Bookmaker SDK

Developer Guide

09.07.2015

sportradar
driven by facts

Table of contents

Sportradar AG
www.sportradar.com

P.O. Box 96
CH-9524 Zuzwil
Switzerland

P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com

VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

Getting started

Before starting to implement any of feeds supported by the SDK please read the appropriate feed documentation found on www.betradar.com under Help/Developer Zone/Downloads then check our README page and API documentation.

SDK Logging

SDK makes various logs during its lifetime. Logs are categorized by feed and based on whether these are critical alerts, invalid messages received, periodical queue stats dumps, configuration updates, message traffic, etc. Level of logging can be configured through each feed configuration element in the "Sdk" configuration section as shown in configuration example. We frequently use them for SDK support so it is recommended you send them along with your issue emails.

SDK logs are rotated periodically and stale logs get cleaned up periodically so that they are maintenance free.

Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

SDK Configuration

In your app.config / web.config you need at least to set-up the following:

```
<configSections>
  <section name="Sdk"
type="Sportradar.SDK.Services.SdkConfiguration.SdkConfigurationSection, BookmakerSdk"/>
</configSections>

<Sdk>
       <BetPal BookmakerId="xxx" BookmakerKey="yyy" />
       <LivePlex BookmakerId="xxx" BookmakerKey="yyy" />
       <SoccerRoulette BookmakerId="xxx" BookmakerKey="yyy" />
       <LiveOdds BookmakerId="xxx" BookmakerKey="yyy" />
       <LiveScout ScoutUsername="xxx" ScoutPassword="yyy" />
       <LiveOddsVbl BookmakerId="xxx" BookmakerKey="yyy" />
       <LiveOddsVfl BookmakerId="xxx" BookmakerKey="yyy" />
       <LiveOddsVhc BookmakerId="xxx" BookmakerKey="yyy"/>
       <LiveOddsVdr BookmakerId="xxx" BookmakerKey="yyy" />
       <LiveOddsVto BookmakerId="xxx" BookmakerKey="yyy" />
       <OddsCreator  Username="xxx" Password="yyy"/>
       <Lcoo Username="xxx" Password="yyy" FeedName="zzz"  />

</Sdk>
```

Note that configSection needs to be on top of the file and you need to provide the actual credentials. For more configuration options please refer to the documentation under Sportradar.SDK.Services.SdkConfiguration section.

Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

Start using Bookmaker SDK

Add reference to SDK project

In Visual Studio you first need to add a "Reference" to BookmakerSdk.dll in the project. The project should be built for .NET 4 framework. Intellisense uses the BookmakerSdk.xml file (located in the same directory as DLL).

SDK singleton

Bookmaker SDK is a singleton. There should be only one active SDK instance per process. If using multiple processes (e.g. failover setup) avoid running SDK instances in parallel, especially if the same access credentials are used. Otherwise you may end up having an inconsistent SDK error recovery state and run into problems due to server-side limits.

Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

SDK Setup and Teardown

Initialize the SDK

```
Sdk.Instance.Initialize();
```

Here you might want to pass a different configuration section name or pass on your implementation of the IDeadLetterQueue interface if you would like to log and/or report all invalid/undelivered SDK messages and events.

Start SDK instance and register for global SDK events

```
Sdk.Instance.OnQueueLimits += OnQueueLimit;
Sdk.Instance.Start();
```

Queue limit notifications are useful to give you alerts when message queue congestion starts to build up (or has settled down) so that you can be proactive about it.

Monitor SDK queue statistics

By looking at the Sdk.Instance.QueueStats you can see various performance characteristics and statistics of the underlying SDK dispatcher queues.

Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

Start SDK provider and register for provider events

Start SDK providers for those XML feeds that you intend to use (e.g. LiveOdds, LiveScout)

```
if (Sdk.Instance.LiveOdds != null)
{
    Sdk.Instance.LiveOdds.OnConnectionStable += ConnectionStable;

    Sdk.Instance.LiveOdds.OnAlive += AliveReceived;
    Sdk.Instance.LiveOdds.OnFeedError += FeedErrorOccurred;
    Sdk.Instance.LiveOdds.OnBetCancel += MatchBetCancelled;
    Sdk.Instance.LiveOdds.OnBetCancelUndone += MatchBetCancelUndone;
    Sdk.Instance.LiveOdds.OnBetClear += MatchBetCleared;
    Sdk.Instance.LiveOdds.OnBetClearRollback += MatchBetClearRollbacked;
    Sdk.Instance.LiveOdds.OnBetStart += BetStarted;
    Sdk.Instance.LiveOdds.OnBetStop += BetStopped;
    Sdk.Instance.LiveOdds.OnMetaInfo += MetaInfoReceived;
    Sdk.Instance.LiveOdds.OnOddsChange += OddsChanged;
    Sdk.Instance.LiveOdds.OnScoreCard += ScoreCardReceived;

    Sdk.Instance.LiveOdds.Start();
}
else
{
    throw new ApplicationException("Error initializing SDK.LiveOdds provider");
}
```

If you don't call Sdk.Instance.Initialize() accessing Sdk.Instance.LiveOdds will throw an InitException.

