Interface with any LRS from your Xamarin Mobile or .NET Apps

The xAPIWrapper component and the xAPI.NET Standard Library enables capturing of xAPI/TINCAN statements and other artifacts that can to pushed to any standard's compliant LRS. It has two main libraries in the component that can be used independently.

- TINCAN Standard .NET Library
- xAPIWrapper, wrapper and simplified interface

What is xAPI/TINCAN?

- For information on xAPI/TINCAN refer to this site ADL
- For technical documentation refer to this site TINCAN API
- For information on Xamarin refer to this site Xamarin

With easy-to-use SDKs for Android and iOS, helpful classes and methods to track user activities, as TINCAN statements, activities or scores and other interacations in a simple Actor, Object, Verb verbatim.

Quick Start Notes:

1. .NET Standard 1.6 compatible library
2. An xAPIWrapper port implemented in C# that can be used to interface with any LRS
3. The library can be used with platforms that supports the .NET Standard
4. The xAPIWrapper eliminates the need for client/end user to know the internals of the TINCAN specification
5. The xAPIWrapper implementation makes generating statements as easy as adding log statements to your code

Available on

![icons] (../icon_set.png)

Installation and Usage

Import component in your Android and iOS apps.

For Android

Brief: Following code can be used to send a xAPI statement to an LRS

```
button.Click += async delegate
{
    var apiWrapper = new APIWrapper("xAPI Endpoint", "Your Identity", "Your
Secret");
    var statement = apiWrapper.PrepareStatement("test@ald.net", "experienced",
"Activity");
```

```
    var task = await apiWrapper.SendStatement(statement);
    if (task.Success)
    {
        button.Text = $"{_count++} sent!";
    }
};
```

For iOS

Brief: Following code can be used to send a xAPI statement to an LRS

```
Button.TouchUpInside += async delegate
{
   var apiWrapper = new APIWrapper("xAPI Endpoint", "Your Identity", "Your Secret");
   var statement = apiWrapper.PrepareStatement("test@ald.net", "experienced",
"Activity");
   var task = await apiWrapper.SendStatement(statement);
   var title = $"sending {_count} statement!";
   if (task.Success)
   {
      title = $"{_count++} statements sent!";
   }
   Button.SetTitle(title, UIControlState.Normal);
};
```