

UOF .Net SDK - Migration Guide



The **Sportradar.OddsFeed.SDKCore** library target framework was downgraded from .NET Standard 2.1 to .NET Standard 2.0, so this package can be used in any .NET Core and .NET project and also in old .NET Framework 4.6.2 or newer. No upgrades are planned for the package **Sportradar.OddsFeed.SDK** in the future and solutions should be upgraded to use **Sportradar.OddsFeed.SDKCore** 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.

Page Contents

1. Upgrade Dependencies
2. Build new UofSdk instance
3. Update the methods and classes in your code
 - Root Classes Renamed
 - Removed methods and classes
 - Added or changed methods
 - Enum values renamed to CamelCase
 - Changes and/or new features
 - Name changes
 - Changed namespaces
4. Update the Configuration
 - Upgrade configuration in App.config
 - Other changes
 - Through ConfigurationBuilder
5. Test your project
6. Update the Documentation
7. Deploy to production
8. Monitoring and Maintenance
9. Feedback and Reporting

Here are the general steps you can follow to complete the transition:

1. Upgrade Dependencies

Upgrade **Sportradar.OddsFeed.SDKCore** Nuget package to 2.0.0. Before upgrading you might need to upgrade some of the dependent libraries, like libraries for logging or dependency injection.

SDK dependent libraries were upgraded or replaced.

Removed Libraries

<i>Newtonsoft.Json</i>
<i>Castle.Core</i>

Libraries Added

<i>Microsoft.Extensions.Diagnostics.HealthChecks</i>	7.0.11
<i>Microsoft.Extensions.Http</i>	7.0.0

Libraries Upgraded / Replaced

	From	To
<i>Humanizer</i>	2.8.26	2.14.1
<i>RabbitMQ.Client</i>	5.1.2	6.5.0
<i>Microsoft.Extensions.Logging.Abstractions</i>	3.1.0	7.0.0
<i>System.Configuration.ConfigurationManager</i>	4.7.0	7.0.0

Replaced	with
<i>App.Metrics</i>	<i>OpenTelemetry 1.6.0</i>
<i>System.Runtime.Caching</i>	<i>Microsoft.Extensions.Caching.Memory 7.0.0</i>

2. Build new UofSdk instance

The feed instance is now named UofSdk. Building new feed instance is changed. You need to build configuration and register all the sdk services before creating new UofSdk instance.

```
var uofConfiguration = UofSdk.GetConfigurationBuilder().BuildFromConfigFile();
var services = new ServiceCollection();
services.AddUofSdk(uofConfiguration);
var uofSdk = new UofSdk(services.BuildServiceProvider());
```

Note



When configuring WebHost services (as in ASP.NET Core apps) via WebApplicationBuilder you should create scope before creating UofSdk. Registering services for logging, telemetry and user classes is omitted for simplicity (but dont forget to add them).

```
var webApplicationBuilder = WebApplication.CreateBuilder(args).WebHost.ConfigureServices(
    (hostBuilderContext, serviceCollection) => {
        serviceCollection.AddUofSdk(configuration);
    });

var app = webApplicationBuilder.Build();

var uofSdk = new UofSdk(app.Services.CreateScope().ServiceProvider);
```

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

- IOddsFeed to IUofSdk
- IOddsFeedSession to IUofSession
- Feed to UofSdk
- ReplayFeed to UofSdkForReplay
- IOddsFeedExt to IUofSdkExtended
- FeedExt to UofSdkExtended
- IOddsFeedConfigurationSection to IUofConfigurationSection
- OddsFeedConfigurationSection to UofConfigurationSection
- IOddsFeedConfiguration to IUofConfiguration
- Removed OperationManager (properties moved to IUofConfiguration)
- IEnvironmentSelector - removed SelectIntegration() and SelectProduction() - use SelectEnvironment (SdkEnvironment ufEnvironment)
- Renamed Feed.CreateBuilder() to UofSdk.GetSessionBuilder() for creating new IUofSession
- config section moved to Sportradar.OddsFeed.SDK.Api.Internal.Config.UofConfigurationSection

Removed methods and classes

- IOOutcomeSettlement.Result
- IOddsFeedConfigurationSection.UseIntegrationEnvironment
- IRound.GroupName

- `IRound.GetGroupName()`

Added or changed methods

- Added support for `IVenue.Courses` (returns list of `ICourse` instead of list of `IHole`)

