

Backendless Unity SDK Start-up Document

2015/02/26 Acrodea, Inc.

V1.0

Date	Version	Note
2015-02-26	1.0	初版

本書は Backendless (<http://backendless.com>) および GMO mBaaS Powerd By Backendless (<http://gmo-mbaas.com>) 向けに開発された “Backendless Unity SDK” の導入方法について説明しています。

Unity のインストール

下記のリンクから Unity の開発ツールをダウンロードしてください。

<http://unity3d.com/unity/download>

※ Backendless Unity SDK は Unity 4.1.4～4.5.1 以上で動作確認しています。Unity 4.1.4 より低いバージョンをご利用の場合、一部の API が利用できなくなる可能性があります。

Backendless のアカウントセットアップ

Backendless のアカウント（もしくは GMO mBaaS Powerd By Backendless のアカウント）をお持ちでない場合は アカウントをご用意下さい。

- Backendless . . . <http://backendless.jp>
- GMO mBaaS Powerd By Backendless . . . <http://www.gmo-mbaas.com>

Backendless Unity SDK のダウンロード

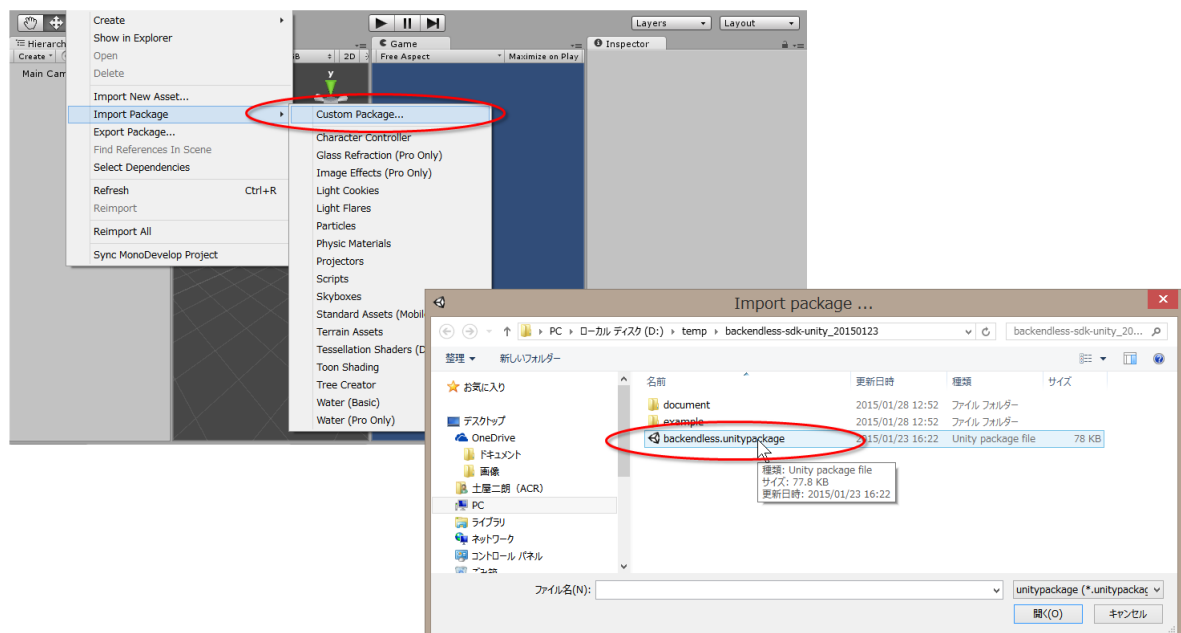
以下の URL から Backendless Unity SDK 一式をダウンロードできます。

