UOF - SDK

betradar
driven by facts

The Unified Odds SDK (Software development kit) provides a simple and efficient way for bookmakers to access Betradar's odds and sport information. It combines subscription of messages and API calls into a unified Java or .NET interface that hides most of the complexity, including recovery.

Our SDK has a dedicated documentation section both for the Java and .NET versions. For more detailed information about using and getting started with the SDK, please visit our dedication SDK sections:

Java: http://sdk.sportradar.com/content/unifiedfeedsdk/java2/javadoc/

.NET: http://sdk.sportradar.com/content/unifiedfeedsdk/net/doc/html/15adaea0-5ed9-0831-79a5-f2702175b2d9.htm

SDK benefits over protocol/API

- The SDK hides the separation between messages and API-lookups. The client system just receives the message objects where all information is prefilled by the SDK caches, and in some cases looked up in the background when the client system requests some more rarely requested information.
- The SDK takes care of translations transparently. The client system just needs to define what languages it needs.
- The SDK takes care of dynamic text markets and outright markets automatically, which requires some extra logic and lookups for someone not using the SDK.
- The SDK handles initial connect and state, as well as recovery in case of a temporary disconnect. This needs to be handled manually by someone not using the SDK.
- The SDK provides an up to date cache of each sport-events current status that is updated automatically.

Remember that your OddsFeedListener call-backs are handled by a single thread; if you do too much processing within your call-back, subsequent odds messages are delayed as they will not be processed until you finish the call-back processing.

The recommended way to handle longer processing times is to put the received messages on a queue, and let them be processed by a different thread.

The first time you try to access information about a sport event, the SDK looks up its details using an API-call. This information is cached for subsequent calls. For that reason, the first call for a new sport event may take a bit longer. It is also during this first time call that the various languages you have requested are looked up for the event – so if you have specified multiple languages the first call may take even longer.

If you can avoid accessing any non-basic properties, you can avoid longer delays most of the time, as the basic information is normally cached for almost all messages by the SDK. (Basic information is sport, category, tournament, match competitors, scheduled start time).

Back to top