**Features of Mikros Unity SDK**

**Mikros Analytics**

Once the SDK is initialized, you can immediately start logging events (preset or custom) for analytics.

Make sure to have the following namespaces defined at the top of your scripts:

using MikrosClient;

using MikrosClient.Analytics;

**Custom Analytics Events**

Custom events can be tracked by the following ways:

1. To log an event without any parameters

MikrosManager.Instance.AnalyticsController.LogEvent("custom\_event\_name", (Hashtable customEventWholeData) =>

{

// handle success

},

onFailure =>

{

// handle failure

});

1. To log an event with only one parameter of String datatype

MikrosManager.Instance.AnalyticsController.LogEvent("custom\_event\_name", "parameter", "parameter\_value", (Hashtable customEventWholeData) =>

{

// handle success

},

onFailure =>

{

// handle failure

});

1. To log an event with only one parameter of Double datatype

MikrosManager.Instance.AnalyticsController.LogEvent("custom\_event\_name", "parameter", 1.5, (Hashtable customEventWholeData) =>

{

// handle success

},

onFailure =>

{

// handle failure

});

1. To log an event with only one parameter of Long datatype

MikrosManager.Instance.AnalyticsController.LogEvent("custom\_event\_name", "parameter", 1, (Hashtable customEventWholeData) =>

{

// handle success

},

onFailure =>

{

// handle failure

});

1. To log an event with multiple parameters of any datatype

Hashtable parameterData = new Hashtable();

parameterData.Add("parameter1", "parameter\_value");

parameterData.Add("parameter2", 1.5);

parameterData.Add("parameter3", 1);

parameterData.Add("parameter4", true);

MikrosManager.Instance.AnalyticsController.LogEvent("custom\_event\_name", parameterData, (Hashtable customEventWholeData) =>

{

// handle success

},

onFailure =>

{

// handle failure

});

All custom events are uploaded automatically to backend at a feasible time whenever the phone is idle. However, these events can also be uploaded any time in the following way:

MikrosManager.Instance.AnalyticsController.FlushEvents();