UOF Java SDK - Migration Guide

The UOF Java SDK 2.x.x is upgraded to 3.0.0 and this is your roadmap to a smooth transition from your current SDK version to the latest version. The upgrade is designed to elevate your experience and align the SDK more closely with your business needs.

This guide is intended to offer practical advice to ensure your transition is not only efficient but also enhances the performance and capabilities of your software.

Here are the general steps you can follow to perform this upgrade:

Page Contents

1. Upgrade Dependencies

You should upgrade `` `com.sportradar.unifiedodds.sdk.unified-feed-sdk` `` maven dependency to **3.0.0**. Before performing the upgrade, it's advisable to consider upgrading certain libraries your project depends on.

The UOF SDK's dependencies have been upgraded, and the revised configuration was tested in Java 8, 11 and 17.

The following libraries are effected by this upgrade.

Upgraded libraries	From	To
apache http client	*4.5.1*	*5.2.1*
rabbitmq.amqp-client	*3.6.0*	*5.17.0*
slf4j	*1.7.13*	*2.0.7*
guava	*21.0*	*31.1-jre*
guice	*4.0*	*5.1.0*
icu4j	*57.1*	*72.1*
snakeyaml	*1.20*	*2.0*
jaxb api	*2.3.0*	*2.3.3*
jaxb runtime changed	*sun.xml.bind 2.3.0*	*org.glassfish.jaxb 2.3.8*
lombok	*1.18.24*	*Removed*
opentelemetry-api	-	*1.25.0 (new addition)*

2. Build new UofSdk instance

The root class is renamed from OddsFeed to UofSdk.

You will need to build configuration before creating new UofSdk instance.

```
UofConfiguration config = UofSdk.getUofConfigurationBuilder().buildConfigFromSdkProperties();
UofSdk uofSdk = new UofSdk(new GlobalEventsListener(), config);
```

3. Update the methods and classes in your code

Review your codebase to identify any parts that might be affected by the upgrade. Look for deprecated methods or classes that have been removed in the new version. Update your code to use the new APIs provided by the UOF Java SDK 3.0.0. This may involve making changes to method calls, imports, and class references. Handle any breaking changes or deprecations by updating your code accordingly. You can contact support if you encounter specific issues.

The following classes and methods are changed. Hence, you will be needed to update your code to use new names.

Root Classes Renamed

- `OddsFeedConfiguration to UofConfiguration`
- `GlobalEventListener to UofGlobalEventListener`
- `OddsFeedException to UofException`
- `OddsFeedConfigurationBuilder to  UofConfigurationBuilder`
- `getOddsFeedConfigurationBuilder to getUofConfigurationBuilder`
- `OddsFeedListener to UofListener`
- `OddsFeedSessionBuilder to UofSessionBuilder`
- `OddsFeedSession to UofSession`
- `OddsFeed to UofSdk`
- `SportsInfoManager to SportDataProvider`
- `CustomisableSDKModule to CustomisableSdkModule`
- `SDKGlobalEventListener to SdkGlobalEventListener`
- `SDKProducerListener to SdkProducerListener`
- `SDKConnectionStatusListener to SdkConnectionStatusListener`
- `SDKEventRecoveryStatusListener to SdkEventRecoveryStatusListener`
- `OddsFeed to UofSdk`
- `ReplayOddsFeed to UofSdkForReplay`
- `OddsFeedExt to UofSdkExt`
- `CustomisableOddsFeed to CustomisableUofSdk`
- `ReplayFeed to UofSdkForReplay`
- `Removed OperationManager (properties moved to UofConfiguration)`
- `EnvironmentSelector - removed SelectIntegration() and SelectProduction() - use SelectEnvironment (SdkEnvironment ufEnvironment)`
- `configuration related classes moved to com.sportradar.unifiedodds.sdk.cfg package`

Removed methods and classes

Removed Methods / Classes	Alternative Methods / Classes
`ConfigurationAccessTokenSetter`	`TokenSetter`
`OddsFeedConfigurationBuilder`	-
`selectIntegration` and `selectProduction` methods from EnvironmentSelector	`selectEnvironment`
`OddsFeedSessionBuilder.setSpecificListeners`	-
`onProducerDown(ProducerDown / producerDown)` and `onProducerUp (ProducerUp / producerUp)` has been removed from SdkProducerStatusListener	The producer status messages are now being received through o `nProducerStatusChange (ProducerStatus)` only
`Environment.Staging`	use `Environment.Integration` instead
`EventClock.getRemainingDate`	use `EventClock.getRemainingTime` instead
`EventsStatus.getApiId`	Use `EventStatus.getApiName` instead

`MarketDescription.getIncludesOutcomeOfTypes`	use `MarketDescription.getOutcomeType` instead.
`PeriodStatistics.getTeamStatisticDtos`	use `PeriodStatistics.getTeamStatistics` instead
`OutcomeOdds.getOdds()`	use `OutcomeOdds.getOdds(OddsDisplayType)` instead To get the same value from `getOdds(OddsDisplayType)` you have to use `outcomeOdds.getOdds(OddsDisplayType.Decimal)`
`OutcomeSettlement.isWinning`	use `OutcomeSettlement.getOutcomeResult == OutcomeResult.Won`
`Message.getTimestamp`	use `Message.getTimestamps.getCreated` instead

Added methods / New Methods

- `categoryData.getName`
- `sportData.getName`
- `currentSeasonInfo.getNames`
- `drawResult.getNames`
- `manager.getNames`
- tournamentInfo.getName

Removed methods

- round.getGroupName

corresponding method removals and constructor argument adjustments were applied on exportableCompleteRoundCi and exportableLoadableRoundCi classes.