<https://github.com/Backendless-Game-SDK/Unity-SDK/releases>

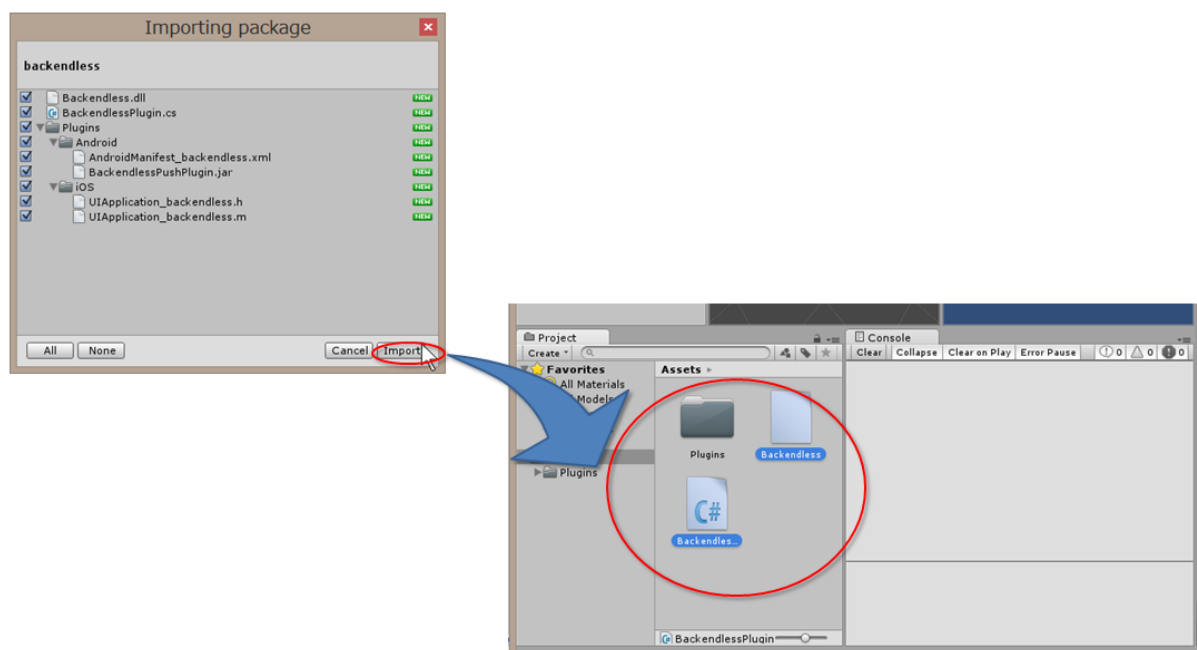
Unity 開発ツールのセットアップ

1. プロジェクトに `backendless.unitypackage` を追加する

Backendless Unity SDK に含まれている backendless.unitypackage を直接開くか、Unity tool の Assets メニューで Import package - Custom package から backendless.unitypackage を選択してください。



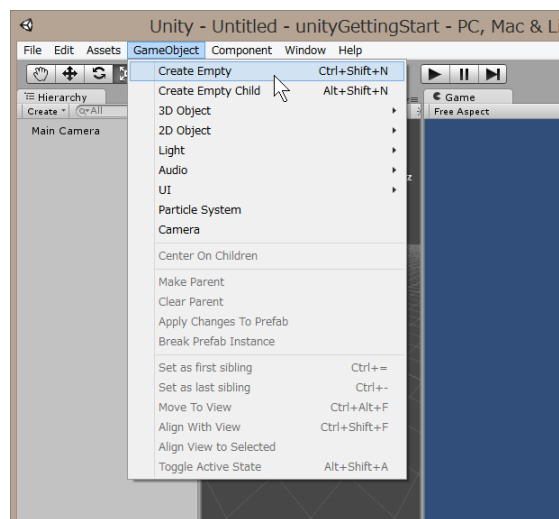
下記のように Import package ウィンドウが現れます。Import ボタンを押して、チェックしたファイルをインポートしてください。



【POINT】 Push Message api を利用しない場合は Plugins より下位のファイルは不要です。
Push Message api を利用する場合は全てのファイルをインポートした上一部ファイル修正
が必要です。詳しくは後述します。

