancellationToken { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Threading.CancellationTokens>	

RawCommandString

The raw command string.

Declaration

```
public string RawCommandString { get; protected set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class CommandRouterResult

The command router context.

Inheritance

System.Object

CommandRouterResult

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Routers](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandRouterResult
```

Interface ICommandRouter

An abstraction of a command router.

Inherited Members

[IRouter<CommandRouterContext, CommandRouterResult, ICommandHandler>.RouteAsync\(CommandRouterContext\)](#)

[IRouter<CommandRouterContext, CommandRouterResult, ICommandHandler>.TryFindHandlerAsync\(CommandRouterContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Routers](#)

Assembly: [PerpetualIntelligence.Cli.dll](#)

Syntax

```
public interface ICommandRouter : IRouter<CommandRouterContext, CommandRouterResult, ICommandHandler>
```

Namespace PerpetualIntelligence.Cli.Commands.Runners

Classes

[ClearScreenRunner](#)

The clear screen command runner.

[CommandRunner](#)

The command runner.

[CommandRunnerContext](#)

The command runner context.

[CommandRunnerResult](#)

The command runner result.

[ExitRunner](#)

The exit command runner.

[LicInfoRunner](#)

The `lic` command displays the current licensing information.

[RunRunner](#)

The `run` command runner.

Interfaces

[ICommandRunner](#)

An abstraction of a command runner.

Class ClearScreenRunner

The clear screen command runner.

Inheritance

System.Object
CommandRunner
ClearScreenRunner

Implements

ICommandRunner
 IRunner<CommandRunnerContext, CommandRunnerResult>

Inherited Members

CommandRunner.Options
CommandRunner.Logger
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)
Assembly: [PerpetualIntelligence.Cli.dll](#)

Syntax

```
public class ClearScreenRunner : CommandRunner, ICommandRunner, IRunner<CommandRunnerContext, CommandRunnerResult>
```

Constructors

ClearScreenRunner(IHost, CliOptions, ILogger<ExitRunner>)

Initializes a new instance.

Declaration

```
public ClearScreenRunner(IHost host, CliOptions options, ILogger<ExitRunner> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Hosting.IHost	host	The host.
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< ExitRunner >	logger	The logger.

Methods

RunAsync(CommandRunnerContext)

Runs a context asynchronously.

Declaration

```
public override Task<CommandRunnerResult> RunAsync(CommandRunnerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandRunnerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandRunnerResult >	The runner result.

Overrides

[CommandRunner.RunAsync\(CommandRunnerContext\)](#)

Implements

- [ICommandRunner](#)
- [IRunner<TContext, TResult>](#)

Class CommandRunner

The command runner.

Inheritance

System.Object
CommandRunner
[ClearScreenRunner](#)
[ExitRunner](#)
[RunRunner](#)

Implements

[ICommandRunner](#)
[IRunner](#)<[CommandRunnerContext](#), [CommandRunnerResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public abstract class CommandRunner : ICommandRunner, IRunner<CommandRunnerContext, CommandRunnerResult>
```

Constructors

CommandRunner(CliOptions, ILogger)

Initialize a new instance.

Declaration

```
protected CommandRunner(CliOptions options, ILogger logger)
```

Parameters

TYPE	NAME	DESCRIPTION
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger	logger	The logger.

Properties

Logger

The command runner logger.

Declaration

```
protected ILogger Logger { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

Options

The configuration options.

Declaration

<pre>public CliOptions Options { get; }</pre>

Property Value

TYPE	DESCRIPTION
CliOptions	

Methods

RunAsync(CommandRunnerContext)

Runs a context asynchronously.

Declaration

<pre>public abstract Task<CommandRunnerResult> RunAsync(CommandRunnerContext context)</pre>

Parameters

TYPE	NAME	DESCRIPTION
CommandRunnerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandRunnerResult >	The runner result.

Implements

[ICommandRunner](#)

[IRunner<TContext, TResult>](#)

Class CommandRunnerContext

The command runner context.

Inheritance

System.Object
CommandRunnerContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandRunnerContext
```

Constructors

CommandRunnerContext(Command)

Initialize a new instance.

Declaration

```
public CommandRunnerContext(Command command)
```

Parameters

TYPE	NAME	DESCRIPTION
Command	command	The command to run.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

Command

The checked command to run.

Declaration

```
public Command Command { get; }
```

Property Value

TYPE	DESCRIPTION
Command	

Class CommandRunnerResult

The command runner result.

Inheritance

System.Object

CommandRunnerResult

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CommandRunnerResult
```

Class ExitRunner

The exit command runner.

Inheritance

System.Object
[CommandRunner](#)
ExitRunner

Implements

[ICommandRunner](#)
[IRunner](#)<[CommandRunnerContext](#), [CommandRunnerResult](#)>

Inherited Members

[CommandRunner.Options](#)
[CommandRunner.Logger](#)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ExitRunner : CommandRunner, ICommandRunner, IRunner<CommandRunnerContext, CommandRunnerResult>
```

Constructors

ExitRunner(IHost, CliOptions, ILogger<ExitRunner>)

Initializes a new instance.

Declaration

```
public ExitRunner(IHost host, CliOptions options, ILogger<ExitRunner> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Hosting.IHost	host	The host.
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< ExitRunner >	logger	The logger.

Methods

RunAsync(CommandRunnerContext)

Runs a context asynchronously.

Declaration

```
public override async Task<CommandRunnerResult> RunAsync(CommandRunnerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandRunnerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandRunnerResult >	The runner result.

Overrides

[CommandRunner.RunAsync\(CommandRunnerContext\)](#)

Implements

[ICommandRunner](#)

[IRunner<TContext, TResult>](#)

Interface ICommandRunner

An abstraction of a command runner.

Inherited Members

[IRunner<CommandRunnerContext, CommandRunnerResult>.RunAsync\(CommandRunnerContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface ICommandRunner : IRunner<CommandRunnerContext, CommandRunnerResult>
```

Class LicInfoRunner

The `lic` command displays the current licensing information.

Inheritance

System.Object
LicInfoRunner

Implements

[ICommandRunner](#)
[IRunner<CommandRunnerContext, CommandRunnerResult>](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LicInfoRunner : ICommandRunner, IRunner<CommandRunnerContext, CommandRunnerResult>
```

Constructors

LicInfoRunner(ILicenseExtractor, ILicenseChecker)

Initialize a new instance.

Declaration

```
public LicInfoRunner(ILicenseExtractor licenseExtractor, ILicenseChecker licenseChecker)
```

Parameters

TYPE	NAME	DESCRIPTION
ILicenseExtractor	licenseExtractor	
ILicenseChecker	licenseChecker	

Methods

RunAsync(CommandRunnerContext)

Runs a context asynchronously.

Declaration

```
public async Task<CommandRunnerResult> RunAsync(CommandRunnerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandRunnerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandRunnerResult >	The runner result.

Implements

[ICommandRunner](#)
[IRunner<TContext, TResult>](#)

Class RunRunner

The `run` command runner.

Inheritance

System.Object

[CommandRunner](#)

RunRunner

Implements

[ICommandRunner](#)

[IRunner](#)<[CommandRunnerContext](#), [CommandRunnerResult](#)>

Inherited Members

[CommandRunner.Options](#)

[CommandRunner.Logger](#)

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Commands.Runners](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class RunRunner : CommandRunner, ICommandRunner, IRunner<CommandRunnerContext, CommandRunnerResult>
```

Constructors

RunRunner(CliOptions, ILogger)

Initialize a new instance.

Declaration

```
public RunRunner(CliOptions options, ILogger logger)
```

Parameters

TYPE	NAME	DESCRIPTION
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger	logger	The logger.

Methods

RunAsync(CommandRunnerContext)

Runs a context asynchronously.

Declaration

```
public override Task<CommandRunnerResult> RunAsync(CommandRunnerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
CommandRunnerContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< CommandRunnerResult >	The runner result.

Overrides

[CommandRunner.RunAsync\(CommandRunnerContext\)](#)

Implements

[ICommandRunner](#)

[IRunner<TContext, TResult>](#)

Namespace PerpetualIntelligence.Cli.Commands.Stores

Interfaces

[ICommandDescriptorStore](#)

A store abstraction to lookup a [CommandDescriptor](#).

Interface ICommandDescriptorStore

A store abstraction to lookup a [CommandDescriptor](#).

Namespace: [PerpetualIntelligence.Cli.Commands.Stores](#)

Assembly: [PerpetualIntelligence.Cli.dll](#)

Syntax

```
public interface ICommandDescriptorStore
```

Methods

TryFindByIdAsync(String)

Attempts to finds a [CommandDescriptor](#) by its id.

Declaration

```
Task<TryResultOrError<CommandDescriptor>> TryFindByIdAsync(string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The command id.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

TryFindByNameAsync(String)

Attempts to find a a [CommandDescriptor](#) by its name.

Declaration

```
Task<TryResultOrError<CommandDescriptor>> TryFindByNameAsync(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	The command name.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

TryFindByPrefixAsync(String)

Attempts to find a a [CommandDescriptor](#) by its prefix.

Declaration

```
Task<TryResultOrError<CommandDescriptor>> TryFindByPrefixAsync(string prefix)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	prefix	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

TryMatchByPrefixAsync(String)

Attempts to match a a [CommandDescriptor](#) by its prefix.

Declaration

```
Task<TryResultOrError<CommandDescriptor>> TryMatchByPrefixAsync(string prefix)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	prefix	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

Namespace PerpetualIntelligence.Cli.Configuration.Options

Classes

[AuthenticationOptions](#)

The authentication configuration options.

[CheckerOptions](#)

The checker configuration option.

[CliOptions](#)

The `pi-cli` configuration options.

[ExtractorOptions](#)

The extractor configuration option.

[HostingOptions](#)

The hosting configuration options.

[HttpOptions](#)

The HTTP configuration options.

[LicensingOptions](#)

The licensing configuration options.

[LoggingOptions](#)

The logging configuration options.

[TerminalOptions](#)

The terminal configuration options.

Class AuthenticationOptions

The authentication configuration options.

Inheritance

System.Object
AuthenticationOptions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class AuthenticationOptions
```

Properties

ApplicationId

The authorized application or the client identifier.

Declaration

```
public string ApplicationId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Authority

The authentication authority.

Declaration

```
public string Authority { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

HttpClientName

The HTTP client name.

Declaration

```
public string HttpClientName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The name must match the [AddAuthentication<TProvider, TAppFactory, TAppCache, TDelegateHandler>\(ICliBuilder, String, String, Nullable<Int32>\)](#)

RedirectUri

The authentication redirect URI. Defaults to <http://localhost>.

Declaration

```
public string RedirectUri { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

See Also

<https://github.com/AzureAD/microsoft-authentication-extensions-for-dotnet/wiki/Cross-platform-Token-Cache>

<https://docs.microsoft.com/en-us/azure/active-directory/develop/msal-net-token-cache-serialization?tabs=desktop>

Scopes

The authentication scopes.

Declaration

```
public string[] Scopes { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

TenantId

The authentication tenant identifier.

Declaration

```
public string TenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

UseEmbeddedView

Specifies if the public client application should used an embedded web browser or the system default browser.

Declaration

```
public bool? UseEmbeddedView { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Class CheckerOptions

The checker configuration option.

Inheritance

System.Object
CheckerOptions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CheckerOptions
```

Properties

AllowObsoleteArgument

Determines whether the checker allows an obsolete argument.

Declaration

```
public bool? AllowObsoleteArgument { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

DataTypeCheck

Defines the checker data type checks. Defaults to `null` or no data type checks.

Declaration

```
public string DataTypeCheck { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

StrictTypeChecking

Determines whether the checker allows strict type checking.

Declaration

```
public bool? StrictTypeChecking { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Class CliOptions

The `pi-cli` configuration options.

Inheritance

System.Object
CliOptions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CliOptions
```

Properties

Authentication

The authentication configuration options.

Declaration

```
public AuthenticationOptions Authentication { get; set; }
```

Property Value

TYPE	DESCRIPTION
AuthenticationOptions	

Checker

The checker configuration options.

Declaration

```
public CheckerOptions Checker { get; set; }
```

Property Value

TYPE	DESCRIPTION
CheckerOptions	

Extractor

The extractor configuration options.

Declaration

```
public ExtractorOptions Extractor { get; set; }
```

Property Value

TYPE	DESCRIPTION
ExtractorOptions	

Hosting

The hosting configuration options.

Declaration

```
public HostingOptions Hosting { get; set; }
```

Property Value

TYPE	DESCRIPTION
HostingOptions	

Http

The HTTP configuration options.

Declaration

```
public HttpOptions Http { get; set; }
```

Property Value

TYPE	DESCRIPTION
HttpOptions	

Licensing

The licensing configuration options.

Declaration

```
public LicensingOptions Licensing { get; set; }
```

Property Value

TYPE	DESCRIPTION
LicensingOptions	

Logging

The logging configuration options.

Declaration

```
public LoggingOptions Logging { get; set; }
```

Property Value

TYPE	DESCRIPTION
LoggingOptions	

Terminal

The terminal configuration options.

Declaration

```
public TerminalOptions Terminal { get; set; }
```

Property Value

TYPE	DESCRIPTION
TerminalOptions	

Class ExtractorOptions

The extractor configuration option.

Inheritance

System.Object
ExtractorOptions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class ExtractorOptions
```

Properties

ArgumentAlias

Determines whether the extractor support extracting the argument by alias. Defaults to `null`.

Declaration

```
public bool? ArgumentAlias { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Remarks

Argument alias supports the legacy apps that identified a command argument with an id and an alias string. For modern console apps, we recommend using just an argument identifier. The core data model is optimized to work with argument id. In general, an app should not identify the same argument with multiple string. Using alias will degrade the performance.

ArgumentAliasPrefix

The argument alias prefix. Defaults to `-`.

Declaration

```
public string ArgumentAliasPrefix { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The argument alias prefix cannot be `null` or whitespace.

ArgumentPrefix

The argument prefix. Defaults to `-`.

Declaration

```
public string ArgumentPrefix { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The argument prefix cannot be `null` or whitespace.

ArgumentSeparator

The argument value separator. Defaults to equals char `=`.

Declaration

```
public string ArgumentSeparator { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ArgumentValueWithIn

Defines the token within which to extract an argument value. Default to `null`.

Declaration

```
public string ArgumentValueWithIn { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

CommandIdRegexPattern

Defines the Regex pattern for command identifier. Defaults to `^[A-Za-z0-9_-]*$`.

Declaration

```
public string CommandIdRegexPattern { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

DefaultArgument

Determines whether the extractor support extracting default arguments. Defaults to `null`.

Declaration

```
public bool? DefaultArgument { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

DefaultArgumentValue

Determines whether the extractor support extracting default argument values. Defaults to `null`.

Declaration

```
public bool? DefaultArgumentValue { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Separator

The command string separator. Defaults to a single space char.

Declaration

```
public string Separator { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class HostingOptions

The hosting configuration options.

Inheritance

System.Object

[LoggingOptions](#)

HostingOptions

Inherited Members

[LoggingOptions.ObsecureErrorArgumentString](#)

[LoggingOptions.ObsureErrorArguments](#)

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class HostingOptions : LoggingOptions
```

Properties

CommandRouterTimeout

The command router timeout in milliseconds. Defaults to 10 seconds. Use System.Threading.Timeout.Infinite for no timeout.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

ErrorHandling

Defines the hosting and routing error handling. Defaults to `default` error handling.

Declaration

```
public string ErrorHandling { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ServiceImplementation

Defines the hosting and routing service implementation. Defaults to `default` service implementation.

Declaration

```
public string ServiceImplementation { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Store

Defines the hosting and routing store. Defaults to `in_memory` store implementation.

Declaration

```
public string Store { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

UnicodeSupport

Defines the hosting and routing service implementation. Defaults to `default` service implementation.

Declaration

```
public string UnicodeSupport { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class HttpOptions

The HTTP configuration options.

Inheritance

System.Object

HttpOptions

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class HttpOptions
```

Class LicensingOptions

The licensing configuration options.

Inheritance

System.Object

LicensingOptions

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LicensingOptions
```

Properties

AuthorizedApplicationId

The license authorized application id.

Declaration

```
public string AuthorizedApplicationId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

CheckMode

The license check mode. Defaults to [Online](#).

Declaration

```
public string CheckMode { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantId

The license consumer tenant id.

Declaration

```
public string ConsumerTenantId { get; set; }
```


Property Value

TYPE	DESCRIPTION
System.String	

HttpClientName

The HTTP client name for [Online](#) checks.

Declaration

```
public string HttpClientName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Remarks

The name must match the [AddLicensingClient\(ICliBuilder, String, TimeSpan\)](#)

KeySource

The license key source. Defaults to [JsonFile](#).

Declaration

```
public string KeySource { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

LicenseKey

The license key or the file containing license key.

Declaration

```
public string LicenseKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ProviderId

The license SaaS provider id or the provider tenant id.

Declaration

```
public string ProviderId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Subject

The subject or a licensing context to check the license. Your subscription id or any other domain identifier usually establishes your licensing context.

Declaration

```
public string Subject { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class LoggingOptions

The logging configuration options.

Inheritance

System.Object

[LoggingOptions](#)

LoggingOptions

Inherited Members

[LoggingOptions.ObsecureErrorArgumentString](#)

[LoggingOptions.ObsureErrorArguments](#)

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LoggingOptions : LoggingOptions
```

Class TerminalOptions

The terminal configuration options.

Inheritance

System.Object

TerminalOptions

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Configuration.Options](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class TerminalOptions
```

Namespace PerpetualIntelligence.Cli.Extensions

Classes

[CommandExtensions](#)

The System.Console extension methods.

[ICliBuilderExtensions](#)

The [ICliBuilder](#) extension methods.

[IHostExtensions](#)

The Microsoft.Extensions.Hosting.IHost extension methods.

[IServiceCollectionExtensions](#)

The Microsoft.Extensions.DependencyInjection.IServiceCollection extension methods.

Class CommandExtensions

The System.Console extension methods.

Inheritance

System.Object
CommandExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Extensions](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public static class CommandExtensions
```

Methods

ReadAnswerAsync(Command, String, String[])

Prints the question to the console and reads an answer from the standard input stream.

Declaration

```
[WriteUnitTest]  
public static Task<string> ReadAnswerAsync(this Command command, string question, params string[] answers)
```

Parameters

TYPE	NAME	DESCRIPTION
Command	command	The command.
System.String	question	The question to print. The <code>?</code> will be appended at the end.
System.String[]	answers	The allowed answers or <code>null</code> if all answers are allowed. It is recommended to keep the answers short for readability. If specified this method will print the answers with question in the format <code>{question} {answer1}/{answer2}/{answer3}?</code>

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	The answer for the question or <code>null</code> if canceled.

Class ICliBuilderExtensions

The [ICliBuilder](#) extension methods.

Inheritance

System.Object
ICliBuilderExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Extensions](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public static class ICliBuilderExtensions
```

Methods

AddArgumentChecker<TMapper, TChecker>(ICliBuilder)

Adds the [IArgumentDataTypesMapper](#) and [IArgumentChecker](#) to the service collection.

Declaration

```
public static ICliBuilder AddArgumentChecker<TMapper, TChecker>(this ICliBuilder builder)
    where TMapper : class, IArgumentDataTypesMapper where TChecker : class, IArgumentChecker
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TMapper	The argument mapper type.
TChecker	The argument checker type.

AddAuthentication<TProvider, TAppFactory, TAppCache, TDelegateHandler>(ICliBuilder, String, String, Nullable<Int32>)

Adds authentication to the service collection.

Declaration

```
public static ICliBuilder AddAuthentication<TProvider, TAppFactory, TAppCache, TDelegateHandler>(this
ICliBuilder builder, string name, string baseAddress = null, int? timeout = 120000)
    where TProvider : class, IAuthenticationProvider where TAppFactory : class,
IMsalPublicClientApplicationFactory where TAppCache : class, IClientCrossPlatformTokenCache where
TDelegateHandler : DelegatingHandler
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.
System.String	name	The HTTP client name.
System.String	baseAddress	The HTTP base address. Specify <code>null</code> for no specific well known base address.
System.Nullable<System.Int32>	timeout	The HTTP request timeout in milliseconds. Defaults to <code>2</code> minutes or <code>120000</code> milliseconds. We recommend a timeout of at least a few minutes, to take into account cases where the user is prompted to change password or perform 2FA.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TProvider	The authentication provider.
TAppFactory	The authentication application factory.
TAppCache	The authentication application cache.
TDelegateHandler	The authentication application delegate handler.

Remarks

Use [ClientCrossPlatformNoTokenCache](#) if your application does not require token caching.

AddCliOptions(ICliBuilder)

Adds the [CliOptions](#) to the service collection.

Declaration

```
public static ICliBuilder AddCliOptions(this ICliBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

AddDescriptor<TRunner, TChecker>(ICliBuilder, CommandDescriptor, Boolean, Boolean, Boolean)

Adds the [CommandDescriptor](#) to the service collection.

Declaration

```
public static ICliBuilder AddDescriptor<TRunner, TChecker>(this ICliBuilder builder, CommandDescriptor  
commandDescriptor, bool isGroup = false, bool isRoot = false, bool isProtected = false)  
    where TRunner : class, ICommandRunner where TChecker : class, ICommandChecker
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.
CommandDescriptor	commandDescriptor	The command descriptor.
System.Boolean	isGroup	<code>true</code> if the descriptor represents a command group; otherwise, <code>false</code> .
System.Boolean	isRoot	<code>true</code> if the descriptor represents a command root; otherwise, <code>false</code> .
System.Boolean	isProtected	<code>true</code> if the descriptor represents a protected command; otherwise, <code>false</code> .

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TRunner	The command runner type.
TChecker	The command checker type.

AddDescriptorStore<TStore>(ICliBuilder)

Adds the [ICommandDescriptorStore](#) to the service collection.

Declaration

```
public static ICliBuilder AddDescriptorStore<TStore>(this ICliBuilder builder)
    where TStore : class, ICommandDescriptorStore
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TStore	The command descriptor store type.

AddErrorPublisher<TError, TException>(ICliBuilder)

Adds the [IErrorPublisher](#) and [IExceptionPublisher](#) to the service collection.

Declaration

```
public static ICliBuilder AddErrorPublisher<TError, TException>(this ICliBuilder builder)
    where TError : class, IErrorPublisher where TException : class, IExceptionPublisher
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TError	The IErrorPublisher type.
TException	The IExceptionPublisher type.

AddExtractor<TCommand, TArgument>(ICliBuilder)

Adds the [ICommandExtractor](#) and [IArgumentExtractor](#) to the service collection.

Declaration

```
public static ICliBuilder AddExtractor<TCommand, TArgument>(this ICliBuilder builder)
    where TCommand : class, ICommandExtractor where TArgument : class, IArgumentExtractor
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TCommand	The command extractor type.
TArgument	The argument extractor type.

AddExtractor<TCommand, TArgument, TDefaultArgumentValue>(ICliBuilder)

Adds the [ICommandExtractor](#), [IArgumentExtractor](#) and [IDefaultArgumentValueProvider](#) to the service collection.

Declaration

```
public static ICliBuilder AddExtractor<TCommand, TArgument, TDefaultArgumentValue>(this ICliBuilder builder)
    where TCommand : class, ICommandExtractor where TArgument : class, IArgumentExtractor where
    TDefaultArgumentValue : class, IDefaultArgumentValueProvider
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TCommand	The command extractor type.
TArgument	The argument extractor type.
TDefaultArgumentValue	The argument default value provider type.

AddExtractor<TCommand, TArgument, TDefaultArgument, TDefaultArgumentValue>(ICliBuilder)

Adds the [ICommandExtractor](#), [IArgumentExtractor](#), [IDefaultArgumentProvider](#) and [IDefaultArgumentValueProvider](#) to the service collection.

Declaration

```
public static ICliBuilder AddExtractor<TCommand, TArgument, TDefaultArgument, TDefaultArgumentValue>(this ICliBuilder builder)
    where TCommand : class, ICommandExtractor where TArgument : class, IArgumentExtractor where
TDefaultArgument : class, IDefaultArgumentProvider where TDefaultArgumentValue : class,
IDefaultArgumentValueProvider
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TCommand	The command extractor type.
TArgument	The argument extractor type.
TDefaultArgument	The default argument provider type.
TDefaultArgumentValue	The default argument value provider type.

AddLicensing(ICliBuilder)

Adds `pi-cli` licensing to the service collection.

Declaration

```
public static ICliBuilder AddLicensing(this ICliBuilder builder)
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

AddLicensingClient(ICliBuilder, String, TimeSpan)

Adds a System.Net.Http.IHttpClientFactory to the service collection.

Declaration

```
public static ICliBuilder AddLicensingClient(this ICliBuilder builder, string name, TimeSpan timeout)
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.
System.String	name	The HTTP client name.
System.TimeSpan	timeout	The HTTP client timeout.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Remarks

The [AddLicensingClient\(ICliBuilder, String, TimeSpan\)](#) is required if you are using [Online](#) check. Please set the [HttpClientName](#) to match the name used to register this service.

AddRouter<TRouter, THandler>(ICliBuilder)

Adds the [ICommandRouter](#) and [ICommandHandler](#) to the service collection.