SDK provider(s) will try to connect to the corresponding XML feed server and keep the connection alive. If the connection is lost the provider will try to reconnect automatically and you will be informed of this through corresponding events.

Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

Stop SDK providers and unregister from provider events

In the finally block you can stop all SDK providers and unregister from provider events.

```
if (Sdk.Instance.LiveOdds != null)
{
    Sdk.Instance.LiveOdds.Stop();

    Sdk.Instance.LiveOdds.OnConnectionStable -= ConnectionStable;

    Sdk.Instance.LiveOdds.OnAlive -= AliveReceived;
    Sdk.Instance.LiveOdds.OnFeedError -= FeedErrorOccurred;
    Sdk.Instance.LiveOdds.OnBetCancel -= MatchBetCancelled;
    Sdk.Instance.LiveOdds.OnBetCancelUndone -= MatchBetCancelUndone;
    Sdk.Instance.LiveOdds.OnBetClear -= MatchBetCleared;
    Sdk.Instance.LiveOdds.OnBetClearRollback -= MatchBetClearRollbacked;
    Sdk.Instance.LiveOdds.OnBetStart -= BetStarted;
    Sdk.Instance.LiveOdds.OnBetStop -= BetStopped;
    Sdk.Instance.LiveOdds.OnMetaInfo -= MetaInfoReceived;
    Sdk.Instance.LiveOdds.OnOddsChange -= OddsChanged;
    Sdk.Instance.LiveOdds.OnScoreCard -= ScoreCardReceived;
}
```

Stop the SDK instance and unregister from global SDK events

After all SDK providers have been stopped it is time to shut down the main SDK instance.

```
Sdk.Instance.OnQueueLimits -= OnQueueLimit;
Sdk.Instance.Stop();
```


Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

Tips and Tricks

Processing of SDK events

SDK triggers various events for all the feeds within SDK dispatcher threads. There are 3 dispatcher threads by default. The dispatching is made sure to be in order for every event. That is why you should make heavy processing of events in your own threads and use SDK threads only to pass the event data to your own threads. So if any of your processing takes more than a few milliseconds pass the data to your own threads or else the SDK queue will start piling up which can lead to various problems.

Bet Start

You will receive “bet start” events immediately as they get transmitted over XML feeds. There is a special case after LiveOdds error recovery is finished and the connection is stable again. In this case SDK does NOT generate explicit trailing "bet start" events (there was an explicit leading “bet stop” though).

Besides handling the OnBetStart events properly you should also check MatchState.BetStatus provided with all SDK match related events (e.g. OnOddsChange) and start accepting bets based on this as well.

Bet stop

SDK generates explicit "bet stop" message if a client gets disconnected from XML feeds and also performs automatic error recovery on behalf of the client.

Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

Bet clearing

SDK does NOT keep track of bet clearings. If you get disconnected for a longer period of time it may happen that after you come back a match is already over. In that time-frame bets are not accepted as artificial bet stops were dispatched (so you are safe) but it might still be necessary to clear all bets placed at the beginning of the match. In this case you should properly handle removed events and invoke the GetEventStatus method to obtain the most up-to-date match status and also all bet clearings for that match.

In any case it is wise to handle stale matches by defining a grace period (TTL) after which you can do match related cleanup and housekeeping.

Match suspended or cancelled

If a match is suspended or cancelled you will receive OnMetaInfo event.

Current match status will be reflected in the periodical OnAlive event (see AliveEventArgs.Alive.EventHeaders[x].Status). Again if you get disconnected for a longer period of time and miss that match suspended/cancelled event you should explicitly invoke the GetEventStatus to get the final match status.

Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0

Life Cycle of Odds setup

SDK uses HTTP pull method of obtaining data for LCoO feed. So the Sportradar sales person / support should provide you with a link like so:

https://www.betradar.com/betradar/getXmlFeed.php?bookmakerName=username&key=password&xmlFeedName=feed_name

Then the SDK config should be:

```
<Sdk>
    <Lcoo Username="username" Password="password" FeedName="feed_name"/>
</Sdk>
```


Sportradar AG
www.sportradar.com
P.O. Box 96
CH-9524 Zuzwil
Switzerland
P: +41 71 544 46 56
F: +41 71 544 46 57
E: sales@sportradar.com
VAT Code: 686 261
Identification No:
CH-320.3.062.151-0