Restructured methods

- venue.getCourses signature was changed from returning List<Hole> to List<Course> to accommodate multiple courses.
  - Course entity is introduced and contains the following methods:
    - List<Hole> getHoles
    - Urn getId
    - getName and getNames consistent with these method semantics elsewhere in the SDK having same signatures.
  - Corresponding changes were made to ExportableVenueCi class along with introduction of ExportableCourseCi class.
- teamCompetitor.getDivision method was moved up in the inheritance hierarchy to competitor and its signature was changed from returning Integer to Division in order for Decision to also accommodate division name
  - Division entity is introduced containing:
    - Integer getDivision
    - String getDivisionName
  - Corresponding changes were made to ExportableCompetitorCi class along with introduction of ExportableDivisionCi class.
- timelineEvent.getHomeScore and timelineEvent.getAwayScore signatures were changed from returning double to BigDecimal to make these methods return types consistent with other occurrences of score related methods elsewhere in the SDK.
  - Corresponding changes were made to ExportableTimelineEventCi
- change of behaviour of round.getName, round.getNames and round.getPhaseOrGroupLongName
  - round.getName and round.getNames methods returns the value of "name" property from the Sports API or empty string in case API does not provide it (or follows the SDK convention for error handling in case API call failing), and no longer falls back to "group_name" and "group_long_name" attributes.
  - round.getPhaseOrGroupLongName method returns the value of "group_long_name" property from the Sports API or empty string in case API does not provide it (or follows the SDK convention for error handling in case API call failing) no longer falls back to "name" and "name" attributes.

Name changes

The following are changed to improve the consistency.

- URN Urn
- Classes within the package com.sportradar.unifiedodds.sdk.caching.exportable
  used to have CI suffix like ExportableCI or ExportableSportEventCI
  but this suffix is renamed to Ci like ExportableCi or ExportableSportEventCi.
  The package has dozens of classes, but the change was performed to all classes in a uniform fashion.
- All classes within *com.sportradar.uf.datamodel* used to start with a preffix "UF" like UFReportingStatus and UFEventStatus,
  but the prefix is renamed to "Uf" like UfReportingStatus and UfEventStatus. This package contains hundreds of classes,
  but the convention was changed for all of them in an identical fashion.

4. Update the Configuration

The configuration settings were split between configuration class and OperationManager. OperationManager is removed and all settings are consolidated within the **`UofConfiguration`** interface.

You have three choices for constructing the final configuration through:

1. properties file or
2. YAML file or
3. programmatically using ConfigurationBuilder.

Certain options that were previously available in properties and YAML files have been removed and can exclusively be configured using the configuration builder.

Through Properties / YAML file

You'll need to re-configure the following either in the properties file or YAML.

Keys		Comments
accessToken	required	The token you are currently using can also be used with our new version.
defaultLanguage or desiredLanguages	required	desiredLanguages is renamed from desiredLocales.
nodeId	optional	recommended to be set – must be unique per sdk instance
environment	optional	if not set, '*Integration*' will be used. Renamed from ufEnvironment.
inactivitySeconds	-	This is removed from properties and yaml files. However, you can set this through configuration builder.
messagingHost	optional	sets the URL of the messaging host (broker) i.e.mq.betradar.com. This setting is used only when using custom environment.
messagingUseSsl	optional	sets the value indicating whether a secure connection to the message broker should be used. This setting is used only when using custom environment. This is renamed from useMessagingSsl.
messagingVirtualHost	optional	sets the name of the virtual host configured on the messaging server (broker). This setting is used only when using custom environment.
apiHost	optional	sets the URL of the API host (https://api.betradar.com). This setting is used only when using custom environment.
apiUseSsl	optional	sets the value indicating whether a secure connection to the Sports API should be used. This setting is used only when using Custom environment. This is renamed from useApiSsl.
exceptionHandlingStrategy	optional	sets a ExceptionHandlingStrategy enum member specifying how to handle exceptions thrown to outside callers (‘Catch’ or ‘Throw’)
disabledProducers	optional	sets the comma delimited list of ids of disabled producers (e.g. 1,2,7,9)
maxRecoveryTime	optional	This is removed from properties and yaml files. However, you can set this through configuration builder
adjustAfterAge	optional	This is removed from properties and yaml files. However, you can set this through configuration builder
httpClientTimeout	optional	This is removed from properties and yaml files. However, you can set this through configuration builder
recoveryHttpClientTimeout	optional	This is removed from properties and yaml files. However, you can set this through configuration builder

Through ConfigurationBuilder

The full configuration can be also setup via ConfigurationBuilder obtained via `uofSdk.getUofConfigurationBuilder()` . The resulting UofConfiguration contains all the previously set configurations for the SDK.

5. Test your project

Thoroughly test your project after making the changes. Test all critical functionality to ensure that everything still works as expected. Pay special attention to any areas of your setup that interact with the UOF Java SDK, as these are likely to be the most affected by the upgrade.

6. Update the Documentation

Update your project's documentation and any training materials to reflect the changes introduced by the upgrade. This will help your team members understand and work with the new version.

7. Deploy to production

Once you are confident that your project works correctly with the upgraded UOF Java SDK, you can deploy the updated version to your production environment.

8. Monitoring and Maintenance

After deployment, monitor your project closely for any unexpected issues or performance problems. Be prepared to address any post-upgrade issues promptly.

9. Feedback and Reporting

If you encounter any bugs or issues in the UOF Java SDK 3.0.0, consider reporting them to support@sportradar.com. Providing feedback can help improve the SDK for future releases.