Declaration

```
public static ICliBuilder AddRouter<TRouter, THandler>(this ICliBuilder builder)
    where TRouter : class, ICommandRouter where THandler : class, ICommandHandler
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Type Parameters

NAME	DESCRIPTION
TRouter	
THandler	

AddStringComparer(ICliBuilder, StringComparison)

Add the [IStringComparer](#) to the service collection.

Declaration

```
public static ICliBuilder AddStringComparer(this ICliBuilder builder, StringComparison stringComparison)
```

Parameters

TYPE	NAME	DESCRIPTION
ICliBuilder	builder	The builder.

TYPE	NAME	DESCRIPTION
System.StringComparison	stringComparison	The string comparison to use.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder .

Class IHostExtensions

The Microsoft.Extensions.Hosting.IHost extension methods.

Inheritance

System.Object
IHostExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Extensions](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public static class IHostExtensions
```

Methods

RunRouterAsync(IHost, String, Nullable<CancellationToken>)

Returns a task that runs the [ICommandRouter](#) and blocks the calling thread till a cancellation request.

Declaration

```
[WriteDocumentation("Add info about exception handling for RecognitionException")]  
public static async Task RunRouterAsync(this IHost host, string caret = null, Cancellation_token?  
cancellation_token = null)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Hosting.IHost	host	The host.
System.String	caret	The command caret to show in the console.
System.Nullable<System.Threading.Cancellation_token>	cancellation_token	The cancellation token.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Class IServiceCollectionExtensions

The Microsoft.Extensions.DependencyInjection.IServiceCollection extension methods.

Inheritance

System.Object
IServiceCollectionExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Extensions](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public static class IServiceCollectionExtensions
```

Methods

AddCli(IServiceCollection)

Adds the core `pi-cli` services.

Declaration

```
public static ICliBuilder AddCli(this IServiceCollection services)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The services.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder instance.

AddCli(IServiceCollection, IConfiguration)

Adds the core `pi-cli` services.

Declaration

```
public static ICliBuilder AddCli(this IServiceCollection services, IConfiguration configuration)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The services.
Microsoft.Extensions.Configuration.IConfiguration	configuration	The configuration.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder instance.

AddCli(IServiceCollection, Action<CliOptions>)

Adds the core `pi-cli` services.

Declaration

```
public static ICliBuilder AddCli(this IServiceCollection services, Action<CliOptions> setupAction)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The services.
System.Action< CliOptions >	setupAction	The setup action.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICliBuilder instance.

AddCliBuilder(IServiceCollection)

Adds the core [ICliBuilder](#).

Declaration

```
public static ICliBuilder AddCliBuilder(this IServiceCollection services)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The services.

Returns

TYPE	DESCRIPTION
ICliBuilder	The configured ICLiBuilder instance.

Namespace PerpetualIntelligence.Cli.Integration

Classes

[CliBuilder](#)

The default [ICliBuilder](#).

[CliHostedService](#)

The `pi-cli` hosted service to manage the application lifetime and terminal customization.

Interfaces

[ICliBuilder](#)

An abstraction of `pi-cli` service builder.

Class CliBuilder

The default [ICliBuilder](#).

Inheritance

System.Object
CliBuilder

Implements

[ICliBuilder](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Integration](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CliBuilder : ICliBuilder
```

Constructors

CliBuilder(IServiceCollection)

Initializes a new instance.

Declaration

```
public CliBuilder(IServiceCollection services)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	services	The services.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	services

Properties

Services

The service collection.

Declaration

```
public IServiceCollection Services { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	

Implements

[ICliBuilder](#)

Extension Methods

- [ICliBuilderExtensions.AddArgumentChecker<TMapper, TChecker>\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddAuthentication<TProvider, TAppFactory, TAppCache, TDelegateHandler>\(ICliBuilder, String, String, Nullable<Int32>\)](#)
- [ICliBuilderExtensions.AddCliOptions\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddDescriptor<TRunner, TChecker>\(ICliBuilder, CommandDescriptor, Boolean, Boolean, Boolean\)](#)
- [ICliBuilderExtensions.AddDescriptorStore<TStore>\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddErrorPublisher<TError, TException>\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddExtractor<TCommand, TArgument>\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddExtractor<TCommand, TArgument, TDefaultArgumentValue>\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddExtractor<TCommand, TArgument, TDefaultArgument, TDefaultArgumentValue>\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddLicensing\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddLicensingClient\(ICliBuilder, String, TimeSpan\)](#)
- [ICliBuilderExtensions.AddRouter<TRouter, THandler>\(ICliBuilder\)](#)
- [ICliBuilderExtensions.AddStringComparer\(ICliBuilder, StringComparison\)](#)

Class CliHostedService

The `pi-cli` hosted service to manage the application lifetime and terminal customization.

Inheritance

System.Object
CliHostedService

Implements

Microsoft.Extensions.Hosting.IHostedService

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Integration](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class CliHostedService : IHostedService
```

Constructors

CliHostedService(IHost, IHostApplicationLifetime, ILicenseExtractor, ILicenseChecker, CliOptions, ILogger<CliHostedService>)

Initializes a new instance.

Declaration

```
public CliHostedService(IHost host, IHostApplicationLifetime hostApplicationLifetime, ILicenseExtractor licenseExtractor, ILicenseChecker licenseChecker, CliOptions cliOptions, ILogger<CliHostedService> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Hosting.IHost	host	The host.
Microsoft.Extensions.Hosting.IHostApplicationLifetime	hostApplicationLifetime	The host application lifetime.
ILicenseExtractor	licenseExtractor	The license extractor.
ILicenseChecker	licenseChecker	The license checker.
CliOptions	cliOptions	The configuration options.

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger<CliHostedService>	logger	The logger.

Methods

OnStarted()

Triggered when the `pi-cli` application host has fully started.

Declaration

```
protected virtual void OnStarted()
```

OnStopped()

Triggered when the `pi-cli` application host has completed a graceful shutdown. The application will not exit until all callbacks registered on this token have completed.

Declaration

```
protected virtual void OnStopped()
```

OnStopping()

Triggered when the `pi-cli` application host is starting a graceful shutdown. Shutdown will block until all callbacks registered on this token have completed.

Declaration

```
protected virtual void OnStopping()
```

PrintHostApplicationHeaderAsync()

Allows the host application to print the custom header.

Declaration

```
protected virtual Task PrintHostApplicationHeaderAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

PrintHostApplicationLicensingAsync(License)

Allows host application to print custom licensing information.

Declaration

```
protected virtual Task PrintHostApplicationLicensingAsync(License license)
```

Parameters

TYPE	NAME	DESCRIPTION
License	license	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

RegisterHostApplicationEventsAsync(IHostApplicationLifetime)

Allows the application to register its custom Microsoft.Extensions.Hosting.IHostApplicationLifetime events.

Declaration

```
protected virtual Task RegisterHostApplicationEventsAsync(IHostApplicationLifetime hostApplicationLifetime)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Hosting.IHostApplicationLifetime	hostApplicationLifetime	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

StartAsync(CancellationToken)

Starts the `pi-cli` hosted service asynchronously.

Declaration

```
public async Task StartAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Threading.CancellationToken	cancellationToken	The cancellation token.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

StopAsync(CancellationToken)

Stops the `pi-cli` hosted service asynchronously.

Declaration

```
public Task StopAsync(CancellationToken cancellationToken)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Threading.CancellationToken	cancellationToken	The cancellation token.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Implements

Microsoft.Extensions.Hosting.IHostedService

Interface ICliBuilder

An abstraction of `pi-cli` service builder.

Namespace: `PerpetualIntelligence.Cli.Integration`

Assembly: `PerpetualIntelligence.Cli.dll`

Syntax

```
public interface ICliBuilder
```

Properties

Services

The service collection.

Declaration

```
IServiceCollection Services { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.DependencyInjection.IServiceCollection	

Extension Methods

- `ICliBuilderExtensions.AddArgumentChecker<TMapper, TChecker>(ICliBuilder)`
- `ICliBuilderExtensions.AddAuthentication<TProvider, TAppFactory, TAppCache, TDelegateHandler>(ICliBuilder, String, String, Nullable<Int32>)`
- `ICliBuilderExtensions.AddCliOptions(ICliBuilder)`
- `ICliBuilderExtensions.AddDescriptor<TRunner, TChecker>(ICliBuilder, CommandDescriptor, Boolean, Boolean, Boolean)`
- `ICliBuilderExtensions.AddDescriptorStore<TStore>(ICliBuilder)`
- `ICliBuilderExtensions.AddErrorPublisher<TError, TException>(ICliBuilder)`
- `ICliBuilderExtensions.AddExtractor<TCommand, TArgument>(ICliBuilder)`
- `ICliBuilderExtensions.AddExtractor<TCommand, TArgument, TDefaultArgumentValue>(ICliBuilder)`
- `ICliBuilderExtensions.AddExtractor<TCommand, TArgument, TDefaultArgument, TDefaultArgumentValue>(ICliBuilder)`
- `ICliBuilderExtensions.AddLicensing(ICliBuilder)`
- `ICliBuilderExtensions.AddLicensingClient(ICliBuilder, String, TimeSpan)`
- `ICliBuilderExtensions.AddRouter<TRouter, THandler>(ICliBuilder)`
- `ICliBuilderExtensions.AddStringComparer(ICliBuilder, StringComparison)`

Namespace PerpetualIntelligence.Cli.Licensing

Classes

[License](#)

A `pi-cli` license.

[LicenseChecker](#)

The default [ILicenseChecker](#) for all `pi-cli` features.

[LicenseCheckerContext](#)

The default [ILicenseChecker](#) context.

[LicenseCheckerResult](#)

The default [ILicenseChecker](#) result.

[LicenseExtractor](#)

The default [ILicenseExtractor](#)

[LicenseExtractorContext](#)

The default [ILicenseChecker](#) context.

[LicenseExtractorResult](#)

The default [ILicenseChecker](#) result.

[LicenseLimits](#)

Defines the licensing limits based on the [SaaSPlans](#).

[LicenseProviderResolver](#)

The default [ILicenseProviderResolver](#).

Interfaces

[ILicenseChecker](#)

An abstraction to check the [License](#) object.

[ILicenseExtractor](#)

An abstraction to extract the software [License](#).

[ILicenseProviderResolver](#)

An abstraction to resolve the license provider.

Interface ILicenseChecker

An abstraction to check the [License](#) object.

Inherited Members

[IChecker<LicenseCheckerContext, LicenseCheckerResult>.CheckAsync\(LicenseCheckerContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Licensing](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface ILicenseChecker : IChecker<LicenseCheckerContext, LicenseCheckerResult>
```

Interface ILicenseExtractor

An abstraction to extract the software [License](#).

Inherited Members

[IExtractor<LicenseExtractorContext, LicenseExtractorResult>.ExtractAsync\(LicenseExtractorContext\)](#)

Namespace: [PerpetualIntelligence.Cli.Licensing](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface ILicenseExtractor : IExtractor<LicenseExtractorContext, LicenseExtractorResult>
```

Methods

GetLicenseAsync()

Gets the extracted license asynchronously.

Declaration

```
Task<License> GetLicenseAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< License >	

Interface ILicenseProviderResolver

An abstraction to resolve the license provider.

Namespace: [PerpetualIntelligence.Cli.Licensing](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public interface ILicenseProviderResolver
```

Methods

ResolveAsync(String)

Resolves the provider and returns the provider identifier.

Declaration

```
Task<string> ResolveAsync(string resolveId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	resolveId	The resolve identifier.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	The resolved identifier.

Class License

A `pi-cli` license.

Inheritance

System.Object
System.ComponentModel.License
License

Implements

System.IDisposable

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)

Assembly: `PerpetualIntelligence.Cli.dll`

Syntax

```
public sealed class License : License, IDisposable
```

Constructors

License(String, String, String, String, String, String, LicenseClaimsModel, LicenseLimits)

Initialize a new instance.

Declaration

```
public License(string providerTenantId, string checkMode, string plan, string usage, string licenseKeySource, string licenseKey, LicenseClaimsModel claims, LicenseLimits limits)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	providerTenantId	The license provider tenant id.
System.String	checkMode	The license check mode.
System.String	plan	The license plan.
System.String	usage	The license usage.
System.String	licenseKeySource	

TYPE	NAME	DESCRIPTION
System.String	licenseKey	The license key.
LicenseClaimsModel	claims	The license claims.
LicenseLimits	limits	The license limits.

Properties

CheckMode

The license check mode.

Declaration

```
public string CheckMode { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Claims

The license claims.

Declaration

```
public LicenseClaimsModel Claims { get; }
```

Property Value

TYPE	DESCRIPTION
LicenseClaimsModel	

LicenseKey

The license key.

Declaration

```
public override string LicenseKey { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Overrides

System.ComponentModel.License.LicenseKey

LicenseKeySource

The license key source.

Declaration

```
public string LicenseKeySource { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Limits

The license limits.

Declaration

```
public LicenseLimits Limits { get; }
```

Property Value

TYPE	DESCRIPTION
LicenseLimits	

Plan

The license plan.

Declaration

```
public string Plan { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ProviderId

The license provider tenant id.

Declaration

```
public string ProviderId { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Usage

The license usage.

Declaration

```
public string Usage { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

Dispose()

Disposes the license.

Declaration

```
public override void Dispose()
```

Overrides

System.ComponentModel.License.Dispose()

Implements

System.IDisposable

Class LicenseChecker

The default [ILicenseChecker](#) for all `pi-cli` features.

Inheritance

System.Object
LicenseChecker

Implements

[ILicenseChecker](#)
[IChecker](#)<[LicenseCheckerContext](#), [LicenseCheckerResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class LicenseChecker : ILicenseChecker, IChecker<LicenseCheckerContext, LicenseCheckerResult>
```

Constructors

`LicenseChecker(IEnumerable<CommandDescriptor>, CliOptions, ILogger<LicenseChecker>)`

Initializes a new instance.

Declaration

```
public LicenseChecker(IEnumerable<CommandDescriptor> commandDescriptors, CliOptions cliOptions, ILogger<LicenseChecker> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable< CommandDescriptor >	commandDescriptors	
CliOptions	cliOptions	
Microsoft.Extensions.Logging.ILogger< LicenseChecker >	logger	

Properties

Initialized

Determines whether the checker is initialized.

Declaration

```
public bool Initialized { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Methods

CheckAsync(LicenseCheckerContext)

Checks the licensing context.

Declaration

```
public async Task<LicenseCheckerResult> CheckAsync(LicenseCheckerContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
LicenseCheckerContext	context	The licensing context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< LicenseCheckerResult >	

Exceptions

TYPE	CONDITION
System.NotImplementedException	

Implements

[ILicenseChecker](#)
[IChecker<TContext, TResult>](#)

Class LicenseCheckerContext

The default [ILicenseChecker](#) context.

Inheritance

System.Object
LicenseCheckerContext

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LicenseCheckerContext
```

Constructors

LicenseCheckerContext(License)

Initialize a new instance.

Declaration

```
public LicenseCheckerContext(License license)
```

Parameters

TYPE	NAME	DESCRIPTION
License	license	

Properties

License

The license to check.

Declaration

```
public License License { get; }
```

Property Value

TYPE	DESCRIPTION
License	

Class LicenseCheckerResult

The default [ILicenseChecker](#) result.

Inheritance

System.Object
LicenseCheckerResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LicenseCheckerResult
```

Constructors

LicenseCheckerResult(License)

Initializes a new instance.

Declaration

```
public LicenseCheckerResult(License license)
```

Parameters

TYPE	NAME	DESCRIPTION
License	license	The checked license.

Properties

ArgumentCount

The argument count.

Declaration

```
public long ArgumentCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

CommandGroupCount

The command group count.

Declaration

```
public long CommandGroupCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

License

The valid license.

Declaration

```
public License License { get; }
```

Property Value

TYPE	DESCRIPTION
License	

RootCommandCount

The root command count.

Declaration

```
public long RootCommandCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

SubCommandCount

The sub command count.

Declaration

```
public long SubCommandCount { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int64	

Class LicenseExtractor

The default [ILicenseExtractor](#)

Inheritance

System.Object
LicenseExtractor

Implements

[ILicenseExtractor](#)
[IExtractor](#)<[LicenseExtractorContext](#), [LicenseExtractorResult](#)>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class LicenseExtractor : ILicenseExtractor, IExtractor<LicenseExtractorContext, LicenseExtractorResult>
```

Constructors

LicenseExtractor(ILicenseProviderResolver, CliOptions, IHttpClientFactory)

Initialize a new instance.

Declaration

```
public LicenseExtractor(ILicenseProviderResolver licenseProviderResolver, CliOptions cliOptions, IHttpClientFactory httpClientFactory = null)
```

Parameters

TYPE	NAME	DESCRIPTION
ILicenseProviderResolver	licenseProviderResolver	The license provider resolver.
CliOptions	cliOptions	The configuration options.
System.Net.Http.IHttpClientFactory	httpClientFactory	The optional HTTP client factory

Methods

ExtractAsync(LicenseExtractorContext)

Extracts the [License](#) from the license keys.

Declaration

```
public async Task<LicenseExtractorResult> ExtractAsync(LicenseExtractorContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
LicenseExtractorContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< LicenseExtractorResult >	

GetLicenseAsync()

Gets the extracted license asynchronously.

Declaration

```
public Task<License> GetLicenseAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< License >	

Implements

[ILicenseExtractor](#)

[IExtractor<TContext, TResult>](#)

Class LicenseExtractorContext

The default [ILicenseChecker](#) context.

Inheritance

System.Object

LicenseExtractorContext

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LicenseExtractorContext
```

Constructors

LicenseExtractorContext()

Initialize a new instance.

Declaration

```
public LicenseExtractorContext()
```

Class LicenseExtractorResult

The default [ILicenseChecker](#) result.

Inheritance

System.Object
LicenseExtractorResult

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LicenseExtractorResult
```

Constructors

LicenseExtractorResult(License)

Initializes a new instance.

Declaration

```
public LicenseExtractorResult(License license)
```

Parameters

TYPE	NAME	DESCRIPTION
License	license	The extracted license.

Properties

License

The valid licenses.

Declaration

```
public License License { get; }
```

Property Value

TYPE	DESCRIPTION
License	

Class LicenseLimits

Defines the licensing limits based on the [SaaSPlans](#).

Inheritance

System.Object
LicenseLimits

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public sealed class LicenseLimits
```

Properties

ArgumentLimit

The maximum arguments or options. Defaults to `null` or no limit.

Declaration

```
public long? ArgumentLimit { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int64>	

CommandGroupLimit

The maximum command groups. Defaults to `null` or no limit.

Declaration

```
public int? CommandGroupLimit { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int32>	

DataTypeChecks

Supports the command argument data type checks. Defaults to `null` or no data type checks.

Declaration

```
public string[] DataTypeChecks { get; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

DefaultArgument

Supports the default command argument. Defaults to `false` or no default arguments.

Declaration

```
public bool DefaultArgument { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

DefaultArgumentValue

Supports the default argument value. Defaults to `false` or no default argument value.

Declaration

```
public bool DefaultArgumentValue { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

ErrorHandling

The maximum sub commands. Defaults to `null` or no limit.

Declaration

```
public string[] ErrorHandling { get; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

Plan

The SaaS plan.

Declaration

```
public string Plan { get; }
```

Property Value

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.String	

RedistributionLimit

The maximum sub commands. Defaults to `null` or no redistribution limit.

Declaration

```
public long? RedistributionLimit { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int64>	

RootCommandLimit

The maximum root commands. Defaults to `null` or no limit.

Declaration

```
public int? RootCommandLimit { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int32>	

ServiceImplementations

The maximum sub commands. Defaults to `null` or no redistributions.

Declaration

```
public string[] ServiceImplementations { get; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

Stores

The maximum sub commands. Defaults to `null` or no redistributions.

Declaration

```
public string[] Stores { get; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

StrictDataType

The maximum sub commands. Defaults to `null` or no limit.

Declaration

```
public bool StrictDataType { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

SubCommandLimit

The maximum sub commands. Defaults to `null` or no limit.

Declaration

```
public long? SubCommandLimit { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Int64>	

UnicodeSupport

The maximum sub commands. Defaults to `null` or no limit.

Declaration

```
public string[] UnicodeSupport { get; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

Methods

Create(String)

Creates a new instance of [LicenseLimits](#) based on the specified SaaS plan.

Declaration

```
public static LicenseLimits Create(string saasPlan)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	saasPlan	The SaaS plan.

Returns

TYPE	DESCRIPTION
LicenseLimits	

Class LicenseProviderResolver

The default [ILicenseProviderResolver](#).

Inheritance

System.Object
LicenseProviderResolver

Implements

[ILicenseProviderResolver](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Licensing](#)
Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class LicenseProviderResolver : ILicenseProviderResolver
```

Methods

ResolveAsync(String)

Resolves the provider and returns the provider identifier.

Declaration

```
public Task<string> ResolveAsync(string resolveId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	resolveId	The resolve identifier.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	The resolved identifier.

Implements

[ILicenseProviderResolver](#)

Namespace PerpetualIntelligence.Cli.Services

Classes

[ConsoleHelper](#)

The System.Console helper methods.

Structs

[ConsoleClearPosition](#)

Defines the positioning data to clear the console output.

Struct ConsoleClearPosition

Defines the positioning data to clear the console output.

Inherited Members

- System.ValueType.Equals(System.Object)
- System.ValueType.GetHashCode()
- System.ValueType.ToString()
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetType()
- System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [PerpetualIntelligence.Cli.Services](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public struct ConsoleClearPosition
```

Fields

Left

The left cursor clear position.

Declaration

```
public int Left
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Length

The length of output to clear.

Declaration

```
public int Length
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Top

The top cursor clear position.

Declaration

```
public int Top
```

Field Value

TYPE	DESCRIPTION
System.Int32	

Class ConsoleHelper

The System.Console helper methods.

Inheritance

System.Object
ConsoleHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Services](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
[WriteUnitTest]
public static class ConsoleHelper
```

Methods

ClearOutput(ConsoleClearPosition)

Clears the written output for the specified position.

Declaration

```
public static void ClearOutput(ConsoleClearPosition clearPosition)
```

Parameters

TYPE	NAME	DESCRIPTION
ConsoleClearPosition	clearPosition	The console clear position.

GetClearPosition(Int32)

Gets the [ConsoleClearPosition](#) for the specified output length.

Declaration

```
public static ConsoleClearPosition GetClearPosition(int length)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	length	The output length to clear.

Returns

TYPE	DESCRIPTION
ConsoleClearPosition	ConsoleClearPosition object with System.Console.CursorLeft, System.Console.CursorTop and specified length.

WriteColor(ConsoleColor, String, Object[])

Writes the specified string with specified color to the standard output stream.

Declaration

```
public static void WriteColor(ConsoleColor color, string value, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ConsoleColor	color	
System.String	value	
System.Object[]	args	

Remarks

[WriteColor\(ConsoleColor, String, Object\[\]\)](#) resets the color after writing the string using System.Console.ResetColor() method.

WriteLineColor(ConsoleColor, String, Object[])

Writes the specified string with specified color followed by the current line terminator to the standard output stream.

Declaration

```
public static void WriteLineColor(ConsoleColor color, string value, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.ConsoleColor	color	
System.String	value	
System.Object[]	args	

Remarks

[WriteLineColor\(ConsoleColor, String, Object\[\]\)](#) resets the color after writing the string using System.Console.ResetColor() method.

Namespace PerpetualIntelligence.Cli.Stores.InMemory

Classes

[InMemoryCommandDescriptorStore](#)

The default in-memory [ICommandDescriptorStore](#).

Class InMemoryCommandDescriptorStore

The default in-memory [ICommandDescriptorStore](#).

Inheritance

System.Object
InMemoryCommandDescriptorStore

Implements

[ICommandDescriptorStore](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Cli.Stores.InMemory](#)

Assembly: PerpetualIntelligence.Cli.dll

Syntax

```
public class InMemoryCommandDescriptorStore : ICommandDescriptorStore
```

Constructors

InMemoryCommandDescriptorStore(IStringComparer, IEnumerable<CommandDescriptor>, CliOptions, ILogger<InMemoryCommandDescriptorStore>)

Initialize a new instance.

Declaration

```
public InMemoryCommandDescriptorStore(IStringComparer stringComparer, IEnumerable<CommandDescriptor> commandDescriptors, CliOptions options, ILogger<InMemoryCommandDescriptorStore> logger)
```

Parameters

TYPE	NAME	DESCRIPTION
IStringComparer	stringComparer	The string comparer.
System.Collections.Generic.IEnumerable< CommandDescriptor >	commandDescriptors	The command identities.
CliOptions	options	The configuration options.
Microsoft.Extensions.Logging.ILogger< InMemoryCommandDescriptorStore >	logger	The logger.

Methods

TryFindByIdAsync(String)

Attempts to finds a [CommandDescriptor](#) by its id.

Attempts to find a [CommandDescriptor](#) by its id.

Declaration

```
public Task<TryResultOrError<CommandDescriptor>> TryFindByIdAsync(string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The command id.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

TryFindByNameAsync(String)

Attempts to find a a [CommandDescriptor](#) by its name.

Declaration

```
public Task<TryResultOrError<CommandDescriptor>> TryFindByNameAsync(string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	The command name.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

TryFindByPrefixAsync(String)

Attempts to find a a [CommandDescriptor](#) by its prefix.

Declaration

```
public Task<TryResultOrError<CommandDescriptor>> TryFindByPrefixAsync(string prefix)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	prefix	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

TryMatchByPrefixAsync(String)

Attempts to match a a [CommandDescriptor](#) by its prefix.