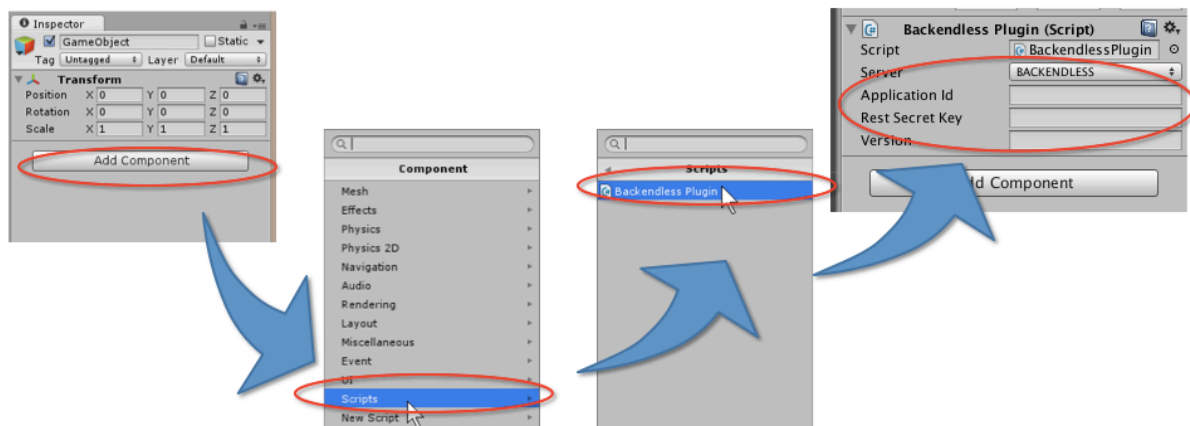
2. GameObject を追加する

“GameObject メニュー” → “Create Empty”から新しい GameObject を追加します。



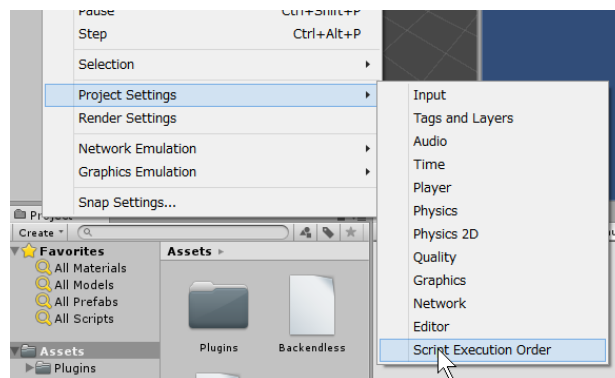
3. 追加した GameObject に Backendless Plugin コンポーネントを追加する

追加した GameObject の Inspector ウィンドウで“Add Component”→ “Scripts”→
“Backendless Plugin”を選択して追加してください。



また、追加した Game Object の名前は自由に付けて頂いて構いません。ここではわかりやすいように Backendless Object としています。

【POINT】 ここで追加するスクリプトは Backendless Unity SDK の初期化を行いますので、他の Backendless Unity SDK の API を利用する前に実行されなくてはなりません。もし、他の script より実行順序が遅くなってしまう場合は、Unity 開発ツールの“Edit メニュー”→ “Project settings”→ “Script Execution Order”を利用して、Backendless Plugin script の実行順序を調整してください。

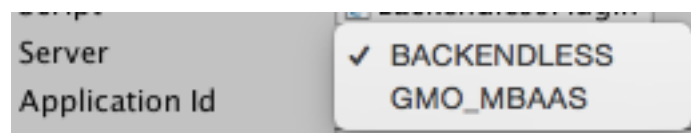


4. Backendless の Application ID, Secret Key を設定する

BackendlessPlugin.cs が GameObject に追加されると Inspector ウィンドウから Server, Application Id, REST Secret Key, Version が入力できるようになります。

