

# AppMetr Unity Plugin

## Интеграция плагина

1. Импортируете плагин ***\*AppmetrUnityPlugin.unitypackage\**** в проект (двойным кликом по файлу или через меню *\*Assets/Import Package/Custom Package...\**).
2. Добавьте на сцену скрипт *Assets/Plugins/AppMetr/AppMetrBehaviour.cs* и не выгружайте его при перезагрузке сцены (можно для этого установить флаг *Undeletable* в его свойствах). Пропишите свойство *Token*.
3. Вызывайте статические методы AppMetr для регистрации игровых событий (см. AppMetr API)
4. *Опционально.* Подписываетесь на события *AppMetrCommandListener.OnCommand*, чтобы выполнять удаленные серверные вызовы. Пример кода:

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class AppMetrManager : MonoBehaviour {
```

```
    void Awake() {
```

```
        AppMetrCommandListener.OnCommand += HandleAppMetrCommand;
```

```
    }
```

```
    void OnDisable() {
```

```
        AppMetrCommandListener.OnCommand -= HandleAppMetrOnCommand;
```

```
    }
```

```
    public void HandleAppMetrCommand(string commandJson) {
```

```
        Debug.Log("AppMetrManager: HandleAppMetrCommand\n" + commandJson);
```

```
    }
```

```
}
```

# AppMetr API

## Класс AppMetrBehaviour

Служит для прослушивания основных событий системы, предоставляемых MonoBehaviour, и передачи их в AppMetr. Должен присутствовать на сцене и не выгружаться в течение всей игровой сессии.

## Класс AppMetr

Это основной класс, предоставляющий доступ к функциям библиотеки.

Методы:

- *Setup*

Инициализирует библиотеку Параметры:

*token* токен приложения

- *TrackSession*

Регистрирует запуск приложения

Параметры:

*properties* дополнительные параметры (сведения о пользователе, например, дата регистрации, имя, возраст)

- *TrackLevel*

Регистрирует достижения нового уровня игроком Параметры:

*level* достигнутый уровень

*properties* дополнительные параметры

- *TrackEvent*

Регистрирует игровое событие Параметры:

*eventName* название события *properties* дополнительные параметры

- *TrackPayment*

Регистрирует совершение платежа Параметры:

*payment* сведения о выполненной покупке *properties* дополнительные параметры

- *AttachProperties*

Добавление параметров Параметры:

*properties* параметры

- *TrackExperimentStart*

Регистрирует начало эксперимента Параметры:

*experiment* имя эксперимента *groupId* группа

- *TrackExperimentEnd*

Регистрирует конец эксперимента Параметры:

*experiment* имя эксперимента

- *Identify*

Идентификация пользователя Параметры:

*userId* идентификатор пользователя

- *Flush*

Форсированная отправка событий на сервер

## **Класс AppMetrCommandListener**

Этот класс предоставляет доступ к событию получения команды от сервера.

Пример использования:

```
AppMetrCommandListener.OnCommand += HandleOnCommand; public void  
HandleOnCommand(string command) {
```

```
Debug.Log("HandleOnCommand:\n" + command); }
```

Пример использования плагина можно посмотреть в

***\*Assets/Plugins/Appmetr/Sample/TestScene.unity\****.

### **ВАЖНО:**

Библиотека не поддерживает Stripping Level = use micro mscorlib.