Declaration

```
public Task<TryResultOrError<CommandDescriptor>> TryMatchByPrefixAsync(string prefix)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	prefix	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError < CommandDescriptor >>	

Implements

[ICommandDescriptorStore](#)

Namespace PerpetualIntelligence.Protocols

Classes

[Constants](#)

Defines the supported protocols.

Class Constants

Defines the supported protocols.

Inheritance

System.Object
Constants

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public static class Constants
```

Fields

Cli

The Perpetual Intelligence's cli protocol.

Declaration

```
public const string Cli = "cli"
```

Field Value

TYPE	DESCRIPTION
System.String	

CliUrn

The Perpetual Intelligence's cli protocol.

Declaration

```
public const string CliUrn = "urn:oneimlx:cli"
```

Field Value

TYPE	DESCRIPTION
System.String	

OneImlx

The Perpetual Intelligence's oneimlx protocol.

Declaration

```
public const string OneImlx = "oneimlx"
```

Field Value

TYPE	DESCRIPTION
System.String	

OneImlxUrn

The Perpetual Intelligence's `oneimlx` protocol.

Declaration

```
public const string OneImlxUrn = "urn:oneimlx"
```

Field Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Protocols.Abstractions

Interfaces

[IChecker<TContext, TResult>](#)

An abstraction of a context specific checker.

[IExtractor<TContext, TResult>](#)

An abstraction of a context specific extractor.

[IFormatter<TContext, TResult>](#)

An abstraction of a context specific formatter.

[IHandler<TContext, TResult>](#)

An abstraction of a context specific handler.

[IMapper<TContext, TResult>](#)

An abstraction of a context specific mapper.

[IProcessor<TContext>](#)

An abstraction of context specific processor.

[IProducer<TContext, TResult>](#)

An abstraction of a context specific result producer.

[IProvider<TContext, TResult>](#)

An abstraction of context specific provider.

[IPublisher<TContext, TResult>](#)

An abstraction of context specific publisher.

[IPublisherNoResult<TContext>](#)

An abstraction of a context specific publisher with no explicit result.

[IRecorder<TContext, TResult>](#)

An abstraction of a context specific recorder.

[IRouter<TContext, TResult, THandler>](#)

An abstraction of a context specific router.

[IRunner<TContext, TResult>](#)

An abstraction of a context specific runner.

[ISubscriber<TContext, TResult>](#)

An abstraction of a context specific subscriber.

Interface IChecker<TContext, TResult>

An abstraction of a context specific checker.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IChecker<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The checker context.
TResult	The checker result.

Methods

CheckAsync(TContext)

Checks the context asynchronously.

Declaration

```
Task<TResult> CheckAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The checker context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The checker result.

Interface IExtractor<TContext, TResult>

An abstraction of a context specific extractor.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IExtractor<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The extractor context.
TResult	The extractor result.

Methods

ExtractAsync(TContext)

Extracts from a context asynchronously.

Declaration

```
Task<TResult> ExtractAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The extractor context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The extractor result.

Interface IFormatter<TContext, TResult>

An abstraction of a context specific formatter.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IFormatter<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The checker context.
TResult	The checker result.

Methods

FormatAsync(TContext)

Formats the context asynchronously.

Declaration

```
Task<TResult> FormatAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The formatter context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The formatter result.

Interface IHandler<TContext, TResult>

An abstraction of a context specific handler.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IHandler<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The handler context.
TResult	The handler result.

Methods

HandleAsync(TContext)

Handles the context.

Declaration

```
Task<TResult> HandleAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The handler context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The handler result.

Interface IMapper<TContext, TResult>

An abstraction of a context specific mapper.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IMapper<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	
TResult	

Methods

MapAsync(TContext)

Maps asynchronously.

Declaration

```
Task<TResult> MapAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The producer context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

Interface IProcessor<TContext>

An abstraction of context specific processor.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IProcessor<TContext>
    where TContext : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The processor context.

Methods

ProcessAsync(TContext)

Processes asynchronously.

Declaration

```
Task ProcessAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The processor context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Interface IProducer<TContext, TResult>

An abstraction of a context specific result producer.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IProducer<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	
TResult	

Methods

ProduceAsync(TContext)

Produces the result asynchronously.

Declaration

```
Task<TResult> ProduceAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The producer context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

Interface IProvider<TContext, TResult>

An abstraction of context specific provider.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IProvider<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The provider context.
TResult	The provider result.

Methods

ProvideAsync(TContext)

Provides asynchronously.

Declaration

```
Task<TResult> ProvideAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The provider context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

Interface IPublisher<TContext, TResult>

An abstraction of context specific publisher.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IPublisher<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The publisher context.
TResult	The publisher context.

Methods

ProcessAsync(TContext)

Publishes asynchronously.

Declaration

```
Task<TResult> ProcessAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The publisher context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

Interface IPublisherNoResult<TContext>

An abstraction of a context specific publisher with no explicit result.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IPublisherNoResult<TContext>
    where TContext : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The publisher context.

Remarks

The [IPublisherNoResult<TContext>](#) does not return any result during publish. See [IPublisher<TContext, TResult>](#) for publisher with result.

Methods

PublishAsync(TContext)

Publishes asynchronously.

Declaration

```
Task PublishAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The publisher context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Interface IRecorder<TContext, TResult>

An abstraction of a context specific recorder.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IRecorder<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The recorder context.
TResult	The recorder result.

Methods

RecordAsync<TEvent>(TContext)

Records asynchronously.

Declaration

```
Task<TResult> RecordAsync<TEvent>(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The recorder context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The recorder result.

Type Parameters

NAME	DESCRIPTION
TEvent	

Interface IRouter<TContext, TResult, THandler>

An abstraction of a context specific router.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: [PerpetualIntelligence.Protocols.dll](#)

Syntax

```
public interface IRouter<TContext, TResult, THandler>
    where TContext : class where TResult : class where THandler : class
```

Type Parameters

NAME	DESCRIPTION
TContext	
TResult	
THandler	

Methods

RouteAsync(TContext)

Routes the request asynchronously.

Declaration

```
Task<TResult> RouteAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The router context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The router result.

TryFindHandlerAsync(TContext)

Finds a matching route handler for a request asynchronously.

Declaration

```
Task<TryResultOrError<THandler>> TryFindHandlerAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
TContext	context	The router context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TryResultOrError<THandler>>	The request handler.

Interface IRunner<TContext, TResult>

An abstraction of a context specific runner.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IRunner<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The runner context.
TResult	The runner result.

Methods

RunAsync(TContext)

Runs a context asynchronously.

Declaration

```
Task<TResult> RunAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The runner context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The runner result.

Interface ISubscriber<TContext, TResult>

An abstraction of a context specific subscriber.

Namespace: [PerpetualIntelligence.Protocols.Abstractions](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface ISubscriber<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The subscriber context.
TResult	The subscriber result.

Methods

SubscribeAsync(TContext)

Subscribes asynchronously.

Declaration

```
Task<TResult> SubscribeAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The subscriber context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	The subscriber result.

Namespace

PerpetualIntelligence.Protocols.Abstractions.Authorization

Interfaces

[IAuthHandler<TContext, TResult>](#)

An abstraction of a generic `OAuth` and `OpenID Connect` request handler.

Interface IAuthHandler<TContext, TResult>

An abstraction of a generic OAuth and OpenID Connect request handler.

Inherited Members

IHandler<TContext, TResult>.HandleAsync(TContext)

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Authorization](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IAuthHandler<TContext, TResult> : IHandler<TContext, TResult> where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	The handler context.
TResult	The handler result.

Remarks

An authorization server typically provides both authorization and authentication capabilities.

Properties

Disabled

Determines whether the handler is disabled.

Declaration

```
bool? Disabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	true is the handler is disabled, otherwise false.

EndpointId

Identifies an endpoint for this handler.

Declaration

```
string EndpointId { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	The endpoint identifier.

Redirect

Determines whether the handler issues a redirect callback.

Declaration

```
bool? Redirect { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	<code>true</code> is the handler issues a redirect callback, otherwise <code>false</code> .

Namespace

PerpetualIntelligence.Protocols.Abstractions.Comparers

Interfaces

[IClaimComparer](#)

An abstraction of System.Security.Claims.Claim comparer.

[IStringComparer](#)

An abstraction to compare System.String.

Interface IClaimComparer

An abstraction of System.Security.Claims.Claim comparer.

Inherited Members

System.Collections.Generic.IEqualityComparer<System.Security.Claims.Claim>.Equals(System.Security.Claims.Claim, System.Security.Claims.Claim)

System.Collections.Generic.IEqualityComparer<System.Security.Claims.Claim>.GetHashCode(System.Security.Claims.Claim)

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Comparers](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IClaimComparer : IEqualityComparer<Claim>
```

Interface IStringComparer

An abstraction to compare System.String.

Inherited Members

System.Collections.Generic.IEqualityComparer<System.String>.Equals(System.String, System.String)
System.Collections.Generic.IEqualityComparer<System.String>.GetHashCode(System.String)

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Comparers](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IStringComparer : IEqualityComparer<string>
```

Properties

Comparison

System.StringComparison used to compare string.

Declaration

```
StringComparison Comparison { get; }
```

Property Value

TYPE	DESCRIPTION
System.StringComparison	

Namespace

PerpetualIntelligence.Protocols.Abstractions.Concurrency

Interfaces

[IThreadLock<T>](#)

An abstraction to lock or release the concurrent thread execution.

Interface IThreadLock<T>

An abstraction to lock or release the concurrent thread execution.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Concurrency](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IThreadLock<T>
```

Type Parameters

NAME	DESCRIPTION
T	The type requesting thread lock.

Methods

LockAsync(Int32)

Asynchronously lock the current thread for the specified time interval.

Declaration

```
Task<bool> LockAsync(int timeout)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	timeout	The timeout in millisecond.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Boolean>	<code>true</code> if the lock is acquired, <code>false</code> otherwise.

ReleaseAsync()

Asynchronously release the locked thread.

Declaration

```
Task ReleaseAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	System.Threading.Tasks.Task instance.

Namespace

PerpetualIntelligence.Protocols.Abstractions.Events

Interfaces

[IEvent](#)

The abstraction of an event in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

[IEventPublisher<TContext, TResult>](#)

The abstraction to publish an [IEvent](#) in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

[IEventRecorder<TContext, TResult>](#)

The abstraction to record an [IEvent](#) in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

[IEventSubscriber<TContext, TResult>](#)

The abstraction to subscribe an [IEvent](#) in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

Interface IEvent

The abstraction of an event in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Events](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IEvent
```

Properties

Name

The unique event name.

Declaration

```
string Name { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

See Also

- [IEventSubscriber](#)<TContext, TResult>
- [IEventPublisher](#)<TContext, TResult>
- [IEventRecorder](#)<TContext, TResult>

Interface IEventPublisher<TContext, TResult>

The abstraction to publish an [IEvent](#) in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Events](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IEventPublisher<TContext, TResult>
    where TContext : class
```

Type Parameters

NAME	DESCRIPTION
TContext	
TResult	

Methods

CanPublishAsync(TContext)

Determines whether the [IEventPublisher<TContext, TResult>](#) can publish the specified event.

Declaration

```
Task<bool> CanPublishAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The event context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Boolean>	

PublishAsync(TContext)

Publishes the specified event asynchronously.

Declaration

```
Task<TResult> PublishAsync(TContext evt)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	evt	The event context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

See Also

- [IEvent](#)
- [IEventSubscriber<TContext, TResult>](#)
- [IEventRecorder<TContext, TResult>](#)
- [Implementing event-based communication between micro-services \(integration events\)](#)

Interface IEventRecorder<TContext, TResult>

The abstraction to record an [IEvent](#) in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Events](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IEventRecorder<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	
TResult	

Methods

CanRecordAsync(TContext)

Determines whether the [IEventRecorder<TContext, TResult>](#) can record the specified event.

Declaration

```
Task<bool> CanRecordAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The event context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Boolean>	

RecordAsync<TEvent>(TContext)

Records the specified event asynchronously.

Declaration

```
Task<TResult> RecordAsync<TEvent>(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The event context to record.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

Type Parameters

NAME	DESCRIPTION
TEvent	

See Also

- [IEvent](#)
- [IEventPublisher<TContext, TResult>](#)
- [IEventSubscriber<TContext, TResult>](#)

Interface IEventSubscriber<TContext, TResult>

The abstraction to subscribe an [IEvent](#) in the publish/subscribe-style communication without requiring the components to explicitly be aware of each other.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Events](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IEventSubscriber<TContext, TResult>
    where TContext : class where TResult : class
```

Type Parameters

NAME	DESCRIPTION
TContext	
TResult	

Methods

CanSubscribeAsync(TContext)

Determines whether the [IEventSubscriber<TContext, TResult>](#) can subscribe the specified event.

Declaration

```
Task<bool> CanSubscribeAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The event context

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Boolean>	

SubscribeAsync(TContext)

Subscribes the specified event asynchronously.

Declaration

```
Task<TResult> SubscribeAsync(TContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	The event context subscribe.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

See Also

[IEvent](#)

[IEventPublisher](#)<TContext, TResult>

[IEventRecorder](#)<TContext, TResult>

[Implementing event-based communication between micro-services \(integration events\)](#)

Namespace

PerpetualIntelligence.Protocols.Abstractions.Stores

Interfaces

[IEntity](#)

The abstraction of a persisted entity.

[IEntityTenancy](#)

Provides the tenancy information.

[IMarketplaceEntity](#)

An abstraction of a multi-tenant entity that establishes a commercial relationship between businesses, precisely provider and consumer. It is a typical model when a business or a company sells its products or services directly to other companies. The customers of a product or service are the users, collaborators, or employees within the consumer business.

[IStore<TSContext, TMContext, TSResult, TMResult>](#)

The abstraction of a back-end store.

[IStoreMultipleResult<TEntity>](#)

The abstraction of a store result that yields multiple entities.

[IStoreSingleResult<TEntity>](#)

The abstraction of a store result that yields a single entity.

[ITrackedStore<TSContext, TMContext, TSResult, TMResult>](#)

The abstraction of a back-end store that supports tracked operation.

Interface IEntity

The abstraction of a persisted entity.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Stores](#)

Assembly: [PerpetualIntelligence.Protocols.dll](#)

Syntax

```
public interface IEntity
```

Properties

ConcurrencyStamp

The optimistic concurrency stamp during updates.

Declaration

```
string ConcurrencyStamp { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

CreateStamp

The create stamp expressed in Coordinated Universal Time (UTC).

Declaration

```
DateTimeOffset? CreateStamp { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable< DateTimeOffset >	

DeleteStamp

Delete request stamp of the entity expressed in the Coordinated Universal Time (UTC).

Declaration

```
DateTimeOffset? DeleteStamp { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable< DateTimeOffset >	

Disabled

The locked state.

Declaration

```
bool? Disabled { get; }
```


Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Locked

The locked state.

Declaration

```
bool? Locked { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

Name

The entity name.

Declaration

```
string Name { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Properties

The additional properties.

Declaration

```
Dictionary<string, string> Properties { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.String>	

Tags

The tags.

Declaration

```
string[] Tags { get; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

UpdateStamp

The update time-stamp expressed in Coordinated Universal Time (UTC).

Declaration

```
DateTimeOffset? UpdateStamp { get; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable< DateTimeOffset >	

See Also

[ITrackedStore](#)<TSContext, TMContext, TSResult, TMResult>

Interface IEntityTenancy

Provides the tenancy information.

Inherited Members

- [IEntity.ConcurrencyStamp](#)
- [IEntity.CreateStamp](#)
- [IEntity.DeleteStamp](#)
- [IEntity.Disabled](#)
- [IEntity.Locked](#)
- [IEntity.Name](#)
- [IEntity.Properties](#)
- [IEntity.Tags](#)
- [IEntity.UpdateStamp](#)

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Stores](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IEntityTenancy : IEntity
```

Properties

TenantId

The tenant identifier.

Declaration

```
string TenantId { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

TenantMode

The tenant mode.

Declaration

```
string TenantMode { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Interface IMarketplaceEntity

An abstraction of a multi-tenant entity that establishes a commercial relationship between businesses, precisely provider and consumer. It is a typical model when a business or a company sells its products or services directly to other companies. The customers of a product or service are the users, collaborators, or employees within the consumer business.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Stores](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IMarketplaceEntity
```

Properties

ConsumerTenantId

The consumer tenant id.

Declaration

```
string ConsumerTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ProviderTenantId

The provider tenant id.

Declaration

```
string ProviderTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

PublisherTenantId

The publisher tenant id.

Declaration

```
string PublisherTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Interface IStore<TSContext, TMContext, TSResult, TMResult>

The abstraction of a back-end store.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Stores](#)

Assembly: [PerpetualIntelligence.Protocols.dll](#)

Syntax

```
public interface IStore<TSContext, TMContext, TSResult, TMResult>
    where TSContext : class where TMContext : class where TSResult : class where TMResult : class
```

Type Parameters

NAME	DESCRIPTION
TSContext	The single entity context.
TMContext	The multiple entities context.
TSResult	The single entity result.
TMResult	The multiple entities result.

Methods

CreateAsync(TSContext)

Creates an entity in the back-end store asynchronously.

Declaration

```
Task<TSResult> CreateAsync(TSContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The create context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The create result.

DeleteAsync(TSContext)

Deletes an entity from the back-end store asynchronously.

Declaration

```
Task<TSResult> DeleteAsync(TSContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The delete context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The delete result.

FindAsync(TSContext)

Finds an entity in the back-end store asynchronously.

Declaration

Task<TSResult> FindAsync(TSContext context)

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The find context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The find result.

ListAsync(TMContext)

Lists the entities from the back-end store asynchronously.

Declaration

Task<TMResult> ListAsync(TMContext context)

Parameters

TYPE	NAME	DESCRIPTION
TMContext	context	The list context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TMResult>	The list result.

LockAsync(TSContext)

Locks an entity in the back-end store asynchronously.

Declaration

```
Task<TSResult> LockAsync(TSContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The lock context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The lock result.

PutAsync(TSContext)

Creates or updates an entity in the back-end store asynchronously.

Declaration

```
Task<TSResult> PutAsync(TSContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The put context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The put result.

ReplaceAsync(TSContext)

Replaces an entity in the back-end store asynchronously.

Declaration

```
Task<TSResult> ReplaceAsync(TSContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The replace context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The replace result.

Remarks

Replace is a delete and create operation of the same entity.

UnlockAsync(TSContext)

Unlocks an entity in the back-end store asynchronously.

Declaration

Task<TSResult> UnlockAsync(TSContext context)

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The unlock context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The unlock result.

UpdateAsync(TSContext)

Updates an entity in the back-end store asynchronously.

Declaration

Task<TSResult> UpdateAsync(TSContext context)

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The update context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The update result.

See Also

[IEntity](#)

[IStoreSingleResult<TEntity>](#)

[IStoreMultipleResult<TEntity>](#)

[ITrackedStore](#)<TSContext, TMContext, TSResult, TMResult>

Interface IStoreMultipleResult<TEntity>

The abstraction of a store result that yields multiple entities.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Stores](#)

Assembly: [PerpetualIntelligence.Protocols.dll](#)

Syntax

```
public interface IStoreMultipleResult<TEntity>
    where TEntity : IEntity
```

Type Parameters

NAME	DESCRIPTION
TEntity	

Properties

ContinuationToken

The continuation token to get the next set of entities.

Declaration

```
string ContinuationToken { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Result

The store entities.

Declaration

```
TEntity[] Result { get; }
```

Property Value

TYPE	DESCRIPTION
TEntity[]	

Interface IStoreSingleResult<TEntity>

The abstraction of a store result that yields a single entity.

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Stores](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public interface IStoreSingleResult<TEntity>
    where TEntity : IEntity
```

Type Parameters

NAME	DESCRIPTION
TEntity	

Properties

Result

The single entity.

Declaration

```
TEntity Result { get; }
```

Property Value

TYPE	DESCRIPTION
TEntity	

See Also

[IStoreMultipleResult<TEntity>](#)

Interface ITrackedStore<TSContext, TMContext, TSResult, TMResult>

The abstraction of a back-end store that supports tracked operation.

Inherited Members

- [IStore<TSContext, TMContext, TSResult, TMResult>.CreateAsync\(TSContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.DeleteAsync\(TSContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.FindAsync\(TSContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.ListAsync\(TMContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.LockAsync\(TSContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.PutAsync\(TSContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.ReplaceAsync\(TSContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.UnlockAsync\(TSContext\)](#)
- [IStore<TSContext, TMContext, TSResult, TMResult>.UpdateAsync\(TSContext\)](#)

Namespace: [PerpetualIntelligence.Protocols.Abstractions.Stores](#)

Assembly: [PerpetualIntelligence.Protocols.dll](#)

Syntax

```
public interface ITrackedStore<TSContext, TMContext, TSResult, TMResult> : IStore<TSContext, TMContext, TSResult, TMResult> where TSContext : class where TMContext : class where TSResult : class where TMResult : class
```

Type Parameters

NAME	DESCRIPTION
TSContext	The single entity context.
TMContext	The multiple entity context.
TSResult	The single entity result.
TMResult	The multiple entities result.

Methods

ClearDeleteAsync(TSContext)

Removes or clears the entity deletion in the back-end store asynchronously.

Declaration

```
Task<TSResult> ClearDeleteAsync(TSContext context)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
TSContext	context	The clear delete context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The clear delete result.

MarkDeleteAsync(TSContext)

Marks an entity for deletion in the back-end store asynchronously.

Declaration

```
Task<TSResult> MarkDeleteAsync(TSContext context)
```

Parameters

TYPE	NAME	DESCRIPTION
TSContext	context	The mark delete context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TSResult>	The mark delete result.

See Also

[IStore](#)<TSContext, TMContext, TSResult, TMResult>

Namespace PerpetualIntelligence.Protocols.Authorization

Classes

[AppTypes](#)

The well-known app types.

[ClaimTypes](#)

The well known claims.

[HeaderKeys](#)

The well known header keys.

[Policies](#)

The authorization policies.

[WellKnownBaseAddress](#)

Defines a well known base address with associated HTTP client name.

Interfaces

[IAppIdentifiers](#)

An abstractions of application identifiers.

Class AppTypes

The well-known app types.

Inheritance

System.Object
AppTypes

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Authorization](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class AppTypes
```

Fields

CliApplication

The cli application.

Declaration

```
public const string CliApplication = "urn:oneimlx:apps:cliapp"
```

Field Value

TYPE	DESCRIPTION
System.String	

ConsumerPortal

The consumer portal.

Declaration

```
public const string ConsumerPortal = "urn:oneimlx:apps:consumerportal"
```

Field Value

TYPE	DESCRIPTION
System.String	

PublisherPortal

The publisher portal.

Declaration

```
public const string PublisherPortal = "urn:oneimlx:apps:publisherportal"
```

Field Value

TYPE	DESCRIPTION
System.String	

Undefined

The publisher portal.

Declaration

```
public const string Undefined = "urn:oneimlx:apps:undefined"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class ClaimTypes

The well known claims.

Inheritance

System.Object
ClaimTypes

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Authorization](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class ClaimTypes
```

Fields

ConsumerObjectId

Indicates an `acr` consumer claim.

Declaration

```
public const string ConsumerObjectId = "urn:oneimlx:claims:coid"
```

Field Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantId

Indicates an `acr` consumer claim.

Declaration

```
public const string ConsumerTenantId = "urn:oneimlx:claims:ctid"
```

Field Value

TYPE	DESCRIPTION
System.String	

ProviderTenantId

Indicates an `acr` provider claim.

Declaration

```
public const string ProviderTenantId = "urn:oneimlx:claims:rtid"
```

Field Value

TYPE	DESCRIPTION
System.String	

PublisherTenantId

Indicates an `acr` publisher claim.

Declaration

```
public const string PublisherTenantId = "urn:oneimlx:claims:ptid"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class HeaderKeys

The well known header keys.

Inheritance

System.Object
HeaderKeys

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Authorization](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class HeaderKeys
```

Fields

AppId

The `x-oneimlx-app-id` header key.

Declaration

```
public const string AppId = "x-oneimlx-app-id"
```

Field Value

TYPE	DESCRIPTION
System.String	

AppType

The `x-oneimlx-app-type` header key.

Declaration

```
public const string AppType = "x-oneimlx-app-type"
```

Field Value

TYPE	DESCRIPTION
System.String	

LoginObjectId

The `x-oneimlx-login-objectid` header key.

Declaration

```
public const string LoginObjectId = "x-oneimlx-login-object-id"
```

Field Value

TYPE	DESCRIPTION
System.String	

LoginTenantId

The `x-oneimlx-login-tenantid` header key.

Declaration

```
public const string LoginTenantId = "x-oneimlx-login-tenant-id"
```

Field Value

TYPE	DESCRIPTION
System.String	

Interface IAppIdentifiers

An abstractions of application identifiers.

Namespace: [PerpetualIntelligence.Protocols.Authorization](#)

Assembly: [PerpetualIntelligence.Protocols.dll](#)

Syntax

```
public interface IAppIdentifiers
```

Methods

DefaultAsync(ClaimsPrincipal)

Sets the [IAppIdentifiers](#) service from the System.Security.Claims.ClaimsPrincipal. For client apps this can be used for singleton state container. For server apps this can be a no-ops as the service implementation will be transient or scoped.

Declaration

```
Task DefaultAsync(ClaimsPrincipal claimsPrincipal)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Security.Claims.ClaimsPrincipal	claimsPrincipal	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

GetAppIdAsync()

Gets the client app id.