Server にはアカウントを発行したサービスを選択してください。

- Backendless → BACKENDLESS
- GMO mBaaS Powerd By Backendless → GMO_MBAAS



次に、Backendless の開発者コンソールにログインして「アプリケーション ID」と「REST シークレットキー」を確認してください。（“管理”>“アプリ設定”で確認できます）。これらの値を先ほど追加した Inspector ウィンドウの “Application ID”, “Rest Secret Key”の欄に記入してください。また、“Version” には “v1” を入れてください。



【POINT】 Backendless の Secret Key はプラットフォーム毎に別のものが用意されていますが、Unity SDK では「REST シークレットキー」を指定してください。

以上で Unity から Backendless を利用する準備ができました。

実際に API を呼び出してみる

それでは 正しく設定できているかどうか、データ保存処理を試してみましょう。

1. オブジェクトを追加してスクリプトを記述

新規の GameObject と C# のスクリプト (“hello”) を作成して、スクリプトを GameObject にドラッグアンドドロップした後、スクリプトを開いて以下の内容を記述してください。

```
using System.Collections;
using System.Collections.Generic;
using BackendlessAPI;
using BackendlessAPI.Async;
using BackendlessAPI.Data;
using BackendlessAPI.Persistence;
using UnityEngine;

public class hello : MonoBehaviour {

    // データ型を #C クラスで定義します
    class Contact : BackendlessEntity
    {
        public string Name { get; set; }
        public int Age { get; set; }
        public string Phone { get; set; }
        public string Title { get; set; }
    }

    void Start () {
        // オブジェクト生成します。
        Contact contact = new Contact { Name = "Jack Daniels", Age = 47, Phone = "0123-45-6789", Title =
"Favorites" };
        // 非同期呼び出し完了時のコールバックメソッド
        AsyncCallback<Contact> callback = new AsyncCallback<Contact> (
            savedData =>
            {
                Debug.Log (" saved " + savedData.Name);
            },
            fault =>
            {
                Debug.Log (fault);
            }
        );
        // 上記コールバックメソッドを指定してデータ保存
        Backendless.Persistence.Of<Contact>().Save(contact, callback);
    }

    // Update is called once per frame
    void Update () {

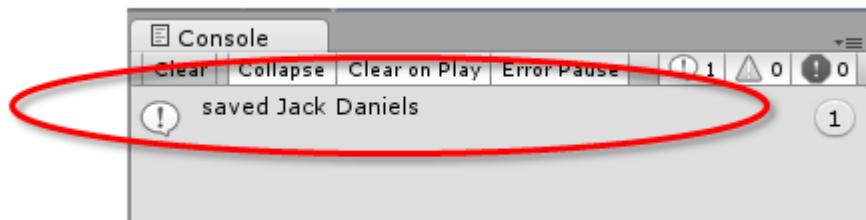
    }

}
```

上のサンプルコードでは、データ保存用の C# クラス “Contact” を定義して、そのオブジェクトインスタンスを新規に作成したあと、非同期メソッドを用いて保存しています。

2. プログラムの実行

書き終わったらプログラムを実行してみてください。Debug.log コマンドにより Unity のコンソールにメッセージが現れます。



3. コンソールの動作確認

次に Backendless のコンソールで「データサービス」を確認してみましょう。新しいテーブル“Contact” が生成されていること、そのテーブルにデータが保存されていることが確認できます。