Enum values renamed to CamelCase

- `MessageType`
- `ExceptionHandlingStrategy`
- `CashoutStatus`
- `FixtureChangeType`
- `MarketStatus`
- `OddsChangeReason`
- `PropertyUsage`
- `ResourceTypeGroup`

Changes and/or new features

- Added support for `IVenue.Courses` (returns list of `ICourse` instead of list of `IHole`)
- Added support for `ICompetitor.Division` - now contains division id and name (moved and replaced from `ITeamCompetitor`)
- Extended `IJersey` with `SquareColor` and `HorizontalStripesColor`

Name changes

The following are changed to improve the consistency. Some classes were also moved to different namespace:

Renamed	to
URN	Urn
ReplayPlayerStatus.Setting_up	SettingUp
IFixture.StartTimeTBD	StartTimeTbd
EventStatus.Not_Started	NotStarted
FeedMessage.EventURN	EventUrn
IRound.Name	Names
IRound.PhaseOrGroupLongName	PhaseOrGroupLongNames
IProducerManager.Get()	GetProducer()

Changed namespaces

	renamed to
API namespace	Api
REST namespace	Rest

	moved to
replay interfaces	Api.Replay
feed managers	Api.Managers
feed providers	Api.Managers

enum types	Common.Enums
IUofConfigurationSection	Api.Internal.Config
configuration interfaces	Api.Config
MessageInterest class	Api.Config

4. Update the Configuration

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

You have two ways for constructing to build final configuration:

1. combining App.config and IConfigurationBuilder or
2. just programmatically via IConfigurationBuilder.

Some of the options were removed from App.config section options and can only be configured via configuration builder.

Upgrade configuration in App.config

Replace

```
<configSections>
<section name="oddsFeedSection" type="Sportradar.OddsFeed.SDK.Api.Internal.OddsFeedConfigurationSection,
Sportradar.OddsFeed.SDK" />
</configSections>
```

with

```
<configSections>
<section name="uofSdkSection" type="Sportradar.OddsFeed.SDK.Api.Internal.Config.UofConfigurationSection,
Sportradar.OddsFeed.SDK" />
</configSections>
```

Other changes

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

Keys		Comments
<i>accessToken</i>	required	The token you are currently using can also be used with our new version.
<i>defaultLanguage or desiredLanguages</i>	required	<i>Note: desiredLanguages is renamed from supportedLanguages</i>
<i>nodeld</i>	optional	Recommended to be set – must be unique per sdk instance
<i>environment</i>	optional	if not set, 'Integration' will be used. <i>Note: Renamed from ufEnvironment.</i>
<i>supportedLanguages</i>		This is removed. Use desiredLanguages.
<i>inactivitySeconds</i>	-	This is removed from App.config. However, you can set this through configuration builder.
<i>host</i>	optional	Sets the URL of the messaging host/broker i.e. mq.betradar.com . This setting is used only when using Custom environment.
<i>useSsl</i>	optional	Sets the value indicating whether a secure connection to the message broker should be used (true or false). This setting is used only when using Custom environment. <i>Note: Renamed from useSSL.</i>
<i>virtualHost</i>	optional	Sets the name of the virtual host configured on the messaging server (broker). This setting is used only when using Custom environment.
<i>apiHost</i>	optional	sets the URL of the API host (https://api.betradar.com). This setting is used only when using custom environment.

<i>apiUseSsl</i>	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. <i>Note: This is renamed from useApiSSL.</i>
<i>exceptionHandlingStrategy</i>	optional	Sets a ExceptionHandlingStrategy enum member specifying how to handle exceptions thrown to outside callers ('Catch' or 'Throw').
<i>disabledProducers</i>	optional	Sets the comma delimited list of ids of disabled producers (e.g. '1,2,7,9').
<i>maxRecoveryTime</i>	optional	This is removed from App.config. However, you can set this through configuration builder.
<i>adjustAfterAge</i>	optional	This is removed from App.config. However, you can set this through configuration builder.
<i>httpClientTimeout</i>	optional	This is removed from App.config. However, you can set this through configuration builder.
<i>recoveryHttpClientTimeout</i>	optional	This is removed from App.config. 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 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 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 NetStandard SDK 2.0.0, consider reporting them to support@sportradar.com. Providing feedback can help improve the SDK for future releases.