Declaration

```
Task<string> GetAppIdAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetAppTypeAsync()

Gets the client app type.

Declaration

```
Task<string> GetAppTypeAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetConsumerObjectIdAsync()

Gets the login tenant id.

Declaration

```
Task<string> GetConsumerObjectIdAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetConsumerTenantIdAsync()

Gets the login tenant id.

Declaration

```
Task<string> GetConsumerTenantIdAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetLoginObjectIdAsync()

Gets the login object id.

Declaration

```
Task<string> GetLoginObjectIdAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetLoginTenantIdAsync()

Gets the login tenant id.

Declaration

```
Task<string> GetLoginTenantIdAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetProviderTenantIdAsync()

Gets the provider tenant id.

Declaration

```
Task<string> GetProviderTenantIdAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

GetPublisherTenantIdAsync()

Gets the publisher tenant id.

Declaration

```
Task<string> GetPublisherTenantIdAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.String>	

InitializeAsync(ClaimsPrincipal)

Initializes the [IApplIdentifiers](#) service from the System.Security.Claims.ClaimsPrincipal. For client apps this can be used for singleton state container. For server apps this can be a no-ops as the service implementation will be transient or scoped.

Declaration

```
Task InitializeAsync(ClaimsPrincipal claimsPrincipal)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Security.Claims.ClaimsPrincipal	claimsPrincipal	The claims principal.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

IsInitializedAsync(ClaimsPrincipal)

Determines whether the [IApplIdentifiers](#) is initialized.

Declaration

```
Task<bool> IsInitializedAsync(ClaimsPrincipal claimsPrincipal)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Security.Claims.ClaimsPrincipal	claimsPrincipal	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Boolean>	

PopulateAsync(ClaimsPrincipal)

Populates the System.Security.Claims.ClaimsPrincipal with claims from [IApplidentifiers](#) service.

Declaration

Task PopulateAsync(ClaimsPrincipal claimsPrincipal)

Parameters

TYPE	NAME	DESCRIPTION
System.Security.Claims.ClaimsPrincipal	claimsPrincipal	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

ResetAsync()

Resets the [IApplidentifiers](#) service for the next login.

Declaration

Task ResetAsync()

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Class Policies

The authorization policies.

Inheritance

System.Object
Policies

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Authorization](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class Policies
```

Fields

Consumer

Consumer portal authorization policy.

Declaration

```
public const string Consumer = "urn:oneimlx:policies:consumer"
```

Field Value

TYPE	DESCRIPTION
System.String	

Publisher

Publisher portal authorization policy.

Declaration

```
public const string Publisher = "urn:oneimlx:policies:publisher"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class WellKnownBaseAddress

Defines a well known base address with associated HTTP client name.

Inheritance

System.Object
WellKnownBaseAddress

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Authorization](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class WellKnownBaseAddress
```

Constructors

WellKnownBaseAddress(String, String)

Initialize a new instance.

Declaration

```
public WellKnownBaseAddress(string name, string baseAddress)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	The HTTP client name.
System.String	baseAddress	The base address.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

BaseAddress

The base address.

Declaration

```
public string BaseAddress { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Name

The HTTP client name.

Declaration

```
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Namespace

PerpetualIntelligence.Protocols.Defaults.Comparers

Classes

[OrderIndependentComparer](#)

The default order independent comparer makes the order of values irrelevant. E.g., the "code id_token token" is equivalent to "code token id_token".

Class OrderIndependentComparer

The default order independent comparer makes the order of values irrelevant. E.g., the "code id_token token" is equivalent to "code token id_token".

Inheritance

System.Object
OrderIndependentComparer

Implements

System.Collections.Generic.IEqualityComparer<System.String>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Defaults.Comparers](#)

Assembly: PerpetualIntelligence.Protocols.Defaults.dll

Syntax

```
public class OrderIndependentComparer : IEqualityComparer<string>
```

Methods

Equals(String, String)

Determines whether the specified values are equal.

Declaration

```
public bool Equals(string x, string y)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	x	The string to compare to <code>y</code> .
System.String	y	The string to compare to <code>x</code> .

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the specified values are equal; otherwise, <code>false</code> .

GetHashCode(String)

Returns a hash code for the value.

Declaration

```
public int GetHashCode(string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	The value for which a hash code is to be returned.

Returns

TYPE	DESCRIPTION
System.Int32	A hash code for the value, suitable for use in hashing algorithms and data structures like a hash table.

Implements

System.Collections.Generic.IEqualityComparer<T>

Namespace

PerpetualIntelligence.Protocols.Defaults.Concurrency

Classes

[SemaphoreSlimThreadLock<T>](#)

The default [IThreadLock<T>](#) using System.Threading.SemaphoreSlim.

Class SemaphoreSlimThreadLock<T>

The default [IThreadLock<T>](#) using System.Threading.SemaphoreSlim.

Inheritance

System.Object
SemaphoreSlimThreadLock<T>

Implements

[IThreadLock<T>](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Defaults.Concurrency](#)

Assembly: PerpetualIntelligence.Protocols.Defaults.dll

Syntax

```
public class SemaphoreSlimThreadLock<T> : IThreadLock<T>
```

Type Parameters

NAME	DESCRIPTION
T	The type requesting the thread lock.

Methods

LockAsync(Int32)

Asynchronously lock the current thread for the specified time interval.

Declaration

```
public Task<bool> LockAsync(int timeout)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	timeout	The timeout in millisecond.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<System.Boolean>	<code>true</code> if the lock is acquired, <code>false</code> otherwise.

ReleaseAsync()

Asynchronously release the locked thread.

Declaration

```
public Task ReleaseAsync()
```

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	System.Threading.Tasks.Task instance.

Implements

[IThreadLock<T>](#)

Namespace PerpetualIntelligence.Protocols.Licensing

Classes

[SaaSCheckModes](#)

Defines the standard SaaS check modes.

[SaaSKeySources](#)

The SaaS key sources.

[SaaSPlans](#)

Defines the standard SaaS pricing plans.

[SaaSProviders](#)

Defines the standard SaaS providers.

[SaaSUsages](#)

The SaaS usage.

Class SaaSCheckModes

Defines the standard SaaS check modes.

Inheritance

System.Object

SaaSCheckModes

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class SaaSCheckModes
```

Fields

Offline

The license check is done offline via signed public keys. This is not yet supported.

Declaration

```
public const string Offline = "urn:oneimlx:lic:saascmode:offline"
```

Field Value

TYPE	DESCRIPTION
System.String	

OfflineThenOnline

First check [Offline](#) then [Online](#). This is not yet supported.

Declaration

```
public const string OfflineThenOnline = "urn:oneimlx:lic:saascmode:offthenon"
```

Field Value

TYPE	DESCRIPTION
System.String	

Online

The license check is done online via REST API.

Declaration

```
public const string Online = "urn:oneimlx:lic:saascmode:online"
```

Field Value

TYPE	DESCRIPTION
System.String	

OnlineThenOffline

First check [Online](#) then [Offline](#). This is not yet supported.

Declaration

```
public const string OnlineThenOffline = "urn:oneimlx:lic:saascmode:onthenoff"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

IsValid(String)

Determines if the check mode is valid.

Declaration

```
public static bool IsValid(string plan)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	plan	

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the plan is valid; otherwise, <code>false</code> .

Class SaaSKeySources

The SaaS key sources.

Inheritance

System.Object
SaaSKeySources

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public sealed class SaaSKeySources
```

Fields

JsonFile

The key source is a json file.

Declaration

```
public const string JsonFile = "urn:oneimlx:lic:ksource:jsonfile"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

IsValid(String)

Determines if the key source is valid.

Declaration

```
public static bool IsValid(string keySource)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	keySource	

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the key source is valid; otherwise, <code>false</code> .

Class SaaSPlans

Defines the standard SaaS pricing plans.

Inheritance

System.Object
SaaSPlans

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class SaaSPlans
```

Fields

Community

The community SaaS pricing plan.

Declaration

```
public const string Community = "urn:oneimlx:lic:saasplan:community"
```

Field Value

TYPE	DESCRIPTION
System.String	

Custom

The custom SaaS pricing plan.

Declaration

```
public const string Custom = "urn:oneimlx:lic:saasplan:custom"
```

Field Value

TYPE	DESCRIPTION
System.String	

Enterprise

The enterprise SaaS pricing plan.

Declaration

```
public const string Enterprise = "urn:oneimlx:lic:saasplan:enterprise"
```

Field Value

TYPE	DESCRIPTION
System.String	

ISV

The ISV SaaS pricing plan.

Declaration

```
public const string ISV = "urn:oneimlx:lic:saasplan:isv"
```

Field Value

TYPE	DESCRIPTION
System.String	

ISVU

The ISV unlimited SaaS pricing plan.

Declaration

```
public const string ISVU = "urn:oneimlx:lic:saasplan:isvu"
```

Field Value

TYPE	DESCRIPTION
System.String	

Micro

The solo or micro SaaS pricing plan.

Declaration

```
public const string Micro = "urn:oneimlx:lic:saasplan:micro"
```

Field Value

TYPE	DESCRIPTION
System.String	

None

The neutral or no SaaS pricing plan.

Declaration

```
public const string None = "urn:oneimlx:lic:saasplan:none"
```

Field Value

TYPE	DESCRIPTION
System.String	

SMB

The SMB SaaS pricing plan.

Declaration

```
public const string SMB = "urn:oneimlx:lic:saasplan:smb"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

IsValid(String)

Determines if the plan is valid.

Declaration

```
public static bool IsValid(string plan)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	plan	

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the plan is valid; otherwise, <code>false</code> .

Class SaaSProviders

Defines the standard SaaS providers.

Inheritance

System.Object
SaaSProviders

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class SaaSProviders
```

Fields

PerpetualIntelligence

The registered licensing provider hosted by `Perpetual Intelligence L.L.C.`

Declaration

```
public const string PerpetualIntelligence = "urn:oneimlx:lic:saaspvdr:pi"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

IsValid(String)

Determines if the provider is valid.

Declaration

```
public static bool IsValid(string pvdr)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	pvdr	

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Boolean	true if the plan is valid; otherwise, false.

Class SaaSUsages

The SaaS usage.

Inheritance

System.Object
SaaSUsages

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class SaaSUsages
```

Fields

CommercialBusiness

The commercial usage for an organization or business account.

Declaration

```
public const string CommercialBusiness = "urn:oneimlx:lic:saasusage:org"
```

Field Value

TYPE	DESCRIPTION
System.String	

CommercialPersonal

The commercial usage for a personal or social account.

Declaration

```
public const string CommercialPersonal = "urn:oneimlx:lic:saasusage:per"
```

Field Value

TYPE	DESCRIPTION
System.String	

Educational

The community usage for educational account.

Declaration

```
public const string Educational = "urn:oneimlx:lic:saasusage:edu"
```

Field Value

TYPE	DESCRIPTION
System.String	

None

The neutral or no SaaS usage plan.

Declaration

```
public const string None = "urn:oneimlx:lic:saasusage:none"
```

Field Value

TYPE	DESCRIPTION
System.String	

RnD

The community test and demo usage for any account.

Declaration

```
public const string RnD = "urn:oneimlx:lic:saasusage:rnd"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

IsValid(String)

Determines if the usage is valid.

Declaration

```
public static bool IsValid(string usage)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	usage	

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the usage is valid; otherwise, <code>false</code> .

Namespace PerpetualIntelligence.Protocols.Licensing.Models

Classes

[LicenseCheckModel](#)

The `jws` B2B keys check model.

[LicenseClaimsModel](#)

Defines the licensing claims model.

[LicenseKeyJsonFileModel](#)

The `jws` B2B keys download model.

[LicenseKeysModel](#)

The `jws` B2B keys model.

[LicenseProvisioningModel](#)

The license provisioning model.

Class LicenseCheckModel

The `jws` B2B keys check model.

Inheritance

System.Object
LicenseCheckModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing.Models](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class LicenseCheckModel
```

Properties

AuthorizedApplicationId

The Authorized application. This is also one of the `auth_apps` claim value.

Declaration

```
[JsonPropertyName("authorized_application_id")]  
public string AuthorizedApplicationId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

AuthorizedParty

The Authorized party. This is also the `azp` claim.

Declaration

```
[JsonPropertyName("authorized_party")]  
public string AuthorizedParty { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerObjectId

The optional consumer object id.

Declaration

```
[JsonPropertyName("consumer_object_id")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public string ConsumerObjectId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantId

The consumer tenant id.

Declaration

```
[JsonPropertyName("consumer_tenant_id")]
public string ConsumerTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Key

The key to check.

Declaration

```
[JsonPropertyName("key")]
public string Key { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

KeyType

The key type. `primary_key` or `secondary_key`.

Declaration

```
[JsonPropertyName("key_type")]
public string KeyType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ProviderTenantId

The registered provider or the provider tenant id.

Declaration


```
[JsonPropertyName("provider_tenant_id")]
public string ProviderTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Subject

The subject. This is also the `sub` claim.

Declaration

```
[JsonPropertyName("subject")]
public string Subject { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class LicenseClaimsModel

Defines the licensing claims model.

Inheritance

System.Object
LicenseClaimsModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing.Models](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public sealed class LicenseClaimsModel
```

Constructors

LicenseClaimsModel()

Initialize a new instance. This constructor is part of the internal infrastructure. Please do not use it directly in the application code. To create a new instance from claims please use [Create\(IDictionary<String, Object>\)](#).

Declaration

```
public LicenseClaimsModel()
```

Properties

AcrValues

The `acr` claim values.

Declaration

```
[JsonPropertyName("acr_values")]  
public string AcrValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Audience

The `aud` claim.

Declaration

```
[JsonPropertyName("audience")]  
public string Audience { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

AuthorizedApplicationIds

The `auth_apps` claim.

Declaration

```
[JsonPropertyName("authorized_application_ids")]
public string AuthorizedApplicationIds { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

AuthorizedParty

The `azp` claim.

Declaration

```
[JsonPropertyName("authorized_party")]
public string AuthorizedParty { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Custom

The custom claims.

Declaration

```
[JsonPropertyName("custom")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public Dictionary<string, object> Custom { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.Dictionary<System.String, System.Object>	

Expiry

The `exp` claim.

Declaration

```
[JsonPropertyName("expiry")]
public DateTimeOffset? Expiry { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable< DateTimeOffset >	

IssuedAt

The `iat` claim.

Declaration

```
[JsonPropertyName("issued_at")]
public DateTimeOffset? IssuedAt { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable< DateTimeOffset >	

Issuer

The `iss` claim.

Declaration

```
[JsonPropertyName("issuer")]
public string Issuer { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Jti

The `jti` claim.

Declaration

```
[JsonPropertyName("jti")]
public string Jti { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Name

The `name` claim.

Declaration

```
[JsonPropertyName("name")]
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

NotBefore

The `nbf` claim.

Declaration

```
[JsonPropertyName("not_before")]
public DateTimeOffset? NotBefore { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable< DateTimeOffset >	

ObjectCountry

The optional `ctry` claim.

Declaration

```
[JsonPropertyName("object_country")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public string ObjectCountry { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ObjectId

The optional `oid` claim.

Declaration

```
[JsonPropertyName("object_id")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public string ObjectId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Subject

The `sub` claim.

Declaration

```
[JsonPropertyName("subject")]
public string Subject { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

TenantCountry

The `tenant_ctry` claim.

Declaration

```
[JsonPropertyName("tenant_country")]
public string TenantCountry { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

TenantId

The `tid` claim.

Declaration

```
[JsonPropertyName("tenant_id")]
public string TenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

Create(IDictionary<String, Object>)

Creates a new instance of [LicenseClaimsModel](#) based on the specified claims dictionary.

Declaration

```
public static LicenseClaimsModel Create(IDictionary<string, object> claims)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IDictionary<System.String, System.Object>	claims	The source claims.

Returns

TYPE	DESCRIPTION
LicenseClaimsModel	

Class LicenseKeyJsonFileModel

The `jws` B2B keys download model.

Inheritance

System.Object
LicenseKeyJsonFileModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing.Models](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class LicenseKeyJsonFileModel
```

Properties

AuthorizedParty

The Authorized party. This is also the `azp` claim.

Declaration

```
[JsonPropertyName("authorized_party")]  
public string AuthorizedParty { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerObjectId

The optional consumer object id.

Declaration

```
[JsonPropertyName("consumer_object_id")]  
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]  
public string ConsumerObjectId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantId

The consumer tenant id.

Declaration

```
[JsonPropertyName("consumer_tenant_id")]  
public string ConsumerTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ExpiresIn

The expiry in days.

Declaration

```
[JsonPropertyName("expires_in")]  
public int ExpiresIn { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Key

The key to check.

Declaration

```
[JsonPropertyName("key")]  
public string Key { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

KeyType

The key type. `primary_key` or `secondary_key`.

Declaration

```
[JsonPropertyName("key_type")]  
public string KeyType { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ProviderId

The registered provider or the provider tenant id.

Declaration


```
[JsonPropertyName("provider_id")]
public string ProviderId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Subject

The subject. This is also the `sub` claim.

Declaration

```
[JsonPropertyName("subject")]
public string Subject { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class LicenseKeysModel

The `jws` B2B keys model.

Inheritance

System.Object
LicenseKeysModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing.Models](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class LicenseKeysModel
```

Properties

AcrValues

The `acr` claim.

Declaration

```
[JsonPropertyName("acr_values")]  
public string[] AcrValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

AuthorizedApplicationIds

The authorized application. This is also the `auth_apps` claim.

Declaration

```
[JsonPropertyName("authorized_application_ids")]  
public string[] AuthorizedApplicationIds { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

AuthorizedParty

The Authorized party. This is also the `azp` claim.

Declaration

```
[JsonPropertyName("authorized_party")]
public string AuthorizedParty { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantId

The consumer tenant id.

Declaration

```
[JsonPropertyName("consumer_tenant_id")]
public string ConsumerTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Etag

The etag.

Declaration

```
[JsonPropertyName("etag")]
public string Etag { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ExpiresIn

The expires in days.

Declaration

```
[JsonPropertyName("expires_in")]
public int ExpiresIn { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Operations

The operations available for clients.

Declaration

```
[JsonPropertyName("operations")]
public string[] Operations { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

PrimaryKey

The primary `jwt` license key.

Declaration

```
[JsonPropertyName("primary_key")]
public string PrimaryKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ProviderTenantId

The registered provider or provider tenant id.

Declaration

```
[JsonPropertyName("provider_tenant_id")]
public string ProviderTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

PublisherTenantId

The publisher tenant id.

Declaration

```
[JsonPropertyName("publisher_tenant_id")]
public string PublisherTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

SecondaryKey

The secondary `jwt` license key. Use this key when rotating the primary key.

Declaration

```
[JsonPropertyName("secondary_key")]  
public string SecondaryKey { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Subject

The subject. This is also the `sub` claim.

Declaration

```
[JsonPropertyName("subject")]  
public string Subject { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class LicenseProvisioningModel

The license provisioning model.

Inheritance

System.Object
LicenseProvisioningModel

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Licensing.Models](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public class LicenseProvisioningModel
```

Properties

AcrValues

The `acr` claim.

Declaration

```
[JsonPropertyName("acr_values")]  
public string[] AcrValues { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

AuthorizedApplicationIds

The `auth_apps` claim.

Declaration

```
[JsonPropertyName("authorized_application_ids")]  
public string[] AuthorizedApplicationIds { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String[]	

AuthorizedParty

The `azp` claim.

Declaration

```
[JsonPropertyName("authorized_party")]
public string AuthorizedParty { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantCountry

The consumer tenant id. This is the `tid` claim.

Declaration

```
[JsonPropertyName("consumer_tenant_country")]
public string ConsumerTenantCountry { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantId

The consumer tenant id. This is the `tid` claim.

Declaration

```
[JsonPropertyName("consumer_tenant_id")]
public string ConsumerTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ConsumerTenantName

The consumer tenant id. This is the `tid` claim.

Declaration

```
[JsonPropertyName("consumer_tenant_name")]
public string ConsumerTenantName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ExpiresIn

The expiry in days.

Declaration

```
[JsonPropertyName("expires_in")]
public int ExpiresIn { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

Operation

The space separated keys operations. Valid value is `actions`, `get`, `generate`, `reset_primary`, `reset_secondary` and `delete`.

Declaration

```
[JsonPropertyName("operation")]
public string Operation { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ProviderTenantId

The provider tenant id.

Declaration

```
[JsonPropertyName("provider_tenant_id")]
public string ProviderTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

PublisherTenantId

The publisher tenant id.

Declaration

```
[JsonPropertyName("publisher_tenant_id")]
public string PublisherTenantId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Subject

The `sub` claim.

Declaration


```
[JsonPropertyName("subject")]  
public string Subject { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Protocols.Security

Classes

[HexSaltGenerator](#)

Generates a HEX salt using cryptographic `System.Security.Cryptography.RandomNumberGenerator`.

[IdGenerator](#)

Generates unique identifiers.

Class HexSaltGenerator

Generates a HEX salt using cryptographic System.Security.Cryptography.RandomNumberGenerator.

Inheritance

System.Object
HexSaltGenerator

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Security](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public static class HexSaltGenerator
```

Methods

NewSalt()

Generates a new HEX salt of 32 bytes with no dash.

Declaration

```
public static string NewSalt()
```

Returns

TYPE	DESCRIPTION
System.String	The generated salt

NewSalt(Int32)

Generates a new HEX salt of specified length with no dash.

Declaration

```
public static string NewSalt(int length)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	length	The salt byte length.

Returns

TYPE	DESCRIPTION
System.String	The generated salt.

NewSalt(Int32, Boolean)

Generates a new HEX salt of specified length.

Declaration

```
public static string NewSalt(int length, bool strip)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	length	The salt byte length.
System.Boolean	strip	<code>true</code> to strip dash from the generated salt, otherwise <code>false</code> .

Returns

TYPE	DESCRIPTION
System.String	The generated salt.

Class IdGenerator

Generates unique identifiers.

Inheritance

System.Object
IdGenerator

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Security](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public static class IdGenerator
```

Methods

NewGuid()

Generates a new System.Guid.

Declaration

```
public static string NewGuid()
```

Returns

TYPE	DESCRIPTION
System.String	

NewLongId()

Generates a new unique long identifier. A long identifier is a System.Guid expressed as a string with no '-'.

Declaration

```
public static string NewLongId()
```

Returns

TYPE	DESCRIPTION
System.String	

NewShortId()

Generates a unique short identifier. A short identifier is a System.Guid expressed in a Url safe Base64 characters.

Declaration

```
public static string NewShortId()
```

Returns

TYPE	DESCRIPTION
System.String	A unique short identifier

Namespace

PerpetualIntelligence.Protocols.Security.Certificates

Classes

[X509Store](#)

The System.Security.Cryptography.X509Certificates.X509Certificate2 store for OAuth and OpenID Connect protocols.

Class X509Store

The System.Security.Cryptography.X509Certificates.X509Certificate2 store for OAuth and OpenID Connect protocols.

Inheritance

System.Object
X509Store

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Security.Certificates](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
[WriteUnitTest]
public class X509Store
```

Properties

CurrentUser

The X.509 certificate store used by System.Security.Cryptography.X509Certificates.StoreLocation.CurrentUser.

Declaration

```
public static X509Store CurrentUser { get; }
```

Property Value

TYPE	DESCRIPTION
X509Store	

LocalMachine

The X.509 certificate store assigned to System.Security.Cryptography.X509Certificates.StoreLocation.LocalMachine.

Declaration

```
public static X509Store LocalMachine { get; }
```

Property Value

TYPE	DESCRIPTION
X509Store	

Methods

Find(StoreName, X509FindType, Object, Boolean)

Finds the System.Security.Cryptography.X509Certificates.X509Certificate2 based in the specified System.Security.Cryptography.X509Certificates.X509FindType.

Declaration

```
public IEnumerable<X509Certificate2> Find(StoreName storeName, X509FindType findType, object findValue, bool validOnly = true)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Security.Cryptography.X509Certificates.StoreName	storeName	The System.Security.Cryptography.X509Certificates.StoreName.
System.Security.Cryptography.X509Certificates.X509FindType	findType	The System.Security.Cryptography.X509Certificates.X509FindType.
System.Object	findValue	The find value.
System.Boolean	validOnly	The find value validity.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<System.Security.Cryptography.X509Certificates.X509Certificate2>	

See Also

- System.Security.Cryptography.X509Certificates.X509Certificate2
- System.Security.Cryptography.X509Certificates.StoreLocation
- System.Security.Cryptography.X509Certificates.StoreName

Namespace PerpetualIntelligence.Protocols.Security.Secrets

Classes

[ExtractedSecret](#)

Represents a protected string or a protected key extracted from a context.

[Secret](#)

Represents a protected string or a protected key. A secret is equatable over its [Type](#) and [Value](#). It is a security concern to have multiple secrets with same identifying credentials.

[SecretComparer](#)

Compares two string in length-constant time. This comparison method is used so that password hashes and secrets cannot be extracted from on-line systems using a timing attack and then attacked off-line.

[SecretTypes](#)

Defines the secret types.

Class ExtractedSecret

Represents a protected string or a protected key extracted from a context.

Inheritance

System.Object
ExtractedSecret

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Security.Secrets](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public sealed class ExtractedSecret
```

Properties

Id

The secret identifier.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Type

The secret type.

