

Bookmaker Java SDK

Developer Guide

09.07.2015

Table of contents

Table of contents	2
SDK Interface documentation	3
SDK Logging	3
SDK Configuration.....	3
Start using Bookmaker SDK.....	5
Add reference to SDK project	5
SDK singleton	5
SDK Example Setup and Teardown.....	6
Tips and Tricks	10
Bet stop.....	10
Bet clearing	10
Match suspended or cancelled	11
LiveOdds test server	11

SDK Interface documentation

Please consult the latest Bookmaker SDK interface documentation on SDK website: <http://sdk.sportradar.com/javasdk/javadoc/index.html>.

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 properties as shown in configuration example.

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

SDK Configuration

You can configure SDK over sdk.properties file or you call [*init\(Properties\)*](#). All the SDK properties can be found at <http://sdk.sportradar.com/javasdk/sdk.properties.txt>.

Most of the settings have default values which you rarely need to override.

Here is an example of sdk.properties file with basic feeds enabled. Most LiveOdds based feeds (VFL, VHC, ...) have the same possible settings.

LiveOdds configuration example

```
sdk.liveodds.enabled= true  
sdk.liveodds.bookmaker_id=id  
sdk.liveodds.bookmaker_key=key
```

LiveScout configuration example

```
sdk.livescout.enabled= true  
sdk.livescout.username=scout_username  
sdk.livescout.password=scout_password
```

OddsCreator configuration example

```
sdk.oddscreator.enabled= true  
sdk.oddscreator.username=oddscreator_username  
sdk.oddscreator.password= oddscreator_password
```

LCoO configuration example

Note you need HTTP account and you get a link like so

```
#https://www.betradar.com/betradar/getXmlFeed.php?bookmakerName=username&key=password  
# &xmlFeedName=feedname&deleteAfterTransfer=no
```

```
sdk.lcoo.enabled=true  
sdk.lcoo.username=username  
sdk.lcoo.password= password  
sdk.lcoo.feed_name= feedname
```

Start using Bookmaker SDK

Add reference to SDK project

You need to add SDK jar file to the project which can be obtained on our website.

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.

SDK Example Setup and Teardown

```
final Sdk sdk = Sdk.getInstance();
final LiveOddsFeed liveOddsFeed = sdk.getLiveOdds();
final LiveOddsFeed liveOddsVblFeed = sdk.getLiveOddsVbl();
final LiveOddsFeed liveOddsVf1Feed = sdk.getLiveOddsVf1();
final LiveOddsFeed liveOddsVhcFeed = sdk.getLiveOddsVhc();
final LiveOddsFeed liveOddsVdrFeed= sdk.getLiveOddsVdr();
final LiveOddsFeed liveOddsBetpal = sdk.getLiveOddsBetpal();
final LiveOddsFeed liveOddsLivePlex = sdk.getLiveOddsLivePlex();
final LiveOddsFeed liveOddsSoccerRouletteFeed = sdk.getLiveOddsSoccerRoulette();
final LiveScoutFeed liveScoutFeed = sdk.getLiveScout();
final LcooFeed lcooFeed = sdk.getLcoo();
final OddsCreatorFeed oddsCreatorFeed = sdk.getOddsCreator();

final LiveOddsFeedListener oddsMatchFeedListener = new LiveOddsMatchFeedListenerImpl();

final LiveOddsFeedListener oddsRaceFeedListener = new LiveOddsRaceFeedListenerImpl();

final LiveScoutFeedListener scoutFeedListener = new LiveScoutFeedListenerImpl();

final LcooFeedListener lcooFeedListener = new LcooFeedListenerImpl();
```

```
if (liveScoutFeed != null) {  
    liveScoutFeed.open(scoutFeedListener);  
}  
  
if (liveOddsFeed != null) {  
    liveOddsFeed.open(oddsMatchFeedListener);  
}  
  
if (liveOddsVf1Feed != null) {  
    liveOddsVf1Feed.open(oddsMatchFeedListener);  
}  
  
if (liveOddsVb1Feed != null) {  
    liveOddsVf1Feed.open(oddsMatchFeedListener);  
}  
  
if (liveOddsVhcFeed != null) {  
    liveOddsVhcFeed.open(oddsRaceFeedListener);  
}  
  
if (liveOddsVdrFeed != null) {  
    liveOddsVdrFeed.open(oddsRaceFeedListener);  
}  
  
if (liveOddsBetpal != null) {  
    liveOddsBetpal.open(oddsMatchFeedListener);  
}  
  
if (liveOddsLivePlex != null) {  
    liveOddsLivePlex.open(oddsMatchFeedListener);  
}
```

```
if (liveOddsSoccerRouletteFeed != null) {
    liveOddsSoccerRouletteFeed.open(oddsMatchFeedListener);
}

if (lcooFeed != null) {
    lcooFeed.open(lcooFeedListener);
}

if (oddsCreatorFeed != null) {
    try {
        System.out.println("Listing all sports :");
        for(IdNameEntity sports : oddsCreatorFeed.getSports()){
            System.out.println(sports.getName());
        }
    } catch (OddsCreatorException e) {
        e.printStackTrace();
    }
}
```

```
logger.info("The sdk is running. Hit any key to exit");

BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
try {
    reader.readLine();
} catch (IOException e) {
    e.printStackTrace();
}

logger.info("Closing the sdk");
sdk.close();
logger.info("Sdk successfully closed. Main thread will now exit");
```

You should implement your own feed listeners. You can find example project on our website which has basic implementation of feed listeners. As you can see all LiveOdds orientated feeds can be implemented with Match based listener and Race based listener.

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.

Tips and Tricks

Processing of SDK events

SDK triggers various events for all the feeds within SDK dispatcher threads. There are 3 dispatcher threads per feed 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 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. If the "bet stop" is SDK generated `isArtificial()` will return true;

EventHeaderEntity

EventHeaderEntity is common for both Race and Match oriented LiveOdds feeds. So you can cast it to MatchHeaderEntity or RaceHeaderEntity.

MetaInfoDataContainer

Similar logic applies to MetaInfoEntity which has `getMetaInfoDataContainer()` method which returns MetaInfoDataContainer which you can cast to LiveOddsMetaData or RaceMetaData.

Bet clearing

SDK does NOT keep track of bet clearings. SDK keeps all events in Alive in sync but if you get disconnected for a longer period of time it may happen that after you come back a match is already over and not in Alive message anymore. In that time-frame bets are not accepted (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 `onAliveReceived` event (see `EventHeaderEntity.getStatus()`). 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.

LiveOdds test server

To connect to LiveOdds test server all you need to specify is `sdk.liveodds.test = true`.

`LiveOddsFeed` has `getTestManager` method which returns `TestManager` which you can cast to `LiveOddsTestManager` which you can use to start tests as described in LiveOdds PDF file.