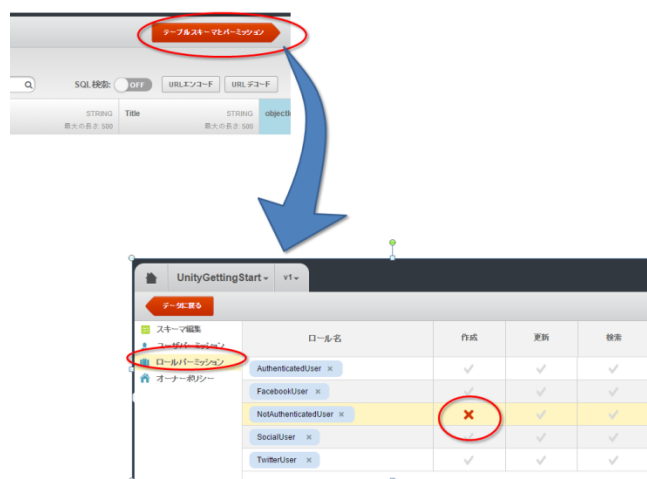


4. エラーハンドリングの確認

次に Backendless のユーザ権限管理機能を利用してわざとデータ保存を拒否してエラーハンドラーを呼び出してみます。

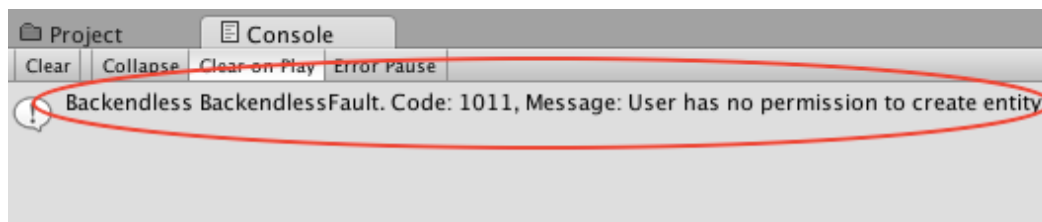
Backendless のコンソールの「データサービス」の右上「テーブルスキーマとパーミッション」ボタンをクリック→左端の「ロールパーミッション」をクリックして、

「NotAuthenticatedUser」ロールに対する「作成」の欄を何回かクリックし「×」印にしてください。



これで「ログイン操作をしていないユーザ」に対する新規データ作成機能が禁止されます。

その後アプリケーションを再生させると Unity のコンソールに権限エラーが検出されていることが確認できます。

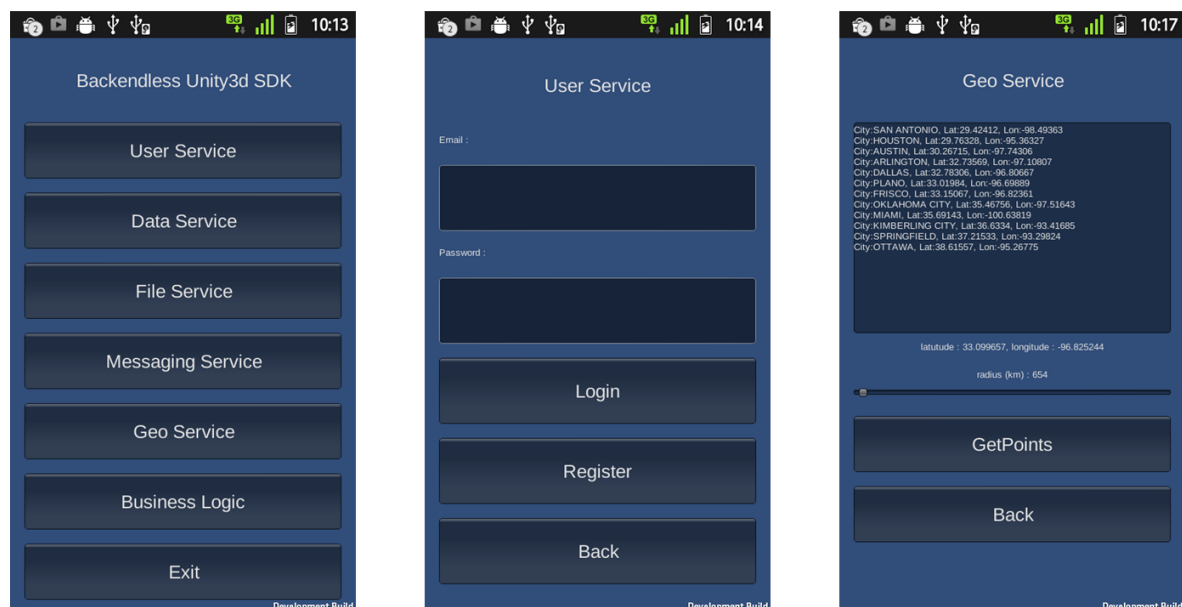


【POINT】 Backendless ではユーザ単位やロール（グループ）単位で、各 API やテーブルに対するアクセス管理が可能です。

SDK 同梱のサンプルアプリについて

Backendless Unity SDK に同梱の Examples フォルダには各 Backendless サービスを利用したサンプルコードを使った例を載せました。以下の機能が利用可能です。