Declaration

```
public string Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Value

The secret value.

Declaration

```
public object Value { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object	

See Also

- [Secret](#)
- [SecretTypes](#)
- [SecretComparer](#)

Class Secret

Represents a protected string or a protected key. A secret is equatable over its [Type](#) and [Value](#). It is a security concern to have multiple secrets with same identifying credentials.

Inheritance

System.Object
Secret

Implements

System.IEquatable<[Secret](#)>

Inherited Members

System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Security.Secrets](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public sealed class Secret : IEquatable<Secret>
```

Remarks

Secrets are subject to timing attack, please use [SecretComparer](#) to compare the secrets explicitly.

Properties

Description

The secret description.

Declaration

```
[JsonPropertyName("description")]  
public string Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Expiration

The secret expiration date and time expressed in coordinated universal time (UTC).

Declaration

```
[JsonPropertyName("expiration")]  
public DateTimeOffset? Expiration { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Nullable< DateTimeOffset >	

Name

The secret name.

Declaration

```
[JsonPropertyName("name")]
public string Name { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Type

The secret type. Defaults to [Sha](#).

Declaration

```
[JsonPropertyName("type")]
public string Type { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

See Also

[SecretTypes](#)

Value

The secret value.

Declaration

```
[JsonPropertyName("value")]
public string Value { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

Equals(Secret)

Declaration

```
public bool Equals(Secret other)
```

Parameters

TYPE	NAME	DESCRIPTION
Secret	other	

Returns

TYPE	DESCRIPTION
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

System.Object.Equals(System.Object)

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

TYPE	DESCRIPTION
System.Int32	

Overrides

System.Object.GetHashCode()

Operators

Equality(Secret, Secret)

Declaration

```
public static bool operator ==(Secret left, Secret right)
```

Parameters

TYPE	NAME	DESCRIPTION
Secret	left	
Secret	right	

Returns

TYPE	DESCRIPTION
System.Boolean	

Inequality(Secret, Secret)

Declaration

```
public static bool operator !=(Secret left, Secret right)
```

Parameters

TYPE	NAME	DESCRIPTION
Secret	left	
Secret	right	

Returns

TYPE	DESCRIPTION
System.Boolean	

Implements

System.IEquatable<T>

See Also

[ExtractedSecret](#)

[SecretTypes](#)

[SecretComparer](#)

Class SecretComparer

Compares two string in length-constant time. This comparison method is used so that password hashes and secrets cannot be extracted from on-line systems using a timing attack and then attacked off-line.

Inheritance

System.Object
SecretComparer

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Security.Secrets](#)

Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public static class SecretComparer
```

Methods

IsEqual(String, String)

Checks two strings in length-constant time.

Declaration

```
[WriteUnitTest]  
public static bool IsEqual(string s1, string s2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	s1	string 1.
System.String	s2	string 2.

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the specified strings are equal; otherwise, <code>false</code> .

Class SecretTypes

Defines the secret types.

Inheritance

System.Object
SecretTypes

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Protocols.Security.Secrets](#)
Assembly: PerpetualIntelligence.Protocols.dll

Syntax

```
public static class SecretTypes
```

Fields

JsonWebKey

The JSON Web Key (JWK). See <https://datatracker.ietf.org/doc/html/rfc7517>.

Declaration

```
public const string JsonWebKey = "urn:oneimlx:security:secret:jwk"
```

Field Value

TYPE	DESCRIPTION
System.String	

JwtBearer

The JWT bearer token. See <https://datatracker.ietf.org/doc/html/rfc7523>.

Declaration

```
[MustDo("For consistency make this urn:oneimlx:", BlockRelease = true)]  
public const string JwtBearer = "urn:ietf:params:oauth:client-assertion-type:jwt-bearer"
```

Field Value

TYPE	DESCRIPTION
System.String	

None

No secret type.

Declaration

```
public const string None = "urn:oneimlx:security:secret:none"
```

Field Value

TYPE	DESCRIPTION
System.String	

Plain

The regular plain text that an application gives to an authorized secret owner.

Declaration

```
public const string Plain = "urn:oneimlx:security:secret:pln"
```

Field Value

TYPE	DESCRIPTION
System.String	

Sha

The SHA hash an application gives to an authorized secret owner.

Declaration

```
public const string Sha = "urn:oneimlx:security:secret:sha"
```

Field Value

TYPE	DESCRIPTION
System.String	

X509Base64

The System.Security.Cryptography.X509Certificates.X509Certificate2 Base64 encoding.

Declaration

```
public const string X509Base64 = "urn:oneimlx:security:secret:x509b64"
```

Field Value

TYPE	DESCRIPTION
System.String	

X509Mtls

The System.Security.Cryptography.X509Certificates.X509Certificate2 Mtls.

Declaration

```
public const string X509Mtls = "urn:oneimlx:security:secret:x509mtls"
```

Field Value

TYPE	DESCRIPTION
System.String	

X509Subject

The System.Security.Cryptography.X509Certificates.X509Certificate2 subject's distinguished name.

Declaration

```
public const string X509Subject = "urn:oneimlx:security:secret:x509sub"
```

Field Value

TYPE	DESCRIPTION
System.String	

X509Thumbprint

The System.Security.Cryptography.X509Certificates.X509Certificate2 thumbprint.

Declaration

```
public const string X509Thumbprint = "urn:oneimlx:security:secret:x509tb"
```

Field Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Shared.Abstractions

Interfaces

[ICommunicationSender](#)

An abstraction to send standard communication via email or text message.

[IComponentIdGenerator](#)

An abstraction to generate a component identifier.

[IEmailSender](#)

An abstraction to send an email.

[IEncoder](#)

An abstraction to encode and decode text.

[IIdAccessor](#)

An abstraction of an id accessor.

[IIdentityAccessor](#)

An abstraction of an identity accessor.

[IIdGenerator](#)

An abstraction of an id generator.

[ILocalizer](#)

An abstraction of the localizer.

[IModeAccessor](#)

An abstraction to access the current mode.

[INameAccessor](#)

An abstraction to return the subject's name.

[IOneImIxResult](#)

An abstraction of a generic `OneImIx` result.

[ITextMessageSender](#)

An abstraction to send a text message.

Interface ICommunicationSender

An abstraction to send standard communication via email or text message.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface ICommunicationSender
```

Properties

EmailSender

The email sender.

Declaration

```
ISEmailSender EmailSender { get; }
```

Property Value

TYPE	DESCRIPTION
ISEmailSender	An instance of ISEmailSender .

TextMessageSender

The text message sender.

Declaration

```
ITextMessageSender TextMessageSender { get; }
```

Property Value

TYPE	DESCRIPTION
ITextMessageSender	An instance of ITextMessageSender .

Interface IComponentIdGenerator

An abstraction to generate a component identifier.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface IComponentIdGenerator
```

Methods

NewId()

Generates a new component id.

Declaration

```
string NewId()
```

Returns

TYPE	DESCRIPTION
System.String	A new component id.

Interface IEmailSender

An abstraction to send an email.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: [PerpetualIntelligence.Shared.dll](#)

Syntax

```
public interface IEmailSender
```

Methods

SendAsync(String, String[], String, String, String[])

Sends an email asynchronously.

Declaration

```
Task SendAsync(string from, string[] to, string subject, string message, string[] bcc)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	from	The sender.
System.String[]	to	The recipients.
System.String	subject	The subject.
System.String	message	The message.
System.String[]	bcc	The bcc recipients.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	System.Threading.Tasks.Task instance that represents an asynchronous send operation.

SendAsync(String[], String, String, String[])

Sends an email from a configured sender asynchronously.

Declaration

```
Task SendAsync(string[] to, string subject, string message, string[] bcc)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String[]	to	The recipients.
System.String	subject	The subject.
System.String	message	The message.
System.String[]	bcc	The bcc recipients.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	System.Threading.Tasks.Task instance that represents an asynchronous send operation.

Interface IEncoder

An abstraction to encode and decode text.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface IEncoder
```

Methods

Decode(String)

Decodes the text.

Declaration

```
string Decode(string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	Text to decode.

Returns

TYPE	DESCRIPTION
System.String	The decoded text.

Encode(Byte[])

Encodes the bytes.

Declaration

```
string Encode(byte[] bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
Byte[]	bytes	Bytes to encode.

Returns

TYPE	DESCRIPTION
System.String	The encoded text.

Encode(String)

Encodes the text.

Declaration

```
string Encode(string text)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	text	Text to encode.

Returns

TYPE	DESCRIPTION
System.String	The encoded text.

Interface IIdAccessor

An abstraction of an id accessor.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface IIdAccessor
```

Methods

GetId()

Gets the identifier.

Declaration

```
string GetId()
```

Returns

TYPE	DESCRIPTION
System.String	An identifier.

See Also

[IIdGenerator](#)

[IIdentityAccessor](#)

Interface IIdentityAccessor

An abstraction of an identity accessor.

Inherited Members

[IIdAccessor.GetId\(\)](#)

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface IIdentityAccessor : IIdAccessor
```

Remarks

The identity accessor returns see [Identity](#) of a target entity. It also implements [IIdAccessor](#) to return string representation of an [Identity](#).

Methods

[GetIdentity\(\)](#)

Gets the identity.

Declaration

```
Identity GetIdentity()
```

Returns

TYPE	DESCRIPTION
Identity	An identity instance.

Interface IIdGenerator

An abstraction of an id generator.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: [PerpetualIntelligence.Shared.dll](#)

Syntax

```
public interface IIdGenerator
```

Remarks

The [IIdGenerator](#) can generate regular text id, numeric id, partitioned id and mode id.

Methods

DecodeModelId(String)

Decodes the specified mode identifier.

Declaration

```
ModeIdentity DecodeModeId(string modeId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	modeId	The mode identifier to decode.

Returns

TYPE	DESCRIPTION
ModeIdentity	An identity instance that represents the decoded mode identifier.

DecodePartitionedId(String, Boolean)

Decodes the specified partitioned identifier.

Declaration

```
Identity DecodePartitionedId(string partitionedId, bool splitId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	partitionedId	The partitioned identifier to decode.
System.Boolean	splitId	<code>true</code> to split the partitionedId identifier, otherwise <code>false</code> .

Returns

TYPE	DESCRIPTION
Identity	An identity instance that represents a decoded partitioned identifier.

NewLongId()

Generates a new long identifier. Typically this a long version of a System.Guid.

Declaration

```
string NewLongId()
```

Returns

TYPE	DESCRIPTION
System.String	A new long identifier.

NewModelId(String, String)

Generates a new mode identifier. A mode identifier is a compound identifier of the specified mode, partition identifier and a new unique identifier.

Declaration

```
string NewModeId(string mode, string partitionId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	mode	The mode. See Modes .
System.String	partitionId	The partition identifier.

Returns

TYPE	DESCRIPTION
System.String	A new mode identifier.

NewNumericId()

Generates a new numeric identifier.

Declaration

```
int NewNumericId()
```

Returns

TYPE	DESCRIPTION
System.Int32	A new numeric identifier.

NewPartitionedId(String)

Generates a new partitioned identifier. A partitioned identifier is a compound identifier of the specified partition identifier and a new unique identifier.

Declaration

```
string NewPartitionedId(string partitionId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	partitionId	The partition identifier.

Returns

TYPE	DESCRIPTION
System.String	A new partitioned identifier.

NewShortId()

Generates a new short identifier. Typically this a short version of a System.Guid while maintaining the uniqueness.

Declaration

```
string NewShortId()
```

Returns

TYPE	DESCRIPTION
System.String	A new short identifier.

See Also

[Identity](#)

[ModelIdentity](#)

[Modes](#)

Interface ILocalizer

An abstraction of the localizer.

Inherited Members

Microsoft.Extensions.Localization.IStringLocalizer.GetAllStrings(System.Boolean)
Microsoft.Extensions.Localization.IStringLocalizer.Item[System.String]
Microsoft.Extensions.Localization.IStringLocalizer.Item[System.String, System.Object[]]

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface ILocalizer : IStringLocalizer
```

Methods

FindAsync(String)

Finds a locale specific string for the specified translation unit id.

Declaration

```
Task<LocalizedString> FindAsync(string tuId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	tuId	The translation unit identifier.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<Microsoft.Extensions.Localization.LocalizedString>	An instance of Microsoft.Extensions.Localization.LocalizedString.

FindAsync(String, Object[])

Finds a formatted locale specific string for the specified translation unit id.

Declaration

```
Task<LocalizedString> FindAsync(string tuId, params object[] arguments)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	tuId	The translation unit identifier.
System.Object[]	arguments	The format arguments.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<Microsoft.Extensions.Localization.LocalizedString>	An instance of Microsoft.Extensions.Localization.LocalizedString.

TranslateAsync(String)

Translates the specified string input and returns a locale specific string.

Declaration

Task<LocalizedString> TranslateAsync(string input)
--

Parameters

TYPE	NAME	DESCRIPTION
System.String	input	String to translate.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<Microsoft.Extensions.Localization.LocalizedString>	An instance of Microsoft.Extensions.Localization.LocalizedString.

Interface IModeAccessor

An abstraction to access the current mode.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface IModeAccessor
```

Methods

GetMode()

Returns the current mode.

Declaration

```
string GetMode()
```

Returns

TYPE	DESCRIPTION
System.String	The current mode.

See Also

[All\(\)](#)

See Also

[Modes](#)

Interface INameAccessor

An abstraction to return the subject's name.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public interface INameAccessor
```

Methods

GetName()

Returns the subject's name.

Declaration

```
string GetName()
```

Returns

TYPE	DESCRIPTION
System.String	The name.

Interface IOneImlxResult

An abstraction of a generic `OneImlx` result.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: `PerpetualIntelligence.Shared.dll`

Syntax

```
public interface IOneImlxResult
```

Properties

IsError

Determines if the result is an error.

Declaration

```
bool IsError { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	A boolean value.

Interface ITextMessageSender

An abstraction to send a text message.

Namespace: [PerpetualIntelligence.Shared.Abstractions](#)

Assembly: [PerpetualIntelligence.Shared.dll](#)

Syntax

```
public interface ITextMessageSender
```

Methods

SendAsync(String, String)

Sends a text message asynchronously

Declaration

```
Task SendAsync(string message, string phoneNumber)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	The text message.
System.String	phoneNumber	The phone number.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	System.Threading.Tasks.Task instance that represents a send operation.

Namespace PerpetualIntelligence.Shared.Attributes

Classes

ActionAttribute

An actionable attribute.

ArchitectureAttribute

Specifies that the target has an architectural comment that may be resolved in future releases. Applications can use the target element, but the design may change in future releases including breaking changes.

ConceptAttribute

Indicates that the target element represents a concept.

FutureAttribute

Indicates a future capability or a feature request.

InternalInfrastructureAttribute

Indicates that the target is part of the internal infrastructure. Application should not use the target directly.

MustDoAttribute

Indicates a MUST-DO action on the target element.

PerformanceAttribute

Specifies that the target has a performance concern that may be resolved in future releases. Applications can use the target element, but the design may change in future releases including breaking changes.

RedundancyAttribute

Specifies that the target has a redundancy that can be resolved in future releases. Applications can use the target element, but the design may change in future releases including breaking changes.

RefactorAttribute

Specifies that the target should be re-factored. Applications can use the target element, but the design may change in future releases including breaking changes.

RenameAttribute

Specifies that the target should be renamed.

TodoAttribute

Indicates a TODO action on the target element.

WriteApiAttribute

Indicates that the target element should add the relevant back-end API.

WriteDocumentationAttribute

Indicates that the target element requires documentation.

WriteIntegrationTestAttribute

Indicates that the target element should have integration tests.

WriteUnitTestAttribute

Indicates that the target element should have unit tests.

Class ActionAttribute

An actionable attribute.

Inheritance

System.Object

System.Attribute

ActionAttribute

[ArchitectureAttribute](#)

[ConceptAttribute](#)

[FutureAttribute](#)

[MustDoAttribute](#)

[PerformanceAttribute](#)

[RedundancyAttribute](#)

[RefactorAttribute](#)

[RenameAttribute](#)

[TodoAttribute](#)

[WriteApiAttribute](#)

[WriteDocumentationAttribute](#)

[WriteIntegrationTestAttribute](#)

[WriteUnitTestAttribute](#)

Inherited Members

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public abstract class ActionAttribute : Attribute
```

Constructors

ActionAttribute()

Initialize a new instance.

Declaration

```
protected ActionAttribute()
```

ActionAttribute(String)

Initialize a new instance.

Declaration

```
protected ActionAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

Assignee

The action assignee.

Declaration

```
public string Assignee { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Description

The description.

Declaration

```
public string Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Severity

The severity.

Declaration

```
public string Severity { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Version

The version to address an action.

Declaration

```
public string Version { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class ArchitectureAttribute

Specifies that the target has an architectural comment that may be resolved in future releases. Applications can use the target element, but the design may change in future releases including breaking changes.

Inheritance

System.Object

System.Attribute

[ActionAttribute](#)

ArchitectureAttribute

Inherited Members

[ActionAttribute.Assignee](#)

[ActionAttribute.Description](#)

[ActionAttribute.Severity](#)

[ActionAttribute.Version](#)

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, Inherited = false, AllowMultiple = true)]  
public sealed class ArchitectureAttribute : ActionAttribute
```

Constructors

ArchitectureAttribute(String)

Initialize a new instance.

Declaration

```
public ArchitectureAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	The description.

Class ConceptAttribute

Indicates that the target element represents a concept.

Inheritance

System.Object
System.Attribute
[ActionAttribute](#)
ConceptAttribute

Inherited Members

[ActionAttribute.Assignee](#)
[ActionAttribute.Description](#)
[ActionAttribute.Severity](#)
[ActionAttribute.Version](#)
System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class ConceptAttribute : ActionAttribute
```

Constructors

ConceptAttribute(String)

Initialize a new instance.

Declaration

```
public ConceptAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Class FutureAttribute

Indicates a future capability or a feature request.

Inheritance

System.Object
System.Attribute
[ActionAttribute](#)
FutureAttribute

Inherited Members

[ActionAttribute.Assignee](#)
[ActionAttribute.Description](#)
[ActionAttribute.Severity](#)
[ActionAttribute.Version](#)
System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class FutureAttribute : ActionAttribute
```

Constructors

FutureAttribute(String)

Initialize a new instance.

Declaration

```
public FutureAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Class InternalInfrastructureAttribute

Indicates that the target is part of the internal infrastructure. Application should not use the target directly.

Inheritance

System.Object

System.Attribute

InternalInfrastructureAttribute

Inherited Members

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Syntax

```
[AttributeUsage(AttributeTargets.All, Inherited = false, AllowMultiple = true)]
public sealed class InternalInfrastructureAttribute : Attribute
```

Properties

Description

Additional description.

Declaration

```
public string Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class MustDoAttribute

Indicates a MUST-DO action on the target element.

Inheritance

System.Object
System.Attribute
[ActionAttribute](#)
MustDoAttribute

Inherited Members

[ActionAttribute.Assignee](#)
[ActionAttribute.Description](#)
[ActionAttribute.Severity](#)
[ActionAttribute.Version](#)
System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class MustDoAttribute : ActionAttribute
```

Constructors

MustDoAttribute(String)

Initialize a new instance.

Declaration

```
public MustDoAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Properties

BlockRelease

This indicates that DevOps should block the package release until the MUST-DO action is open.

Declaration

```
public bool BlockRelease { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Remarks

You should write tests that execute only in your DevOps release environment and check for this attribute.

Class PerformanceAttribute

Specifies that the target has a performance concern that may be resolved in future releases. Applications can use the target element, but the design may change in future releases including breaking changes.

Inheritance

System.Object

System.Attribute

[ActionAttribute](#)

PerformanceAttribute

Inherited Members

[ActionAttribute.Assignee](#)

[ActionAttribute.Description](#)

[ActionAttribute.Severity](#)

[ActionAttribute.Version](#)

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, Inherited = false, AllowMultiple = true)]  
public sealed class PerformanceAttribute : ActionAttribute
```

Constructors

PerformanceAttribute(String)

Initialize a new instance.

Declaration

```
public PerformanceAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	

Class RedundancyAttribute

Specifies that the target has a redundancy that can be resolved in future releases. Applications can use the target element, but the design may change in future releases including breaking changes.

Inheritance

System.Object

System.Attribute

[ActionAttribute](#)

RedundancyAttribute

Inherited Members

[ActionAttribute.Assignee](#)

[ActionAttribute.Description](#)

[ActionAttribute.Severity](#)

[ActionAttribute.Version](#)

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, Inherited = false, AllowMultiple = true)]  
public sealed class RedundancyAttribute : ActionAttribute
```

Constructors

RedundancyAttribute(String)

Initialize a new instance.

Declaration

```
public RedundancyAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	

Class RefactorAttribute

Specifies that the target should be re-factored. Applications can use the target element, but the design may change in future releases including breaking changes.

Inheritance

System.Object
System.Attribute
[ActionAttribute](#)
RefactorAttribute

Inherited Members

[ActionAttribute.Assignee](#)
[ActionAttribute.Description](#)
[ActionAttribute.Severity](#)
[ActionAttribute.Version](#)
System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, Inherited = false, AllowMultiple = true)]  
public sealed class RefactorAttribute : ActionAttribute
```

Constructors

RefactorAttribute(String)

Initialize a new instance.

Declaration

```
public RefactorAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	

Class RenameAttribute

Specifies that the target should be renamed.

Inheritance

System.Object
System.Attribute
[ActionAttribute](#)
RenameAttribute

Inherited Members

[ActionAttribute.Assignee](#)
[ActionAttribute.Description](#)
[ActionAttribute.Severity](#)
[ActionAttribute.Version](#)
System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, Inherited = false, AllowMultiple = true)]  
public sealed class RenameAttribute : ActionAttribute
```

Constructors

RenameAttribute(String)

Initialize a new instance.

Declaration

```
public RenameAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	

Class TodoAttribute

Indicates a TODO action on the target element.

Inheritance

System.Object
System.Attribute
[ActionAttribute](#)
TodoAttribute

Inherited Members

[ActionAttribute.Assignee](#)
[ActionAttribute.Description](#)
[ActionAttribute.Severity](#)
[ActionAttribute.Version](#)
System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class TodoAttribute : ActionAttribute
```

Constructors

TodoAttribute(String)

Initialize a new instance.

Declaration

```
public TodoAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Class WriteApiAttribute

Indicates that the target element should add the relevant back-end API.

Inheritance

System.Object

System.Attribute

[ActionAttribute](#)

WriteApiAttribute

Inherited Members

[ActionAttribute.Assignee](#)

[ActionAttribute.Description](#)

[ActionAttribute.Severity](#)

[ActionAttribute.Version](#)

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class WriteApiAttribute : ActionAttribute
```

Constructors

WriteApiAttribute(String)

Initialize a new instance.

Declaration

```
public WriteApiAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Class WriteDocumentationAttribute

Indicates that the target element requires documentation.

Inheritance

System.Object

System.Attribute

[ActionAttribute](#)

WriteDocumentationAttribute

Inherited Members

[ActionAttribute.Assignee](#)

[ActionAttribute.Description](#)

[ActionAttribute.Severity](#)

[ActionAttribute.Version](#)

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class WriteDocumentationAttribute : ActionAttribute
```

Constructors

WriteDocumentationAttribute()

Initializes a new instance.

Declaration

```
public WriteDocumentationAttribute()
```

WriteDocumentationAttribute(String)

Initialize a new instance.

Declaration

```
public WriteDocumentationAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Class WriteIntegrationTestAttribute

Indicates that the target element should have integration tests.

Inheritance

System.Object

System.Attribute

[ActionAttribute](#)

WriteIntegrationTestAttribute

Inherited Members

[ActionAttribute.Assignee](#)

[ActionAttribute.Description](#)

[ActionAttribute.Severity](#)

[ActionAttribute.Version](#)

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class WriteIntegrationTestAttribute : ActionAttribute
```

Constructors

WriteIntegrationTestAttribute()

Initialize a new instance.

Declaration

```
public WriteIntegrationTestAttribute()
```

WriteIntegrationTestAttribute(String)

Initialize a new instance with description.

Declaration

```
public WriteIntegrationTestAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Class WriteUnitTestAttribute

Indicates that the target element should have unit tests.

Inheritance

System.Object

System.Attribute

[ActionAttribute](#)

WriteUnitTestAttribute

Inherited Members

[ActionAttribute.Assignee](#)

[ActionAttribute.Description](#)

[ActionAttribute.Severity](#)

[ActionAttribute.Version](#)

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)

System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.Match(System.Object)

System.Attribute.TypeId

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.All, AllowMultiple = true, Inherited = false)]  
public sealed class WriteUnitTestAttribute : ActionAttribute
```

Constructors

WriteUnitTestAttribute()

Initialize a new instance.

Declaration

```
public WriteUnitTestAttribute()
```

WriteUnitTestAttribute(String)

Initialize a new instance with description.

Declaration

```
public WriteUnitTestAttribute(string description)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	description	Additional description.

Namespace PerpetualIntelligence.Shared.Attributes.Api

Classes

[EndpointAttribute](#)

Defines an api endpoint.

[HiddenPropertyAttribute](#)

Hides the target property from the API spec.

[ServiceAttribute](#)

Defines an endpoint service.

Class EndpointAttribute

Defines an api endpoint.

Inheritance

System.Object
System.Attribute
EndpointAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Api](#)

Syntax

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false, Inherited = false)]
public class EndpointAttribute : Attribute
```

Constructors

EndpointAttribute(String, String)

Initialize a new instance.

