Unity SDK for PC

You can send data from your Unity app to Treasure Data, using our Unity SDK library. By using the SDK, you don't have to install anything server-side to track website activities.

GDPR Compliance

To support compliance with national and global data privacy requirements such as the European General Data Privacy Regulation, our SDK provides methods that control the collection and tracking of personal data and metadata in applications and websites. When your company defines data privacy policies around personal data, you can use these methods in your code to implement default data collection behaviors, and add controls for individuals to use to manage data collection and privacy themselves.

Customers of Treasure Data must ensure that their usage of the SDK, including its use that collects personal data, complies with the legal agreement that governs access to and use of the Treasure Data service, including specifically Treasure Data's current Terms of Service, privacy policy, and Privacy Statement for Customer Data.

Prerequisites

- Basic knowledge of Unity Development
- Basic knowledge of Treasure Data.

Install the Library

Download the most recent version of our Unity package and import it into your Unity project using Assets > Import Package > Custom Package.

TD Unity SDK **version 0.1.11 or newer** is available.

Enable PC Mode

Add the symbol TD_SDK_DEV_MODE to Player Settings > Scripting Define Symbols

Initialize the Library

Next, initialize the library in your app as follows.

```
public class MyTreasureDataPlugin : MonoBehaviour {
  private TreasureData td = null;

  void Start() {
    TreasureData.InitializeApiEndpoint("https://in.treasuredata.com");
    td = new TreasureData("YOUR_API_KEY");
    ...
    // If you want to use a TDClient over scenes, pass `true` to
    // `SimpleTDClient.Create` to prevent it from being removed.
    // Otherwise, pass 'false' If you want to use a TDClient only
    // within a scene, don't pass `true` to `SimpleTDClient.Create`
    // so that you can prevent object leaks.
    td.SetSimpleTDClient(SimpleTDClient.Create(true));
  }

  void OnApplicationPause(bool pauseStatus) {
    // Make an open request whenever app is resumed
    if (!pauseStatus) {
      td.UploadEvents(
        delegate() { Debug.LogWarning ("UploadEvents Success!!! "); },
        delegate(string errorCode, string errorMsg) { Debug.LogWarning ("UploadEvents Error!!! errorCode=" +
errorCode + ", errorMsg=" + errorMsg); }
      );
    }
  }

  void OnApplicationQuit()
  {
    // Make an open request when app is quitting
    td.UploadEvents(
      delegate() { Debug.LogWarning ("UploadEvents Success!!! "); },
      delegate(string errorCode, string errorMsg) { Debug.LogWarning ("UploadEvents Error!!! errorCode=" +
errorCode + ", errorMsg=" + errorMsg); }
    );
  }
}
```

The API key can be retrieved from the TD Console. It's recommended to use **write-only API key** for SDKs.

When to upload and how often to upload buffered events depends on the characteristics of your application. Good times to upload include:

- When the current screen is closing or moving to background
- When closing the application

Enable Tracking of Personal Information (Optional)

To comply with data privacy regulations in various domains, and specifically the EU's GDPR, the Treasure Data Unity SDK does not collect certain event metadata that is personally identifiable. Specifically, the following information is not collected by default:

- td_uuid - client's identifier, unique to this installation of this application on this device

The td_uuid is needed if you want to track individual users and analyze their data within and across user sessions, associate the tracked behavior with a real-world individual, and more.

You must review your data collection policy with your company's data privacy officer and legal counsel to determine what if any personal information you should collect. If you decide to enable tracking of individuals, we also recommend that you integrate with a consent management system to track individual user opt-ins to tracking.

When you have determined the user consent, you can enable the collection of personal data. For example:

```
td.EnableAutoAppendUniqId();
```

In testing, review the information you are collecting to ensure that you have the personal information you intended and nothing more.

Send Events to the Cloud

Next, call the AddEvent() function at the appropriate time within your applications. The following example shows that an event is sent to the table table_b within database database_a.

```
Dictionary<string, object> ev = new Dictionary<string, object>();
ev["str"] = "strstr";
ev["int"] = 12345;
ev["long"] = 12345678912345678;
ev["float"] = 12.345;
ev["double"] = 12.3459832987654;
ev["bool"] = true;
td.AddEvent("database_a", "table_b", ev);
```

IP whitelist won't be applied to any import from Unity SDK. Also, we've noted that browsers frequently specify invalid timestamps (such as 1970/01/01), therefore we're currently ignoring records that have a timestamp older than 7 days, and newer than 3 days ahead.

More Features and Options

Tracking Application Lifecycle Events

Optionally, this SDK can be enabled to automatically capture app lifecycle events (disabled by default). You must explicitly enable this option. You can set the target table through `setDefaultTable()`:

```
[RuntimeInitializeOnLoadMethod(RuntimeInitializeLoadType.BeforeSceneLoad)]
static void OnRuntimeInitialization() {
  // TreasureData client setup...
  TreasureData.DefaultTable = "app_lifecycles";
  TreasureData.Instance.EnableAppLifecycleEvent();
}
```

There are 3 kinds of events that are tracked automatically: Application Open, Install and Update. These events are captured along with relevant metadata dependent on the specific type of event:

Application Open

```
 {
     "tdunityevent": "TDUNITYAPPOPEN",
     "tdapp_ver": "1.0",
     ...
 }
```

Application Install

```
{
  "tdunityevent": "TDUNITYAPPINSTALL",
  "tdapp_ver": "1.0",
  ...
}
```

Application Update

```
    {
        "tdunityevent": "TDUNITYAPPUPDATE",
        "tdappver": "1.1",
        "tdappprevioussver": "1.0",
        ...
```

Opt-out

This SDK can opt-out all events tracking for a particular device and de-identify users by resetting (or disabling completely) the `td_uuid.` You can change the `td_uuid.` to a different id for all subsequent events:

```
    TreasureData.Instance.disableCustomEvent();        // disable your own events, ones that collect through
manual calls to addEvent...
    TreasureData.Instance.disableAppLifecycleEvent();  // disable the auto-collected app lifecycle events
```

Unlike other option flags, enable/disableCustomEvent and enable/disableAppLifecycleEvent are persistent settings, which means that the settings survive across app launches. It is important to use this call whenever your users indicate that they don't want to be tracked. It is not necessary to call the options on every Treasure Data client setup.

```
    TreasureData.Instance.resetUniqId();
    TreasureData.Instance.disableAppendUniqId();  // temporary configuration, need to call on Treasure Data
client setup
```

The option flag `resetUniqId` also adds an audit event to the `DefaultTable`

```
    {
        "td_unity_event": "forget_device_uuid",
        "td_uuid": ,

    }
```

Retry uploading and deduplication

The SDK imports events in one style with the combination of these features:

- This SDK keeps buffered events by adding unique keys and retries to upload them until confirming the events are uploaded and stored on the server-side (at least once)
- The server side remembers the unique keys of all events within the past 1 hour by default and can prevent duplicate imports.

Deduplication is a best effort system that identifies a duplicate record if a record with the same identifier is seen in the same dataset, within the last hour at most or within the last 4096 records, whichever comes first.

Next Steps

Our Unity SDK is available through Github. Check the repository to ensure that you have the most recent SDK.

- https://github.com/treasure-data/td-unity-sdk-package