UnifiedOdds Feed SDK .NET library (.NET Standard 2.0)

UnifiedFeed SDK is a client library that enables easier integration with the Betradar XML feeds. SDK exposes XML feed service interface in a more user-friendly way and isolates the client from having to do XML feed parsing, proper connection handling, error recovery, event queuing, data caching and dispatching. It also makes a client solution more stable and robust when it comes to feed handling, especially with the release of new and updated XML feed versions.

For a quick start consider consulting the `DemoProject`, which can be found in GitHub releases.

The SDK uses the following 3rd party libraries:

- OpenTelemetry (1.6.0)
- Dawn.Guard (1.12.0)
- Humanizer (2.14.1)
- Microsoft.Extensions.Logging (7.0.0)
- Microsoft.Extensions.Caching.Memory (7.0.0)
- Microsoft.Extensions.Configuration (7.0.0)
- Microsoft.Extensions.Diagnostics.HealthChecks (7.0.11)
- Microsoft.Extensions.Http (7.0.0)
- RabbitMQ.Client (6.5.0)

UnifiedOdds Feed SDK .NET library (.NET Standard 2.0)

Notice: before starting `DemoProject` make sure to enter your bookmaker access token in app.config file and restore nuget packages by right-clicking the solution item and selecting "Restore NuGet Packages".

Below are basic examples that can help you start using sdk.

A basic way to use the UofSdk

To receive sdk events/messages subscribe to all `Sportradar.OddsFeed.SDK.Api.IUofSdk` and `Sportradar.OddsFeed.SDK.Api. IEntityDispatcher` events.

Note that there is one thread handling message reception and calling your event handler per session, so the processing within that method should be as quick as possible to not prevent following messages from being processed. It is recommended that all `Sportradar.OddsFeed.SDK.Entities.Rest.ISportEvent` processing is done in separate thread.

Below example is the minimum setup to start receiving messages. Note that you open only once, process messages for as long as you want, and then close the feed.

```
var config = UofSdk.GetConfigurationBuilder().BuildFromConfigFile();

var uofSdk = new UofSdk(config);

var session =
uofSdk.GetSessionBuilder().SetMessageInterest(MessageInterest.AllMessages).Build();

uofSdk.ProducerUp += OnProducerUp;
uofSdk.ProducerDown += OnProducerDown;
uofSdk.Disconnected += OnDisconnected;
uofSdk.Closed += OnClosed;

session.OnUnparsableMessageReceived += SessionOnUnparsableMessageReceived;
session.OnBetCancel += SessionOnBetCancel;
session.OnBetSettlement += SessionOnBetSettlement;
session.OnBetStop += SessionOnBetStop;
session.OnFixtureChange += SessionOnFixtureChange;
session.OnOddsChange += SessionOnOddsChange;
session.OnRollbackBetCancel += SessionOnRollbackBetCancel;
session.OnRollbackBetSettlement += SessionOnRollbackBetSettlement;
```

```
uofSdk.Open();
```

Namespace Sportradar.OddsFeed.SDK.Api

Classes

EntityDispatcherBase

A base class for classes used to dispatch messages

UofSdk

A IUofSdk implementation acting as an entry point to the odds feed SDK

UofSdkForReplay

A IUofSdk implementation acting as an entry point to the odds feed Replay Service for doing integration tests against played matches that are older than 48 hours

Interfaces

IEntityDispatcher<T>

Specifies a contract defining events used for user notification

IProducer

Defines a contract for producer which use the feed to dispatch messages

IRecoveryInfo

Defines a contract for recovery info which contains data about last recovery attempt

ISessionBuilder

Represents a second step when building a IUofSession instance

ISpecificEntityDispatcher<T>

Defines a contract implemented by classes capable of dispatching only specific entities

IUofSdk

Represent a root object of the unified odds sdk

IUofSdkForReplay

Represent a root object of the unified odds sdk when using Replay Server

IUofSession

Represents a session to the odds feed

IUofSessionBuilder

Represents a first step when building a IUofSession instance