Declaration

```
public EndpointAttribute(string id, string authority)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The endpoint identifier.
System.String	authority	The endpoint authority.

Properties

Authority

The endpoint authority.

Declaration

```
public string Authority { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Id

The endpoint identifier.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Class HiddenPropertyAttribute

Hides the target property from the API spec.

Inheritance

System.Object
System.Attribute
HiddenPropertyAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Api](#)

Syntax

```
[AttributeUsage(AttributeTargets.Property, AllowMultiple = false, Inherited = false)]  
public class HiddenPropertyAttribute : Attribute
```

Remarks

The property marked as `System.Text.Json.Serialization.JsonIgnoreAttribute` may be visible in the API spec generator such as Swagger UI. Use this attribute to process and remove the property from your API spec schema.

Constructors

`HiddenPropertyAttribute(String, String)`

Initializes a new instance.

Declaration

```
public HiddenPropertyAttribute(string property, string justification)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	property	The property to hide.
System.String	justification	The justification to hide the property.

Properties

`Justification`

The justification to hide the property.

Declaration

```
public string Justification { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Property

The property to hide.

Declaration

```
public string Property { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class ServiceAttribute

Defines an endpoint service.

Inheritance

System.Object
System.Attribute
ServiceAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Api](#)

Syntax

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false, Inherited = false)]  
public class ServiceAttribute : Attribute
```

Constructors

ServiceAttribute(String)

Initialize a new instance.

Declaration

```
public ServiceAttribute(string serviceId)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	serviceId	

Properties

ServiceId

Service identifier.

Declaration

```
public string ServiceId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Shared.Attributes.Rendering

Classes

[GroupOrderAttribute](#)

The relative group order for System.ComponentModel.DataAnnotations.DisplayAttribute.GroupName defined in a class. The group order and System.ComponentModel.DataAnnotations.DisplayAttribute.Order are not the same. The System.ComponentModel.DataAnnotations.DisplayAttribute.Order defines order of elements inside the group.

[OverrideDisplayAttribute](#)

Instructs the UX rendering to ignore the System.ComponentModel.DataAnnotations.DisplayAttribute and use the overridden values specified by [OverrideDisplayAttribute](#).

[SubObjectPropertyAttribute](#)

Instructs the UX rendering to apply the [DataType](#) and [Display](#) to the sub-object [PropertyName](#).

Class GroupOrderAttribute

The relative group order for System.ComponentModel.DataAnnotations.DisplayAttribute.GroupName defined in a class. The group order and System.ComponentModel.DataAnnotations.DisplayAttribute.Order are not the same. The System.ComponentModel.DataAnnotations.DisplayAttribute.Order defines order of elements inside the group.

Inheritance

System.Object
System.Attribute
GroupOrderAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Rendering](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = true, Inherited = false)]
public sealed class GroupOrderAttribute : Attribute
```

Constructors

GroupOrderAttribute(String, Int32)

Initializes a new instance.

Declaration

```
public GroupOrderAttribute(string groupName, int groupOrder)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	groupName	The group name.
System.Int32	groupOrder	The group order.

Properties

GroupName

The group name.

Declaration

```
public string GroupName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

GroupOrder

The group order.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.Int32	

Class OverrideDisplayAttribute

Instructs the UX rendering to ignore the `System.ComponentModel.DataAnnotations.DisplayAttribute` and use the overridden values specified by [OverrideDisplayAttribute](#).

Inheritance

System.Object
System.Attribute
OverrideDisplayAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Rendering](#)

Assembly: [PerpetualIntelligence.Shared.dll](#)

Syntax

```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Property, AllowMultiple = true, Inherited = false)]
public sealed class OverrideDisplayAttribute : Attribute
```

Remarks

You cannot override the data type. Use this attribute to re-position your property of a base entity while rendering it in the context of a derived entity.

Constructors

OverrideDisplayAttribute(String, String, String, String, Int32, String, String, Boolean, Boolean, Type)

Initialize a new instance.

Declaration

```
public OverrideDisplayAttribute(string propertyName, string name, string description, string groupName, int
order, string prompt = null, string shortName = null, bool autoGenerateField = true, bool autoGenerateFilter =
false, Type resourceType = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	propertyName	The display property to override.
System.String	name	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Name.
System.String	description	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Description.
System.String	groupName	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.GroupName.
System.Int32	order	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Order.
System.String	prompt	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Prompt.
System.String	shortName	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.ShortName.
System.Boolean	autoGenerateField	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.AutoGenerateField.
System.Boolean	autoGenerateFilter	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.AutoGenerateFilter.

TYPE	NAME	DESCRIPTION
System.Type	resourceType	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.ResourceType.

Properties

Override

The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.

Declaration

```
public DisplayAttribute Override { get; }
```

Property Value

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.DisplayAttribute	

PropertyName

The display property to override.

Declaration

```
public string PropertyName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class SubObjectPropertyAttribute

Instructs the UX rendering to apply the [DataType](#) and [Display](#) to the sub-object [PropertyName](#).

Inheritance

System.Object
System.Attribute
SubObjectPropertyAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Rendering](#)

Assembly: [PerpetualIntelligence.Shared.dll](#)

Syntax

```
[AttributeUsage(AttributeTargets.Class | AttributeTargets.Property, AllowMultiple = true, Inherited = false)]  
public sealed class SubObjectPropertyAttribute : Attribute
```

Remarks

Use this attribute to re-position your property of a sub-object while rendering it in the context of a parent entity.

Constructors

SubObjectPropertyAttribute(String, DataType, String, String, String, Int32, String, String, Boolean, Boolean, Type)

Initialize a new instance.

Declaration

```
public SubObjectPropertyAttribute(string propertyName, DataType dataType, string name, string description, string groupName,
int order, string prompt = null, string shortName = null, bool autoGenerateField = true, bool autoGenerateFilter = false,
Type resourceType = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	propertyName	The sub-object property to override.
System.ComponentModel.DataAnnotations.DataType	dataType	The overridden System.ComponentModel.DataAnnotations.DataTypeAttribute.DataType
System.String	name	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Name.
System.String	description	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Description.
System.String	groupName	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.GroupName.
System.Int32	order	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Order.
System.String	prompt	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Prompt.
System.String	shortName	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.ShortName.
System.Boolean	autoGenerateField	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.AutoGenerateField.
System.Boolean	autoGenerateFilter	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.AutoGenerateFilter.
System.Type	resourceType	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.ResourceType.

SubObjectPropertyAttribute(String, String, String, String, String, Int32, String, String, Boolean, Boolean, Type)

Initialize a new instance.

Declaration

```
public SubObjectPropertyAttribute(string propertyName, string customDataType, string name, string description, string
groupName, int order, string prompt = null, string shortName = null, bool autoGenerateField = true, bool autoGenerateFilter =
false, Type resourceType = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	propertyName	The sub-object property to override.
System.String	customDataType	The overridden System.ComponentModel.DataAnnotations.DataTypeAttribute.CustomDataType
System.String	name	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Name.
System.String	description	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Description.
System.String	groupName	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.GroupName.
System.Int32	order	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Order.
System.String	prompt	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.Prompt.
System.String	shortName	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.ShortName.
System.Boolean	autoGenerateField	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.AutoGenerateField.
System.Boolean	autoGenerateFilter	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.AutoGenerateFilter.
System.Type	resourceType	The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.ResourceType.

Properties

DataType

The overridden System.ComponentModel.DataAnnotations.DataTypeAttribute.

Declaration

```
public DataTypeAttribute DataType { get; }
```

Property Value

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.DataTypeAttribute	

Display

The overridden System.ComponentModel.DataAnnotations.DisplayAttribute.

Declaration

```
public DisplayAttribute Display { get; }
```

Property Value

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.DisplayAttribute	

PropertyName

The sub-object property to override.

Declaration

```
public string PropertyName { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Shared.Attributes.Validation

Classes

[EmailListAttribute](#)

Specifies that a data field value is a list of well formed emails.

[OneOfAttribute](#)

Specifies the allowed values for a property or a field.

[SelectItemAttribute](#)

Specifies the allowed string values on the property.

[StringLengthListAttribute](#)

Specifies the minimum and maximum length of characters that are allowed in each data field of a list.

[UrlListAttribute](#)

Specifies that a data field value is a list of well formed URL.

Class EmailListAttribute

Specifies that a data field value is a list of well formed emails.

Inheritance

System.Object

System.Attribute

System.ComponentModel.DataAnnotations.ValidationAttribute

EmailListAttribute

Inherited Members

System.ComponentModel.DataAnnotations.ValidationAttribute.FormatErrorMessage(System.String)

System.ComponentModel.DataAnnotations.ValidationAttribute.GetValidationResult(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object, System.String)

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessage

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceName

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceType

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageString

System.ComponentModel.DataAnnotations.ValidationAttribute.RequiresValidationContext

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Validation](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.Property | AttributeTargets.Field | AttributeTargets.Parameter, AllowMultiple = false)]  
public class EmailListAttribute : ValidationAttribute
```

Methods

IsValid(Object)

Declaration

```
public override bool IsValid(object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

System.ComponentModel.DataAnnotations.ValidationAttribute.IsValid(System.Object)

IsValid(Object, ValidationContext)

Validates the specified value with respect to the current validation attribute.

Declaration

```
protected override ValidationResult IsValid(object value, ValidationContext validationContext)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	The value to validate.

TYPE	NAME	DESCRIPTION
System.ComponentModel.DataAnnotations.ValidationContext	validationContext	The context information about the validation operation.

Returns

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.ValidationResult	An instance of the System.ComponentModel.DataAnnotations.ValidationResult.

Overrides

System.ComponentModel.DataAnnotations.ValidationAttribute.IsValid(System.Object, System.ComponentModel.DataAnnotations.ValidationContext)

Class OneOfAttribute

Specifies the allowed values for a property or a field.

Inheritance

System.Object

System.Attribute

System.ComponentModel.DataAnnotations.ValidationAttribute

OneOfAttribute

Inherited Members

System.ComponentModel.DataAnnotations.ValidationAttribute.FormatErrorMessage(System.String)

System.ComponentModel.DataAnnotations.ValidationAttribute.GetValidationResult(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.IsValid(System.Object)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object, System.String)

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessage

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceName

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceType

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageString

System.ComponentModel.DataAnnotations.ValidationAttribute.RequiresValidationContext

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Validation](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.Property | AttributeTargets.Field, AllowMultiple = false, Inherited = false)]  
[WriteUnitTest]  
public sealed class OneOfAttribute : ValidationAttribute
```

Constructors

OneOfAttribute(Object[])

Initializes a new instance with the specified allowed values.

Declaration

```
public OneOfAttribute(params object[] allowedValues)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object[]	allowedValues	Allowed values

Properties

AllowedValues

The allowed values.

Declaration

```
public object[] AllowedValues { get; }
```

Property Value

TYPE	DESCRIPTION
System.Object[]	

Methods

IsValid(Object, ValidationContext)

Validates the specified value with respect to the current validation attribute.

Declaration

protected override ValidationResult IsValid(object value, ValidationContext validationContext)

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	The value to validate.
System.ComponentModel.DataAnnotations.ValidationContext	validationContext	The context information about the validation operation.

Returns

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.ValidationResult	An instance of the System.ComponentModel.DataAnnotations.ValidationResult.

Overrides

System.ComponentModel.DataAnnotations.ValidationAttribute.IsValid(System.Object, System.ComponentModel.DataAnnotations.ValidationContext)

Class SelectListItemAttribute

Specifies the allowed string values on the property.

Inheritance

System.Object
System.Attribute
SelectItemAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Validation](#)

Syntax

```
[AttributeUsage(AttributeTargets.Property, AllowMultiple = true, Inherited = false)]
[Refactor("Move this to rendering.")]
public class SelectListItemAttribute : Attribute
```

Constructors

SelectItemAttribute(String, String)

Declaration

```
public SelectListItemAttribute(string name, string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	name	
System.String	value	

Properties

Description

Item display description.

Declaration

```
public string Description { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Disabled

Item disability.

Declaration

```
public bool Disabled { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Name

Item display name.

Declaration

```
public string Name { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Obsolete

The item value is obsolete.

Declaration

```
public bool Obsolete { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Value

Item value.

Declaration

```
public string Value { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Class StringLengthListAttribute

Specifies the minimum and maximum length of characters that are allowed in each data field of a list.

Inheritance

System.Object

System.Attribute

System.ComponentModel.DataAnnotations.ValidationAttribute

System.ComponentModel.DataAnnotations.StringLengthAttribute

StringLengthListAttribute

Inherited Members

System.ComponentModel.DataAnnotations.StringLengthAttribute.FormatErrorMessage(System.String)

System.ComponentModel.DataAnnotations.StringLengthAttribute.MaximumLength

System.ComponentModel.DataAnnotations.StringLengthAttribute.MinimumLength

System.ComponentModel.DataAnnotations.ValidationAttribute.GetValidationResult(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object, System.String)

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessage

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceName

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceType

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageString

System.ComponentModel.DataAnnotations.ValidationAttribute.RequiresValidationContext

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Validation](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class StringLengthListAttribute : StringLengthAttribute
```

Constructors

StringLengthListAttribute(Int32)

Initialize a new instance.

Declaration

```
public StringLengthListAttribute(int maximumLength)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	maximumLength	

Methods

IsValid(Object)

Declaration

```
public override bool IsValid(object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

System.ComponentModel.DataAnnotations.StringLengthAttribute.IsValid(System.Object)

IsValid(Object, ValidationContext)

Validates the specified value with respect to the current validation attribute.

Declaration

```
protected override ValidationResult IsValid(object value, ValidationContext validationContext)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	The value to validate.
System.ComponentModel.DataAnnotations.ValidationContext	validationContext	The context information about the validation operation.

Returns

TYPE	DESCRIPTION
System.ComponentModel.DataAnnotations.ValidationResult	An instance of the System.ComponentModel.DataAnnotations.ValidationResult.

Overrides

System.ComponentModel.DataAnnotations.ValidationAttribute.IsValid(System.Object, System.ComponentModel.DataAnnotations.ValidationContext)

Class UrlListAttribute

Specifies that a data field value is a list of well formed URL.

Inheritance

System.Object

System.Attribute

System.ComponentModel.DataAnnotations.ValidationAttribute

UrlListAttribute

Inherited Members

System.ComponentModel.DataAnnotations.ValidationAttribute.FormatErrorMessage(System.String)

System.ComponentModel.DataAnnotations.ValidationAttribute.GetValidationResult(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.IsValid(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object,

System.ComponentModel.DataAnnotations.ValidationContext)

System.ComponentModel.DataAnnotations.ValidationAttribute.Validate(System.Object, System.String)

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessage

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceName

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageResourceType

System.ComponentModel.DataAnnotations.ValidationAttribute.ErrorMessageString

System.ComponentModel.DataAnnotations.ValidationAttribute.RequiresValidationContext

System.Attribute.Equals(System.Object)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)

System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)

System.Attribute.GetHashCode()

System.Attribute.IsDefaultAttribute()

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)

System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)

System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Attributes.Validation](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[AttributeUsage(AttributeTargets.Property | AttributeTargets.Field | AttributeTargets.Parameter, AllowMultiple = false)]  
public class UrllistAttribute : ValidationAttribute
```

Methods

IsValid(Object)

Declaration

```
public override bool IsValid(object value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	value	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

System.ComponentModel.DataAnnotations.ValidationAttribute.IsValid(System.Object)

Namespace PerpetualIntelligence.Shared.Exceptions

Classes

[ErrorException](#)

The exception that represents an error.

[MultiErrorException](#)

The exception that represents multiple errors.

Class `ErrorException`

The exception that represents an error.

Inheritance

`System.Object`
`System.Exception`
`ErrorException`

Implements

`System.Runtime.Serialization.ISerializable`

Inherited Members

`System.Exception.GetBaseException()`
`System.Exception.GetObjectData(System.Runtime.Serialization.SerializationInfo, System.Runtime.Serialization.StreamingContext)`
`System.Exception.GetType()`
`System.Exception.ToString()`
`System.Exception.Data`
`System.Exception.HelpLink`
`System.Exception.HResult`
`System.Exception.InnerException`
`System.Exception.Source`
`System.Exception.StackTrace`
`System.Exception.TargetSite`
`System.Exception.SerializeObjectState`
`System.Object.Equals(System.Object)`
`System.Object.Equals(System.Object, System.Object)`
`System.Object.GetHashCode()`
`System.Object.MemberwiseClone()`
`System.Object.ReferenceEquals(System.Object, System.Object)`

Namespace: [PerpetualIntelligence.Shared.Exceptions](#)

Assembly: `PerpetualIntelligence.Shared.dll`

Syntax

```
public class ErrorException : Exception, ISerializable
```

Constructors

`ErrorException()`

Initialize a new instance.

Declaration

```
public ErrorException()
```

`ErrorException(Error)`

Initialize a new instance.

Declaration

```
public ErrorException(Error error)
```

Parameters

TYPE	NAME	DESCRIPTION
Error	error	The error.

ErrorException(String)

Initialize a new instance.

Declaration

```
public ErrorException(string message)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	message	The exception message.

ErrorException(String, String)

Initialize a new instance.

Declaration

```
public ErrorException(string error, string errorDescription)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	error	The error code.
System.String	errorDescription	The error description.

ErrorException(String, String, Object[])

Initializes a new error exception.

Declaration

```
public ErrorException(string error, string errorDescription, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	error	The error code.
System.String	errorDescription	The error description.

TYPE	NAME	DESCRIPTION
System.Object[]	args	The error description format arguments.

Properties

Error

The error.

Declaration

```
public Error Error { get; set; }
```

Property Value

TYPE	DESCRIPTION
Error	

Message

The exception message.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Overrides

System.Exception.Message

Implements

System.Runtime.Serialization.ISerializable

Class MultiErrorException

The exception that represents multiple errors.

Inheritance

System.Object
System.Exception
MultiErrorException

Implements

System.Runtime.Serialization.ISerializable

Inherited Members

System.Exception.GetBaseException()
System.Exception.GetObjectData(System.Runtime.Serialization.SerializationInfo, System.Runtime.Serialization.StreamingContext)
System.Exception.GetType()
System.Exception.ToString()
System.Exception.Data
System.Exception.HelpLink
System.Exception.HResult
System.Exception.InnerException
System.Exception.Message
System.Exception.Source
System.Exception.StackTrace
System.Exception.TargetSite
System.Exception.SerializeObjectState
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [PerpetualIntelligence.Shared.Exceptions](#)

Assembly: [PerpetualIntelligence.Shared.dll](#)

Syntax

```
public class MultiErrorException : Exception, ISerializable
```

Constructors

MultiErrorException()

Initializes a new instance.

Declaration

```
public MultiErrorException()
```

MultiErrorException(Error[], String)

Initialize a new instance with the specified errors and an exception message.

Declaration

```
public MultiErrorException(Error[] errors, string message)
```

Parameters

TYPE	NAME	DESCRIPTION
Error[]	errors	The errors.
System.String	message	The exception message.

MultiErrorException(IEnumerable<Error>)

Initialize a new instance with the specified errors.

Declaration

```
public MultiErrorException(IEnumerable<Error> errors)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<Error>	errors	The errors.

Properties

Errors

The errors in this exception.

Declaration

```
public IEnumerable<Error> Errors { get; }
```

Property Value

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<Error>	

Implements

System.Runtime.Serialization.ISerializable

Namespace PerpetualIntelligence.Shared.Extensions

Classes

[IEnumerableExtensions](#)

System.Collections.Generic.IEnumerable<T> extension methods.

[ILoggerExtensions](#)

The Microsoft.Extensions.Logging.ILogger<TCategoryName> extension methods.

[JsonElementExtensions](#)

System.Text.Json.JsonElement extension methods

[StringExtensions](#)

System.String extension methods.

Class IEnumerableExtensions

System.Collections.Generic.IEnumerable<T> extension methods.

Inheritance

System.Object
IEnumerableExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Extensions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public static class IEnumerableExtensions
```

Methods

HasDuplicates<T, TProp>(IEnumerable<T>, Func<T, TProp>)

Determines whether the sequence contains duplicate elements.

Declaration

```
[WriteUnitTest]  
public static bool HasDuplicates<T, TProp>(this IEnumerable<T> list, Func<T, TProp> selector)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<T>	list	The sequence to check.
System.Func<T, TProp>	selector	The selector predicate.

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
T	The type of element.

NAME	DESCRIPTION
TProp	The selector predicate.

IsNullOrEmpty<T>(IEnumerable<T>)

Determines whether the sequence is null or empty.

Declaration

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

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<T>	list	The sequence to check.

Returns

TYPE	DESCRIPTION
System.Boolean	

Type Parameters

NAME	DESCRIPTION
T	

Class ILoggerExtensions

The Microsoft.Extensions.Logging.ILogger<TCategoryName> extension methods.

Inheritance

System.Object
ILoggerExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Extensions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public static class ILoggerExtensions
```

Methods

FormatAndLog(ILogger, LogLevel, LoggingOptions, String, Object[])

Logs the formatted message and returns the logged message for downstream processing.

Declaration

```
public static string FormatAndLog(this ILogger logger, LogLevel logLevel, LoggingOptions loggingOptions, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	logger	
Microsoft.Extensions.Logging.LogLevel	logLevel	
LoggingOptions	loggingOptions	
System.String	message	
System.Object[]	args	

Returns

TYPE	DESCRIPTION
System.String	

Class JsonElementExtensions

System.Text.Json.JsonElement extension methods

Inheritance

System.Object
JsonElementExtensions

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Extensions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[WriteUnitTest]
public static class JsonElementExtensions
```

Methods

ToClaims(JsonElement, String, String[])

Converts the System.Text.Json.JsonElement to a list of System.Security.Claims.Claim.

Declaration

```
public static Claim[] ToClaims(this JsonElement json, string issuer = null, params string[] excludeKeys)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.Json.JsonElement	json	The System.Text.Json.JsonElement containing claims.
System.String	issuer	Optional issuer name to add to claims.
System.String[]	excludeKeys	Claims to exclude.

Returns

TYPE	DESCRIPTION
System.Security.Claims.Claim[]	An array of System.Security.Claims.Claim.

Remarks

All the claim values are converted to string.

TryGetBoolean(JsonElement, String)

Attempts to get a boolean from a System.Text.Json.JsonElement.

Declaration

```
public static bool? TryGetBoolean(this JsonElement json, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.Json.JsonElement	json	The System.Text.Json.JsonElement
System.String	name	The name of the property to find.

Returns

TYPE	DESCRIPTION
System.Nullable<System.Boolean>	

TryGetInt(JsonElement, String)

Attempts to get an int from a System.Text.Json.JsonElement.

Declaration

```
public static int? TryGetInt(this JsonElement json, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.Json.JsonElement	json	The System.Text.Json.JsonElement.
System.String	name	The name of the property to find.

Returns

TYPE	DESCRIPTION
System.Nullable<System.Int32>	

TryGetString(JsonElement, String)

Attempts to get a string from a System.Text.Json.JsonElement.

Declaration

```
public static string TryGetString(this JsonElement json, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.Json.JsonElement	json	The System.Text.Json.JsonElement.
System.String	name	The name of the property to find.

Returns

TYPE	DESCRIPTION
System.String	

TryGetStringArray(JsonElement, String)

Attempts to get an array of string from a System.Text.Json.JsonElement.

Declaration

```
public static IEnumerable<string> TryGetStringArray(this JsonElement json, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.Json.JsonElement	json	The System.Text.Json.JsonElement.
System.String	name	The name of the property to find.

Returns

TYPE	DESCRIPTION
System.Collections.Generic.IEnumerable<System.String>	

TryGetValue(JsonElement, String)

Attempts to get a value from a System.Text.Json.JsonElement.

Declaration

```
public static JsonElement TryGetValue(this JsonElement json, string name)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Text.Json.JsonElement	json	The System.Text.Json.JsonElement.
System.String	name	The name of the property to find.

Returns

TYPE	DESCRIPTION
System.Text.Json.JsonElement	

Class StringExtensions

System.String extension methods.

Inheritance

System.Object

StringExtensions

Inherited Members

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Extensions](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public static class StringExtensions
```

Methods

EmptyIfNull(String)

Returns an empty string if this string is null.

Declaration

```
public static string EmptyIfNull(this string self)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to check.

Returns

TYPE	DESCRIPTION
System.String	<code>true</code> if string is not null or white space, otherwise <code>false</code> .

IsNotNullOrWhiteSpace(String)

Determines if the string is not null or white space.

Declaration

```
public static bool IsNotNullOrWhiteSpace(this string self)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to check.

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if string is not null or white space, otherwise <code>false</code> .

IsNullOrEmpty(String)

Determines if the string is null or empty.

Declaration

```
public static bool IsNullOrEmpty(this string self)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to check.

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if string is null or empty, otherwise <code>false</code> .

IsNullOrWhiteSpace(String)

Determines if the string is null or white space.

Declaration

```
public static bool IsNullOrWhiteSpace(this string self)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to check.

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if string is null or empty, otherwise <code>false</code> .

JoinByNewline(IEnumerable<String>)

Joins the string collection with a System.Environment.NewLine separator.

Declaration

```
public static string JoinByNewline(this IEnumerable<string> list)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<System.String>	list	A collection of string to join.

Returns

TYPE	DESCRIPTION
System.String	A joined string.

JoinBySpace(IEnumerable<String>)

Joins the string collection with a space separator.

Declaration

```
public static string JoinBySpace(this IEnumerable<string> list)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<System.String>	list	A collection of string to join.

Returns

TYPE	DESCRIPTION
System.String	A joined string.

NotNull(String)

Returns a non null string.

Declaration

```
public static string NotNull(this string self)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to check.

Returns

TYPE	DESCRIPTION
System.String	<code>true</code> if string is not null or white space, otherwise <code>false</code> .

Repeat(String, Int32)

Repeats the string for the specified times.

Declaration

```
public static string Repeat(this string value, int count)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	The value to repeat.
System.Int32	count	The count.

Returns

TYPE	DESCRIPTION
System.String	New repeated string.

SplitByNewline(String)

Splits the string with a System.Environment.NewLine

Declaration

```
public static string[] SplitByNewline(this string self)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to split.

Returns

TYPE	DESCRIPTION
System.String[]	An array of string split by newline separator.

SplitBySpace(String)

Splits the string with a space separator.

Declaration

```
public static string[] SplitBySpace(this string self)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to split.

Returns

TYPE	DESCRIPTION
System.String[]	An array of string split by space separator.

TrimEnd(String, String, StringComparison)

Removes the trailing occurrence of a string recursively.

Declaration

```
public static string TrimEnd(this string self, string trim, StringComparison stringComparison)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to trim.
System.String	trim	The string to trim from start.
System.StringComparison	stringComparison	The string comparison.

Returns

TYPE	DESCRIPTION
System.String	

TrimStart(String, String, StringComparison)

Removes the leading occurrence of a string recursively.

Declaration

```
public static string TrimStart(this string self, string trim, StringComparison stringComparison)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	self	The string to trim.

TYPE	NAME	DESCRIPTION
System.String	trim	The string to trim from start.
System.StringComparison	stringComparison	The string comparison.

Returns

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Shared.Infrastructure

Classes

CustomDataTypes

The custom data types to support System.ComponentModel.DataAnnotations.DataType.Custom.

Error

The generic error.

Identity

Identifies an entity uniquely with an `id` and an optional `partition_id`, if the entity is partitioned.

LocalStorageKeys

The `oneimlx` local storage keys.

LoggingOptions

The generic logging configuration options.

ModelIdentity

An [Identity](#) with mode support.

Modes

Defines the commonly used modes.

OrgConstants

The Perpetual Intelligence organizational constants.

RegexPatterns

Defines commonly used REGEX patterns.

TryResultOrError<T>

The generic result of a trying method. The trying method must return an error or a valid result. Both [Error](#) and [Result](#) cannot be null.

Delegates

ResultDelegate<TContext, TResult>

The delegate that returns a result for the specified context.

Class CustomDataTypes

The custom data types to support System.ComponentModel.DataAnnotations.DataType.Custom.

Inheritance

System.Object
CustomDataTypes

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public sealed class CustomDataTypes
```

Fields

Bool

The System.Boolean custom data type.

Declaration

```
public const string Bool = "urn:oneimlx:cdt:bool"
```

Field Value

TYPE	DESCRIPTION
System.String	

Double

The System.Double custom data type.

Declaration

```
public const string Double = "urn:oneimlx:cdt:double"
```

Field Value

TYPE	DESCRIPTION
System.String	

Enumeration

The System.Enum custom data type.

Declaration

```
public const string Enumeration = "urn:oneimlx:cdt:enum"
```

Field Value

TYPE	DESCRIPTION
System.String	

Float

The System.Single custom data type.

Declaration

```
public const string Float = "urn:oneimlx:cdt:float"
```

Field Value

TYPE	DESCRIPTION
System.String	

Int

The System.Int32 custom data type.

Declaration

```
public const string Int = "urn:oneimlx:cdt:int"
```

Field Value

TYPE	DESCRIPTION
System.String	

Long

The System.Int64 custom data type.

Declaration

```
public const string Long = "urn:oneimlx:cdt:long"
```

Field Value

TYPE	DESCRIPTION
System.String	

MultiSelect

The multi-selection custom data type.

Declaration

```
public const string MultiSelect = "urn:oneimlx:cdt:mselect"
```

Field Value

TYPE	DESCRIPTION
System.String	

Short

The System.Int16 custom data type.

Declaration

```
public const string Short = "urn:oneimlx:cdt:short"
```

Field Value

TYPE	DESCRIPTION
System.String	

SingleSelect

The single-selection custom data type.

Declaration

```
public const string SingleSelect = "urn:oneimlx:cdt:sselect"
```

Field Value

TYPE	DESCRIPTION
System.String	

StringDictionary

The string key-value pairs custom data type.

Declaration

```
public const string StringDictionary = "urn:oneimlx:cdt:sdict"
```

Field Value

TYPE	DESCRIPTION
System.String	

StringList

The string list or collection custom data type.

Declaration

```
public const string StringList = "urn:oneimlx:cdt:slist"
```

Field Value

TYPE	DESCRIPTION
System.String	

SubObject

The sub-object custom data type.

Declaration

```
public const string SubObject = "urn:oneimlx:cdt:so"
```

Field Value

TYPE	DESCRIPTION
System.String	

SubObjectList

The sub-object list or collection custom data type.

Declaration

```
public const string SubObjectList = "urn:oneimlx:cdt:solist"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class Error

The generic error.

Inheritance

System.Object
Error

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class Error
```

Constructors

Error()

Initialize a new instance.

Declaration

```
public Error()
```

Error(String, String, Object[])

Initialize a new instance.

Declaration

```
public Error(string error, string errorDescription, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	error	The error code.
System.String	errorDescription	The error description.
System.Object[]	args	The error description format arguments.

Error(String, String, Object[], String, String)

Initialize a new instance.

Declaration

```
public Error(string error, string errorDescription, object[] args = null, string errorUri = null, string requestId = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	error	The error code.
System.String	errorDescription	The error description.
System.Object[]	args	The error description format arguments./
System.String	errorUri	The error URI.
System.String	requestId	The request id.

Fields

AlreadyExist

The not found error.

Declaration

```
[JsonIgnore]  
public const string AlreadyExist = "already_exist"
```

Field Value

TYPE	DESCRIPTION
System.String	

InvalidConfiguration

The invalid request error.

Declaration

```
[JsonIgnore]  
public const string InvalidConfiguration = "invalid_configuration"
```

Field Value

TYPE	DESCRIPTION
System.String	

InvalidRequest

The invalid request error.

Declaration

```
[JsonIgnore]
public const string InvalidRequest = "invalid_request"
```

Field Value

TYPE	DESCRIPTION
System.String	

NotFound

The not found error.

Declaration

```
[JsonIgnore]
public const string NotFound = "not_found"
```

Field Value

TYPE	DESCRIPTION
System.String	

ServerError

The server error.

Declaration

```
[JsonIgnore]
public const string ServerError = "server_request"
```

Field Value

TYPE	DESCRIPTION
System.String	

Unauthorized

The unauthorized error.

Declaration

```
[JsonIgnore]
public const string Unauthorized = "unauthorized_access"
```

Field Value

TYPE	DESCRIPTION
System.String	

Unexpected

The unexpected error.

Declaration

```
[JsonIgnore]
public const string Unexpected = "unexpected_error"
```

Field Value

TYPE	DESCRIPTION
System.String	

Properties

Args

The error description format arguments.

Declaration

```
[JsonPropertyName("args")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public object[] Args { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Object[]	

ErrorCode

The `error` code. Defaults to `unexpected_error`.

Declaration

```
[JsonPropertyName("error")]
public string ErrorCode { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ErrorDescription

The `error_description`.

Declaration

```
[JsonPropertyName("error_description")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public string ErrorDescription { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

ErrorUri

The `error_uri`.

Declaration

```
[JsonPropertyName("error_uri")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public string ErrorUri { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

RequestId

The request identifier `request_id`.

Declaration

```
[JsonPropertyName("request_id")]
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]
public string RequestId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

FormatDescription()

Format the [ErrorDescription](#) with the [Args](#).

Declaration

```
[WriteUnitTest]
public string FormatDescription()
```

Returns

TYPE	DESCRIPTION
System.String	Formatted error description.

FormatError()

Creates a new [Error](#) instance with formatted [ErrorDescription](#) with the [Args](#).

Declaration

```
[WriteUnitTest]
public Error FormatError()
```

Returns

TYPE	DESCRIPTION
Error	Formatted error instance.

MapHttpStatusCode()

Maps the [ErrorCode](#) to System.Net.HttpStatusCode

Declaration

```
public HttpStatusCode MapHttpStatusCode()
```

Returns

TYPE	DESCRIPTION
System.Net.HttpStatusCode	

setError(String, String, Object[], String, String)

Set an error.

Declaration

```
public void setError(string error, string errorDescription, object[] args = null, string errorUri = null, string requestId = null)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	error	
System.String	errorDescription	
System.Object[]	args	
System.String	errorUri	
System.String	requestId	

Class Identity

Identifies an entity uniquely with an `id` and an optional `partition_id`, if the entity is partitioned.

Inheritance

System.Object

Identity

[ModelIdentity](#)

Inherited Members

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class Identity
```

Constructors

Identity(String)

Initializes a new instance.

Declaration

```
[JsonConstructor]  
public Identity(string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The id.

Identity(String, String)

Initializes a new instance.

Declaration

```
public Identity(string partitionId, string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	partitionId	The partition id.
System.String	id	The id.

Properties

Id

The identifier.

Declaration

```
[JsonPropertyName("id")]  
public string Id { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

IsPartitioned

Determines if the identity has a `partition_id`.

Declaration

```
[JsonIgnore]  
public bool IsPartitioned { get; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

PartitionId

The partition identifier.

Declaration

```
[JsonPropertyName("partition_id")]  
[JsonIgnore(Condition = JsonIgnoreCondition.WhenWritingNull)]  
public string PartitionId { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

Methods

Equals(Identity)

Declaration

```
public bool Equals(Identity other)
```

Parameters

TYPE	NAME	DESCRIPTION
Identity	other	

Returns

TYPE	DESCRIPTION
System.Boolean	

Equals(Object)

Declaration

```
public override bool Equals(object obj)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Object	obj	

Returns

TYPE	DESCRIPTION
System.Boolean	

Overrides

System.Object.Equals(System.Object)

GetHashCode()

Declaration

```
public override int GetHashCode()
```

Returns

TYPE	DESCRIPTION
System.Int32	

Overrides

System.Object.GetHashCode()

ToString()

Declaration

```
public override string ToString()
```

Returns

TYPE	DESCRIPTION
System.String	

Overrides

System.Object.ToString()

Operators

Equality(Identity, Identity)

Declaration

```
public static bool operator ==(Identity left, Identity right)
```

Parameters

TYPE	NAME	DESCRIPTION
Identity	left	
Identity	right	

Returns

TYPE	DESCRIPTION
System.Boolean	

Inequality(Identity, Identity)

Declaration

```
public static bool operator !=(Identity left, Identity right)
```

Parameters

TYPE	NAME	DESCRIPTION
Identity	left	
Identity	right	

Returns

TYPE	DESCRIPTION
System.Boolean	

Class LocalStorageKeys

The `oneimlx` local storage keys.

Inheritance

System.Object

LocalStorageKeys

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ReferenceEquals(System.Object, System.Object)
- System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public sealed class LocalStorageKeys
```

Remarks

NOTE: This class is part of the Perpetual Intelligence infrastructure. Please do not use it directly in your application code.

Fields

IsDiagnostic

Determines the diagnostic state from the local storage.

Declaration

```
public const string IsDiagnostic = "urn:oneimlx:lsk:isdiag"
```

Field Value

TYPE	DESCRIPTION
System.String	

IsLocalizationPrefixed

Determines whether to show the localizer profix.

Declaration

```
public const string IsLocalizationPrefixed = "urn:oneimlx:lsk:isl10prefixed"
```

Field Value

TYPE	DESCRIPTION
System.String	

LayerLuminance

Determines the layer luminance for the `fast` color theme.

Declaration

```
public const string LayerLuminance = "urn:oneimlx:lsk:layer"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class LoggingOptions

The generic logging configuration options.

Inheritance

- System.Object
- LoggingOptions
- [HostingOptions](#)
- [LoggingOptions](#)

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ReferenceEquals(System.Object, System.Object)
- System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class LoggingOptions
```

Properties

ObscureErrorArgumentString

The string used to obscure error arguments. Defaults to `****`.

Declaration

```
public string ObscureErrorArgumentString { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.String	

See Also

[ObscureErrorArguments](#)

ObscureErrorArguments

Obscures the arguments in the error description to hide the sensitive data. Defaults to `true`.

Declaration

```
public bool ObscureErrorArguments { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Boolean	

Remarks

Use [ObscureErrorArgumentString](#) to configure the obscure string.

See Also

[ObscureErrorArgumentString](#)

Class ModelIdentity

An [Identity](#) with mode support.

Inheritance

System.Object

[Identity](#)

ModelIdentity

Inherited Members

[Identity.Id](#)

[Identity.IsPartitioned](#)

[Identity.PartitionId](#)

[Identity.Equals\(Object\)](#)

[Identity.Equals\(Identity\)](#)

[Identity.GetHashCode\(\)](#)

[Identity.ToString\(\)](#)

System.Object.Equals(System.Object, System.Object)

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public sealed class ModelIdentity : Identity
```

Constructors

ModelIdentity(String, String, String)

Declaration

```
public ModelIdentity(string mode, string partitionId, string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	mode	
System.String	partitionId	
System.String	id	

Properties

Mode

The mode.

Declaration

```
[JsonPropertyName("mode")]
public string Mode { get; }
```

Property Value

TYPE	DESCRIPTION
System.String	

See Also

[Modes](#)

Class Modes

Defines the commonly used modes.

Inheritance

System.Object
Modes

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public static class Modes
```

Fields

Live

Live or production mode.

Declaration

```
public const string Live = "urn:oneimlx:mode:live"
```

Field Value

TYPE	DESCRIPTION
System.String	

Neutral

Neutral or no mode.

Declaration

```
public const string Neutral = "urn:oneimlx:mode:neutral"
```

Field Value

TYPE	DESCRIPTION
System.String	

Stage

Staging or pre-production mode.

Declaration

```
public const string Stage = "urn:oneimlx:mode:stage"
```

Field Value

TYPE	DESCRIPTION
System.String	

Test

Test or sandbox mode.

Declaration

```
public const string Test = "urn:oneimlx:mode:test"
```

Field Value

TYPE	DESCRIPTION
System.String	

Methods

All()

Returns all the supported modes.

Declaration

```
public static string[] All()
```

Returns

TYPE	DESCRIPTION
System.String[]	Array of all the supported modes.

IsTestOrLive(String)

Determines if the specified mode is [Test](#) or [Live](#).

Declaration

```
public static bool IsTestOrLive(string mode)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	mode	

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the specified mode is Test or Live , otherwise <code>false</code> .

IsValid(String)

Determines if the mode is valid.

Declaration

```
public static bool IsValid(string mode)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	mode	

Returns

TYPE	DESCRIPTION
System.Boolean	<code>true</code> if the specified mode is valid, otherwise <code>false</code> .

ThrowIfInvalid(String)

Throws an exception if the specified mode is not valid.

Declaration

```
public static void ThrowIfInvalid(string mode)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	mode	Environment to validate.

See Also

[ModelIdentity](#)

Class OrgConstants

The Perpetual Intelligence organizational constants.

Inheritance

System.Object
OrgConstants

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public static class OrgConstants
```

Remarks

NOTE: This class is part of the Perpetual Intelligence infrastructure. Please do not use it directly in your application code.

Fields

FullLogoUrl

The full logo URL.

Declaration

```
public const string FullLogoUrl =  
"https://github.com/perpetualintelligence/branding/blob/main/Logo/pi_full_575_256.png"
```

Field Value

TYPE	DESCRIPTION
System.String	

FullName

The full name.

Declaration

```
public const string FullName = "Perpetual Intelligence L.L.C."
```

Field Value

TYPE	DESCRIPTION
System.String	

GitHub

The `GitHub` URL.

Declaration

```
public const string GitHub = "https://github.com/perpetualintelligence"
```

Field Value

TYPE	DESCRIPTION
System.String	

LogoUrl

The logo URL.

Declaration

```
public const string LogoUrl = "https://github.com/perpetualintelligence/branding/blob/main/Logo/icon.png"
```

Field Value

TYPE	DESCRIPTION
System.String	

Policies

The terms of use, operation and data policies.

Declaration

```
public const string Policies = "https://terms.perpetualintelligence.com/articles/intro.html"
```

Field Value

TYPE	DESCRIPTION
System.String	

ShortName

The short name.

Declaration

```
public const string ShortName = "Perpetual Intelligence"
```

Field Value

TYPE	DESCRIPTION
System.String	

Class RegexPatterns

Defines commonly used REGEX patterns.

Inheritance

System.Object
RegexPatterns

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
[WriteUnitTest]
public static class RegexPatterns
```

Fields

Letters

Pattern containing only letters a-z or A-Z.

Declaration

```
public const string Letters = "^[a-zA-Z]+$"
```

Field Value

TYPE	DESCRIPTION
System.String	

LettersNumbersDashUnderscore

Pattern containing only letters a-z or A-Z, numbers 0-9, dash, and underscore.

Declaration

```
public const string LettersNumbersDashUnderscore = "^[A-Za-z0-9- _?]+$"
```

Field Value

TYPE	DESCRIPTION
System.String	

LettersNumbersDashUnderscoreComma

Pattern containing only letters a-z or A-Z, numbers 0-9, dash, underscore and comma.

Declaration

```
public const string LettersNumbersDashUnderscoreComma = "^[A-Za-z0-9-]_,?)+$"
```

Field Value

TYPE	DESCRIPTION
System.String	

LettersNumbersDashUnderscoreSpace

Pattern containing only letters a-z or A-Z, numbers 0-9, dash, and underscore.

Declaration

```
public const string LettersNumbersDashUnderscoreSpace = "^[A-Za-z0-9][\\-\\h_]?)+$"
```

Field Value

TYPE	DESCRIPTION
System.String	

Delegate ResultDelegate<TContext, TResult>

The delegate that returns a result for the specified context.

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public delegate Task<TResult> ResultDelegate<TContext, TResult>(TContext context)
    where TContext : class where TResult : class;
```

Parameters

TYPE	NAME	DESCRIPTION
TContext	context	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task<TResult>	

Type Parameters

NAME	DESCRIPTION
TContext	
TResult	

Class TryResultOrError<T>

The generic result of a trying method. The trying method must return an error or a valid result. Both [Error](#) and [Result](#) cannot be null.

Inheritance

System.Object
TryResultOrError<T>

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Infrastructure](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public sealed class TryResultOrError<T>
    where T : class
```

Type Parameters

NAME	DESCRIPTION
T	The result type.

Constructors

TryResultOrError(T)

Initialize a new instance.

Declaration

```
public TryResultOrError(T result)
```

Parameters

TYPE	NAME	DESCRIPTION
T	result	

TryResultOrError(Error)

Initialize a new instance.

Declaration

```
public TryResultOrError(Error error)
```

Parameters

TYPE	NAME	DESCRIPTION
Error	error	

Properties

Error

The result of a try.

Declaration

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

Property Value

TYPE	DESCRIPTION
Error	

Result

The result of a try.

Declaration

```
public T Result { get; }
```

Property Value

TYPE	DESCRIPTION
T	

Namespace PerpetualIntelligence.Shared.Patterns

Classes

[Decorator<TObject>](#)

Decorator is a structural pattern that allows adding new behaviors to objects dynamically by placing them inside special wrapper objects. Using decorators you can wrap objects countless number of times since both target objects and decorators follow the same interface.

[DecoratorService<TService, TImpl>](#)

[Decorator<TObject>](#) service for dependency injection.

[DisposableDecorator<TObject>](#)

A disposable [Decorator<TObject>](#).

[DisposableDecoratorService<TService, TImpl>](#)

The disposable [DisposableDecorator<TObject>](#) service for dependency injection.

Class Decorator<TObject>

Decorator is a structural pattern that allows adding new behaviors to objects dynamically by placing them inside special wrapper objects. Using decorators you can wrap objects countless number of times since both target objects and decorators follow the same interface.

Inheritance

System.Object
Decorator<TObject>
[DecoratorService<TService, TImpl>](#)
[DisposableDecorator<TObject>](#)

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Patterns](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class Decorator<TObject>
```

Type Parameters

NAME	DESCRIPTION
TObject	The inner object type.

Constructors

Decorator(TObject)

Initializes a new instance of decorator.

Declaration

```
public Decorator(TObject object)
```

Parameters

TYPE	NAME	DESCRIPTION
TObject	object	The inner object.

Properties

Object

The inner object.

Declaration

```
public TObject Object { get; set; }
```

Property Value

TYPE	DESCRIPTION
TObject	

See Also

https://en.wikipedia.org/wiki/Decorator_pattern

Class DecoratorService<TService, TImpl>

[Decorator<TObject>](#) service for dependency injection.

Inheritance

System.Object
[Decorator<TService>](#)
DecoratorService<TService, TImpl>

Inherited Members

[Decorator<TService>.Object](#)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Patterns](#)
Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class DecoratorService<TService, TImpl> : Decorator<TService> where TImpl : class, TService
```

Type Parameters

NAME	DESCRIPTION
TService	The service type.
TImpl	The service implementation type.

Constructors

DecoratorService(TImpl)

Initialize a new instance.

Declaration

```
public DecoratorService(TImpl implementation)
```

Parameters

TYPE	NAME	DESCRIPTION
TImpl	implementation	The service implementation.

Class DisposableDecorator<TObject>

A disposable [Decorator<TObject>](#).

Inheritance

System.Object
[Decorator<TObject>](#)
DisposableDecorator<TObject>
[DisposableDecoratorService<TService, TImpl>](#)

Implements

System.IDisposable

Inherited Members

[Decorator<TObject>.Object](#)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Patterns](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class DisposableDecorator<TObject> : Decorator<TObject>, IDisposable
```

Type Parameters

NAME	DESCRIPTION
TObject	The inner object type. It can be System.IDisposable.

Constructors

DisposableDecorator(TObject)

Initialize a new instance of decorator.

Declaration

```
public DisposableDecorator(TObject object)
```

Parameters

TYPE	NAME	DESCRIPTION
TObject	object	The inner object.

Methods

Dispose()

Declaration

```
public virtual void Dispose()
```

Implements

System.IDisposable

Class DisposableDecoratorService<TService, TImpl>

The disposable [DisposableDecorator<TObject>](#) service for dependency injection.

Inheritance

System.Object
[Decorator<TService>](#)
[DisposableDecorator<TService>](#)
DisposableDecoratorService<TService, TImpl>

Implements

System.IDisposable

Inherited Members

[DisposableDecorator<TService>.Dispose\(\)](#)
[Decorator<TService>.Object](#)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Patterns](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public class DisposableDecoratorService<TService, TImpl> : DisposableDecorator<TService>, IDisposable where
TImpl : class, TService
```

Type Parameters

NAME	DESCRIPTION
TService	The service type.
TImpl	The service implementation type.

Constructors

DisposableDecoratorService(TImpl)

Initialize a new instance.

Declaration

```
public DisposableDecoratorService(TImpl implementation)
```

Parameters

TYPE	NAME	DESCRIPTION
TImpl	implementation	The service implementation.

Implements

System.IDisposable

Namespace PerpetualIntelligence.Shared.Services

Classes

[InfraHelper](#)

Provides low level infrastructure helper methods shared across application stack.

[UrlSafeBase64Encoder](#)

The URL safe Base64 string encoder. The encoder trims the padding character `=`, replacing the characters `+` and `/` with `-` (dash) and `_` (underscore), respectively.

Class InfraHelper

Provides low level infrastructure helper methods shared across application stack.

Inheritance

System.Object
InfraHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Services](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public static class InfraHelper
```

Methods

EnsureResultAsync<TContext, TResult>(ResultDelegate<TContext, TResult>, TContext)

Ensures an action returns a result or an [Error](#) but does not throw any exception.

Declaration

```
public static async Task<TryResultOrError<TResult>> EnsureResultAsync<TContext, TResult>  
(ResultDelegate<TContext, TResult> action, TContext context)  
    where TContext : class where TResult : class
```

Parameters

TYPE	NAME	DESCRIPTION
ResultDelegate <TContext, TResult>	action	The action to execute.
TContext	context	The action context.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task< TryResultOrError <TResult>>	TryResultOrError <T> instance that contains the result or an Error instance.

Type Parameters

NAME	DESCRIPTION
TContext	The context type.

NAME	DESCRIPTION
TResult	The result type.

Format(LoggingOptions, String, Object[])

Formats the error message for downstream processing.

Declaration

```
public static string Format(LoggingOptions loggingOptions, string message, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
LoggingOptions	loggingOptions	The logging options. See LoggingOptions .
System.String	message	The message to format.
System.Object[]	args	The format arguments.

Returns

TYPE	DESCRIPTION
System.String	The formatted error message.

GetParent(String)

Gets the parent directory of a specified path or sub-directory.

Declaration

```
public static string GetParent(string path)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	Path or sub-directory.

Returns

TYPE	DESCRIPTION
System.String	The parent directory returned by System.IO.DirectoryInfo.Parent property or null if file path denotes a root.

Exceptions

TYPE	CONDITION
System.ArgumentException	Path is null or empty.

See Also

System.IO.Directory.GetParent(String)

GetParent(String, UInt32)

Gets the parent directory at the specified level of a specified path or sub-directory.

Declaration

```
public static string GetParent(string path, uint level)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	path	Path or sub-directory.
System.UInt32	level	The levels to go up.

Returns

TYPE	DESCRIPTION
System.String	The parent directory returned by System.IO.DirectoryInfo.Parent for specified level or null if file path denotes a root.

Exceptions

TYPE	CONDITION
System.ArgumentException	Path is null or empty.

MinPositiveOrZero(Int32, Int32)

Returns a minimum positive of the two numbers or zero.

Declaration

```
public static int MinPositiveOrZero(int num1, int num2)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	num1	First number to check.
System.Int32	num2	Second number to check.

TYPE	NAME	DESCRIPTION

Returns

TYPE	DESCRIPTION
System.Int32	A minimum positive number or zero.

Obscure(String, Object[])

Obscures the arguments based on the specified mask.

Declaration

```
public static object[] Obscure(string mask, params object[] args)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	mask	The obscure mask.
System.Object[]	args	The arguments to obscure.

Returns

TYPE	DESCRIPTION
System.Object[]	The obscured arguments.

Class UriSafeBase64Encoder

The URL safe Base64 string encoder. The encoder trims the padding character `=`, replacing the characters `+` and `/` with `-` (dash) and `_` (underscore), respectively.

Inheritance

System.Object
UriSafeBase64Encoder

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Shared.Services](#)

Assembly: PerpetualIntelligence.Shared.dll

Syntax

```
public static class UriSafeBase64Encoder
```

Methods

Decode(String)

Decodes the specified string.

Declaration

```
public static byte[] Decode(string value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	The Base64 string.

Returns

TYPE	DESCRIPTION
Byte[]	The decoded byte array.

Exceptions

TYPE	CONDITION
System.InvalidOperationException	Invalid Base64 string.

Encode(Byte[])

Encodes the specified byte array.

Declaration

```
public static string Encode(byte[] value)
```

Parameters

TYPE	NAME	DESCRIPTION
Byte[]	value	The byte array.

Returns

TYPE	DESCRIPTION
System.String	The encoded Base64 string.

Utf8Decode(String)

Decodes the specified string and then converts the bytes to a UTF8 string.

Declaration

```
public static string Utf8Decode(string arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	arg	

Returns

TYPE	DESCRIPTION
System.String	The URL safe string with UTF8 encoding.

Utf8Encode(String)

Encodes the specified string with System.Text.Encoding.UTF8 and then converts the UTF8 bytes to Url safe string.

Declaration

```
public static string Utf8Encode(string arg)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	arg	

Returns

TYPE	DESCRIPTION
System.String	The URL safe string with UTF8 encoding.

Namespace PerpetualIntelligence.Test

Classes

[ContextTests](#)

Represents a test class that provides [TestContext](#).

[HttpClientTests](#)

Represents a [ContextTests](#) that provides the [HttpClient](#).

[InitializerTests](#)

Represents a [ContextTests](#) that provides the initialization and cleanup methods.

[LoggerTests<T>](#)

Represents a [ContextTests](#) that provides the Microsoft.Extensions.Logging.ILogger.

[TestIdAttribute](#)

Represents a test id.

Class ContextTests

Represents a test class that provides [TestContext](#).

Inheritance

- System.Object
- ContextTests
- [HttpClientTests](#)
- [InitializerTests](#)
- [LoggerTests<T>](#)

Inherited Members

- System.Object.Equals(System.Object)
- System.Object.Equals(System.Object, System.Object)
- System.Object.GetHashCode()
- System.Object.GetType()
- System.Object.MemberwiseClone()
- System.Object.ReferenceEquals(System.Object, System.Object)
- System.Object.ToString()

Namespace: [PerpetualIntelligence.Test](#)
Assembly: PerpetualIntelligence.Test.dll

Syntax

```
public abstract class ContextTests
```

Properties

TestContext

The test context. This property is set automatically by the test engine.

Declaration

```
public TestContext TestContext { get; set; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.VisualStudio.TestTools.UnitTesting.TestContext	

Class HttpClientTests

Represents a [ContextTests](#) that provides the [HttpClient](#).

Inheritance

System.Object
[ContextTests](#)
HttpClientTests

Inherited Members

[ContextTests.TestContext](#)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Test](#)
Assembly: PerpetualIntelligence.Test.dll

Syntax

```
public abstract class HttpClientTests : ContextTests
```

Constructors

HttpClientTests()

Initialize a new instance.

Declaration

```
protected HttpClientTests()
```

HttpClientTests(ILogger)

Initialize a new instance.

Declaration

```
public HttpClientTests(ILogger logger)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	logger	

Properties

HttpClient

The HTTP client..

Declaration

```
protected HttpClient HttpClient { get; set; }
```

Property Value

TYPE	DESCRIPTION
System.Net.Http.HttpClient	

Logger

The logger.

Declaration

```
public ILogger Logger { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

Methods

GetTestServer()

Gets the test server.

Declaration

```
protected abstract TestServer GetTestServer()
```

Returns

TYPE	DESCRIPTION
Microsoft.AspNetCore.TestHost.TestServer	

ImlxLogTestCleanup()

The test cleanup method that logs and calls [OnTestCleanup\(\)](#).

Declaration

```
[TestCleanup]
public void ImlxLogTestCleanup()
```

ImlxLogTestInitialize()

The test initialize method that logs and calls [OnTestInitialize\(\)](#).

Declaration

```
[TestInitialize]
public void ImlxLogTestInitialize()
```

OnTestCleanup()

On test cleanup callback.

Declaration

```
protected virtual void OnTestCleanup()
```

OnTestInitialize()

On test initialize callback.

Declaration

```
protected virtual void OnTestInitialize()
```

Class InitializerTests

Represents a [ContextTests](#) that provides the initialization and cleanup methods.

Inheritance

System.Object

[ContextTests](#)

InitializerTests

Inherited Members

[ContextTests.TestContext](#)

System.Object.Equals(System.Object)

System.Object.Equals(System.Object, System.Object)

System.Object.GetHashCode()

System.Object.GetType()

System.Object.MemberwiseClone()

System.Object.ReferenceEquals(System.Object, System.Object)

System.Object.ToString()

Namespace: [PerpetualIntelligence.Test](#)

Assembly: PerpetualIntelligence.Test.dll

Syntax

```
public abstract class InitializerTests : ContextTests
```

Constructors

InitializerTests()

Initialize a new instance.

Declaration

```
public InitializerTests()
```

InitializerTests(ILogger)

Initialize a new instance.

Declaration

```
public InitializerTests(ILogger logger)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	logger	

Properties

Logger

The logger.

Declaration

```
public ILogger Logger { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

Methods

Cleanup()

The test cleanup method that logs and calls [OnTestCleanup\(\)](#).

Declaration

```
[TestCleanup]
public void Cleanup()
```

Initialize()

The test initialize method that logs and calls [OnTestInitialize\(\)](#).

Declaration

```
[TestInitialize]
public void Initialize()
```

OnTestCleanup()

On test cleanup callback.

Declaration

```
protected virtual void OnTestCleanup()
```

OnTestInitialize()

On test initialize callback.

Declaration

```
protected virtual void OnTestInitialize()
```


Class LoggerTests<T>

Represents a [ContextTests](#) that provides the Microsoft.Extensions.Logging.ILogger.

Inheritance

System.Object
[ContextTests](#)
LoggerTests<T>

Inherited Members

[ContextTests.TestContext](#)
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Test](#)
Assembly: PerpetualIntelligence.Test.dll

Syntax

```
public abstract class LoggerTests<T> : ContextTests
```

Type Parameters

NAME	DESCRIPTION
T	

Constructors

LoggerTests()

Initialize a new instance.

Declaration

```
public LoggerTests()
```

LoggerTests(ILogger)

Initialize a new instance.

Declaration

```
public LoggerTests(ILogger logger)
```

Parameters

TYPE	NAME	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	logger	

Properties

Logger

The logger.

Declaration

```
public ILogger Logger { get; }
```

Property Value

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger	

Methods

Cleanup()

The test cleanup method that logs and calls [OnTestCleanup\(\)](#).

Declaration

```
[TestCleanup]  
public void Cleanup()
```

Initialize()

The test initialize method that logs and calls [OnTestInitialize\(\)](#).

Declaration

```
[TestInitialize]  
public void Initialize()
```

OnTestCleanup()

On test cleanup callback.

Declaration

```
protected virtual void OnTestCleanup()
```

OnTestInitialize()

On test initialize callback.

Declaration

```
protected virtual void OnTestInitialize()
```

Class TestIdAttribute

Represents a test id.

Inheritance

System.Object
System.Attribute
TestIdAttribute

Inherited Members

System.Attribute.Equals(System.Object)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttribute(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Boolean)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type)
System.Attribute.GetCustomAttributes(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.GetHashCode()
System.Attribute.IsDefaultAttribute()
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type)
System.Attribute.IsDefined(System.Reflection.Assembly, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.MemberInfo, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.Module, System.Type)
System.Attribute.IsDefined(System.Reflection.Module, System.Type, System.Boolean)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type)
System.Attribute.IsDefined(System.Reflection.ParameterInfo, System.Type, System.Boolean)
System.Attribute.Match(System.Object)
System.Attribute.TypeId
System.Object.Equals(System.Object, System.Object)
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Test](#)

Syntax

```
[AttributeUsage(AttributeTargets.Method, AllowMultiple = false, Inherited = false)]
public sealed class TestIdAttribute : Attribute
```

Remarks

[TestIdAttribute](#) is part of the Perpetual Intelligence infrastructure. Please do not use it directly in your application code.

Constructors

TestIdAttribute(String)

Initialize a new instance.

Declaration

```
public TestIdAttribute(string id)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	id	The test id.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	

Properties

Id

The test id.

Declaration

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

Property Value

TYPE	DESCRIPTION
System.String	

Namespace PerpetualIntelligence.Test.Services

Classes

[TestHelper](#)

This class provides helper methods for unit tests.

[TestLogger](#)

The test logger provider.

Class TestHelper

This class provides helper methods for unit tests.

Inheritance

System.Object
TestHelper

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Test.Services](#)
Assembly: PerpetualIntelligence.Test.dll

Syntax

```
public static class TestHelper
```

Remarks

[TestHelper](#) is part of the Perpetual Intelligence infrastructure. Please do not use it directly in your application code.

Methods

AssertAnyOf(String, String[])

Asserts the value is any of the specified values.

Declaration

```
public static void AssertAnyOf(string value, params string[] anyOf)
```

Parameters

TYPE	NAME	DESCRIPTION
System.String	value	The value to check.
System.String[]	anyOf	Allowed values.

AssertApiExplorer(MemberInfo, String, Boolean, Boolean)

Asserts the value is any of the given collection.

Declaration

```
public static void AssertApiExplorer(MemberInfo member, string groupName, bool ignoreApi, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	groupName	
System.Boolean	ignoreApi	
System.Boolean	inherit	

AssertApiVersion(MemberInfo, String, Boolean)

Asserts specified number of attributes are applied on the specified member.

Declaration

```
public static void AssertApiVersion(MemberInfo member, string version, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	version	
System.Boolean	inherit	

AssertArchitectureCommentAttribute(MemberInfo, String, String, String, Boolean)

Asserts [ArchitectureAttribute](#) is applied on the specified member.

Declaration

```
public static void AssertArchitectureCommentAttribute(MemberInfo member, string description, string version, string severity, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	description	
System.String	version	
System.String	severity	
System.Boolean	inherit	

AssertArrayEquals<TValue>(TValue[], TValue[])

Declaration

```
public static void AssertArrayEquals<TValue>(TValue[] expected, TValue[] actual)
```

Parameters

TYPE	NAME	DESCRIPTION
TValue[]	expected	
TValue[]	actual	

Type Parameters

NAME	DESCRIPTION
TValue	

AssertAssemblyTypesLocation(Assembly)

Asserts that the assembly has all types in the valid files or location.

Declaration

```
public static void AssertAssemblyTypesLocation(Assembly assembly)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.Assembly	assembly	

AssertAttribute<TAttribute>(MemberInfo, Boolean)

Asserts specified number of attributes are applied on the specified member.

Declaration

```
public static void AssertAttribute<TAttribute>(MemberInfo member, bool inherit = false)
    where TAttribute : Attribute
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Boolean	inherit	

Type Parameters

NAME	DESCRIPTION
TAttribute	

AssertAttributeCount(MemberInfo, Int32, Boolean)

Asserts specified number of attributes are applied on the specified member.

Declaration

```
public static void AssertAttributeCount(MemberInfo member, int attributes, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Int32	attributes	
System.Boolean	inherit	

AssertBrowsableAttribute(MemberInfo, Object, Boolean)

Asserts System.ComponentModel.BrowsableAttribute is applied on the specified member.

Declaration

```
public static void AssertBrowsableAttribute(MemberInfo member, object value, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Object	value	
System.Boolean	inherit	

AssertConstantCount(Type, Int32)

Asserts specified type has the constants defined.

Declaration

```
public static void AssertConstantCount(Type type, int constants)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	type	
System.Int32	constants	

AssertDataTypeAttribute(MemberInfo, DataType, Boolean)

Asserts System.ComponentModel.DataAnnotations.DataTypeAttribute is applied on the specified member.

Declaration

```
public static void AssertDataTypeAttribute(MemberInfo member, DataType dataType, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.ComponentModel.DataAnnotations.DataType	dataType	

TYPE	NAME	DESCRIPTION
System.Boolean	inherit	

AssertDefaultValueAttribute(MemberInfo, Object, Boolean)

Asserts System.ComponentModel.DefaultValueAttribute is applied on the specified member.

Declaration

```
public static void AssertDefaultValueAttribute(MemberInfo member, object value, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Object	value	
System.Boolean	inherit	

AssertDisplayAttribute(MemberInfo, String, Nullable<Boolean>, String, String, Nullable<Int32>, Boolean)

Asserts System.ComponentModel.DataAnnotations.DisplayAttribute is applied on the specified member.

Declaration

```
public static void AssertDisplayAttribute(MemberInfo member, string displayName, bool? autoGenerateField, string description, string groupName, int? order, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	displayName	
System.Nullable<System.Boolean>	autoGenerateField	
System.String	description	
System.String	groupName	
System.Nullable<System.Int32>	order	
System.Boolean	inherit	

AssertEditableAttribute(MemberInfo, Boolean, Boolean, Boolean)

Asserts System.ComponentModel.DataAnnotations.EditableAttribute is applied on the specified member.

Declaration

```
public static void AssertEditableAttribute(MemberInfo member, bool allowEdit, bool allowInitialEdit, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Boolean	allowEdit	
System.Boolean	allowInitialEdit	
System.Boolean	inherit	

AssertEditorBrowsableAttribute(MemberInfo, Object, Boolean)

Asserts System.ComponentModel.EditorBrowsableAttribute is applied on the specified member.

Declaration

```
public static void AssertEditorBrowsableAttribute(MemberInfo member, object value, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Object	value	
System.Boolean	inherit	

AssertHttpMethod<TMethod>(MemberInfo, String, Boolean)

Asserts Microsoft.AspNetCore.Mvc.Routing.HttpMethodAttribute is applied on the specified member.

Declaration

```
public static void AssertHttpMethod<TMethod>(MemberInfo member, string template, bool inherit = false)
    where TMethod : HttpMethodAttribute
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	template	
System.Boolean	inherit	

Type Parameters

NAME	DESCRIPTION
TMethod	

AssertInternalTypes(Assembly, String)

Asserts the types that belong to the specified namespace are internal types.

Declaration

```
public static void AssertInternalTypes(Assembly assembly, string namespace)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.Assembly	assembly	
System.String	namespace	

AssertJsonPropertyName(MemberInfo, String, Boolean)

Asserts Microsoft.AspNetCore.Mvc.Routing.HttpMethodAttribute is applied on the specified member.

Declaration

```
public static void AssertJsonPropertyName(MemberInfo member, string name, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	name	
System.Boolean	inherit	

AssertMapToApiVersion(MemberInfo, String, Boolean)

Asserts Microsoft.AspNetCore.Mvc.Routing.HttpMethodAttribute is applied on the specified member.

Declaration

```
public static void AssertMapToApiVersion(MemberInfo member, string version, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	version	
System.Boolean	inherit	

AssertMaxLengthAttributeAttribute(MemberInfo, Nullable<Int32>, Boolean)

Asserts System.ComponentModel.DataAnnotations.MaxLengthAttribute is applied on the specified member.

Declaration

```
public static void AssertMaxLengthAttributeAttribute(MemberInfo member, int? maxLength, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Nullable<System.Int32>	maxLength	
System.Boolean	inherit	

AssertMinLengthAttributeAttribute(MemberInfo, Nullable<Int32>, Boolean)

Asserts System.ComponentModel.DataAnnotations.MinLengthAttribute is applied on the specified member.

Declaration

```
public static void AssertMinLengthAttributeAttribute(MemberInfo member, int? minLength, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Nullable<System.Int32>	minLength	
System.Boolean	inherit	

AssertNamespace(Assembly, String)

Asserts that the assembly has valid namespace based on it name.

Declaration

```
public static void AssertNamespace(Assembly assembly, string rootNamespace)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.Assembly	assembly	Assembly to check.
System.String	rootNamespace	The expected root namespace.

AssertNamespace(Type, String)

Asserts that the type has the specified namespace.

Declaration

```
public static void AssertNamespace(Type type, string namespace)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Type	type	Type to check.
System.String	namespace	Expected namespace.

AssertNamespaceTypesLocation(IEnumerable<Type>, String, String)

Asserts that the specified types have valid 1-1 src files or location based on the specified namespace.

Declaration

```
public static void AssertNamespaceTypesLocation(IEnumerable<Type> types, string namespace, string srcDir)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Collections.Generic.IEnumerable<System.Type>	types	Types to check. All types must be in same assembly.
System.String	namespace	Namespace to check.
System.String	srcDir	src directory.

AssertOneImlxError(Error, String, String)

Asserts [Error](#)

Declaration

```
public static void AssertOneImlxError(Error error, string errorCode, string errorDescription)
```

Parameters

TYPE	NAME	DESCRIPTION
Error	error	
System.String	errorCode	
System.String	errorDescription	

AssertProduces(MemberInfo, String, Boolean)

Asserts Microsoft.AspNetCore.Mvc.ProducesAttribute is applied on the specified member.

Declaration

```
public static void AssertProduces(MemberInfo member, string contentType, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	contentType	
System.Boolean	inherit	

AssertProducesDefaultResponseType(MemberInfo, Type, Boolean)

Asserts Microsoft.AspNetCore.Mvc.ProducesDefaultResponseTypeAttribute is applied on the specified member.

Declaration

```
public static void AssertProducesDefaultResponseType(MemberInfo member, Type responseType, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Type	responseType	
System.Boolean	inherit	

AssertProducesResponseType(MemberInfo, Type, Int32, Boolean)

Asserts Microsoft.AspNetCore.Mvc.ProducesDefaultResponseTypeAttribute is applied on the specified member.

Declaration

```
public static void AssertProducesResponseType(MemberInfo member, Type responseType, int statusCode, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Type	responseType	
System.Int32	statusCode	
System.Boolean	inherit	

AssertPropertiesMatchInterface(Type, Type)

To check if properties defined in the class are same as in the interface.

Declaration

```
public static void AssertPropertiesMatchInterface(Type typeClass, Type typeInterface)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Type	typeClass	
System.Type	typeInterface	

AssertRangeAttribute(MemberInfo, Object, Object, String, Boolean)

Asserts System.ComponentModel.DataAnnotations.RangeAttribute is applied on the specified member.

Declaration

```
public static void AssertRangeAttribute(MemberInfo member, object minLength, object maxLength, string message, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Object	minLength	
System.Object	maxLength	
System.String	message	
System.Boolean	inherit	

AssertRefactorAttribute(MemberInfo, String, Boolean)

Asserts Microsoft.AspNetCore.Mvc.Routing.HttpMethodAttribute is applied on the specified member.

Declaration

```
public static void AssertRefactorAttribute(MemberInfo member, string description, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	description	
System.Boolean	inherit	

AssertRegularExpressionAttribute(MemberInfo, String, Boolean)

Asserts System.ComponentModel.DataAnnotations.RegularExpressionAttribute is applied on the specified member.

Declaration

```
public static void AssertRegularExpressionAttribute(MemberInfo member, string pattern, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	pattern	
System.Boolean	inherit	

AssertRoute(MemberInfo, String, Boolean)

Asserts Microsoft.AspNetCore.Mvc.Routing.HttpMethodAttribute is applied on the specified member.

Declaration

```
public static void AssertRoute(MemberInfo member, string template, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	template	
System.Boolean	inherit	

AssertStringLengthAttribute(MemberInfo, Nullable<Int32>, Nullable<Int32>, Boolean)

Asserts System.ComponentModel.DataAnnotations.StringLengthAttribute is applied on the specified member.

Declaration

```
public static void AssertStringLengthAttribute(MemberInfo member, int? maxLength, int? minLength, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.Nullable<System.Int32>	maxLength	
System.Nullable<System.Int32>	minLength	
System.Boolean	inherit	

AssertSwaggerHideProperty(MemberInfo, String, String, Boolean)

Asserts [HiddenPropertyAttribute](#) is applied on the specified member.

Declaration

```
public static void AssertSwaggerHideProperty(MemberInfo member, string propertyName, string justification, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	propertyName	
System.String	justification	
System.Boolean	inherit	

AssertThrowsErrorException(Action, String, String)

Ensures that an action throws [ErrorException](#).

Declaration

```
public static void AssertThrowsErrorException(Action action, string errorCode, string errorDescription)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	The action to execute.
System.String	errorCode	The expected error code.
System.String	errorDescription	The expected error description.

AssertThrowsErrorExceptionAsync(Func<Task>, String, String)

Ensures that an action throws [ErrorException](#).

Declaration

```
public static async Task AssertThrowsErrorExceptionAsync(Func<Task> funcTask, string errorCode, string errorDescription)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.Threading.Tasks.Task>	funcTask	The task to execute.
System.String	errorCode	The expected error code.
System.String	errorDescription	The expected error description.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	instance that contains the result or an Error instance.

AssertThrowsMultiErrorExceptionAsync(Func<Task>, Int32, String[], String[])

Ensures that a task throws [MultiErrorException](#) asynchronously.

Declaration

```
public static async Task AssertThrowsMultiErrorExceptionAsync(Func<Task> funcTask, int errorCount, string[] errorCodes, string[] errorDescriptions)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.Threading.Tasks.Task>	funcTask	The task to execute.
System.Int32	errorCount	The expected error count.
System.String[]	errorCodes	The expected error codes.
System.String[]	errorDescriptions	The expected error descriptions.

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

AssertThrowsWithMessage<TException>(Action, String)

Throws exception if caught with message.

Declaration

```
public static void AssertThrowsWithMessage<TException>(Action action, string message)
    where TException : Exception
```

Parameters

TYPE	NAME	DESCRIPTION
System.Action	action	
System.String	message	

Type Parameters

NAME	DESCRIPTION
TException	

AssertThrowsWithMessageAsync<TException>(Func<Task>, String)

Throws async with message.

Declaration

```
public static async Task AssertThrowsWithMessageAsync<TException>(Func<Task> action, string message)
    where TException : Exception
```

Parameters

TYPE	NAME	DESCRIPTION
System.Func<System.Threading.Tasks.Task>	action	
System.String	message	

Returns

TYPE	DESCRIPTION
System.Threading.Tasks.Task	

Type Parameters

NAME	DESCRIPTION
TException	

AssertTodoAttribute(MemberInfo, String, Boolean)

Asserts [TodoAttribute](#) is applied on the specified member.

Declaration

```
public static void AssertTodoAttribute(MemberInfo member, string description, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	description	
System.Boolean	inherit	

AssertToDocumentationAttribute(MemberInfo, String, Boolean)

Asserts [WriteDocumentationAttribute](#) is applied on the specified member.

Declaration

```
public static void AssertToDocumentationAttribute(MemberInfo member, string description, bool inherit = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Reflection.MemberInfo	member	
System.String	description	
System.Boolean	inherit	

AssertTryResultError<TResult>(TryResultOnError<TResult>, String, String)

Asserts TryResultOnError<T> has an Error and Result is null.

Declaration

```
public static void AssertTryResultError<TResult>(TryResultOnError<TResult> result, string error, string errorDescription)
    where TResult : class
```

Parameters

TYPE	NAME	DESCRIPTION
TryResultOnError<TResult>	result	The result to check.
System.String	error	The expected error.
System.String	errorDescription	The expected error description.

Type Parameters

NAME	DESCRIPTION
TResult	

IsReleasePipelineConfig()

Declaration

```
public static bool IsReleasePipelineConfig()
```

Returns

TYPE	DESCRIPTION
System.Boolean	

Class TestLogger

The test logger provider.

Inheritance

System.Object
TestLogger

Inherited Members

System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.ToString()

Namespace: [PerpetualIntelligence.Test.Services](#)
Assembly: PerpetualIntelligence.Test.dll

Syntax

```
public static class TestLogger
```

Methods

Create<T>()

Creates a new Microsoft.Extensions.Logging.ILogger<TCategoryName>.

Declaration

```
public static ILogger<T> Create<T>()
```

Returns

TYPE	DESCRIPTION
Microsoft.Extensions.Logging.ILogger<T>	

Type Parameters

NAME	DESCRIPTION
T	