- (1) User Service … ユーザの新規登録やログイン/ログアウトが可能です。
- (2) Data Service…データの保存/取得/削除によって簡易な TODO リストを実現しています。
- (3) File Service… File の Upload と Delete が試せます。
- (4) Messaging Service…Message の発行と購読に加え、Push 通知における Device Registration の実装例が見られます。
- (5) Geo Service …（事前に Backendless コンソール上で「サンプルデータのセットアップ」をしておくとも動作確認しやすくなります。）
- (6) Business Logic …カスタムイベントを確認します（予めサーバ側に同梱のサーバロジックを登録しておく必要があります。）



※Push 通知の動作には別途設定が必要です。

また、Backendless Unity SDK を利用した簡単なゲームアプリのサンプル「タイムゲス」も用意しています。こちらではカスタムビジネスロジックを用いてランキングボードの実装をしています。合わせてご利用下さい。

補足 1 ~ BuildSetting で「Stripping Level」が Disable でない場合

Unity の Build settings で Stripping Level が Disabled ではない場合 Backendless Unity SDK が正常に動作しないこともあります。



このような問題が発生した場合は、Assets folder に link.xml ファイルを作って下記のように stripping 除外を指定してください。

```
<linker>
  <assembly fullname="System">
    <type fullname="System.*" preserve="all"/>
  </assembly>
  <assembly fullname="mscorlib">
    <namespace fullname="System.*" preserve="all"/>
  </assembly>
</linker>
```

補足 2 ~ Error Handling について

サーバーエラーをハンドルするため BackendlessFault class が用意されています。error code と message を取得することができます。

型定義

```
using System;

namespace BackendlessAPI.Exception
{
    public class BackendlessFault
    {
        private readonly string _faultCode;
        private readonly string _message;
        private readonly string _detail;

        public string Detail
        {
            get { return _detail; }
        }

        public string FaultCode
        {
            get { return _faultCode; }
        }

        public string Message
        {
            get { return _message; }
        }

        public override string ToString()
        {
            return String.Format("Backendless BackendlessFault. Code: {0}, Message: {1}", FaultCode ?? "N/A", Message ?? "N/A");
        }
    }
}
```

Asynchronous

```
public delegate void ErrorHandler(BackendlessFault _backendlessFault);
```

Synchronous

```
using System;

namespace BackendlessAPI.Exception
{
    public class BackendlessException : System.Exception
    {
        public BackendlessFault BackendlessFault { get; }
        public string FaultCode { get; }
        public override string Message { get; }
        public string Detail { get; }
        public override string ToString();
    }
}
```

補足 3 Push 通知機能を利用する場合

Backendless Unity SDK は APNS 及び GCM による Push 通知の受信機能を利用することができます。これらの機能を利用する場合以下の注意事項をご確認下さい。

- backendless.unitypackage をインストールする際に、Plugins folder 以下のファイルもインポートして下さい。（デフォルトではインポートされるようになっています）。
- BackendlessPlugin.cs 内で ENABLE_PUSH_PLUGIN の DFINE を有効にしてください。（Plugin によるワーニングを発生させないため、デフォルトではコメントアウトされています。）

本ドキュメントは米国 Backendless 社が提供するサービスを Unity プラットフォームから利用するための SDK の導入方法について説明したものです。

免責および再配布条件

- 本ドキュメント及び本ドキュメントが説明するソフトウェア・サービス・プログラムの使用は、これらの利用者（以下利用者と略す）自身の責任でなされるものとします。
- すべての資料、情報、ソフトウェア、プログラムおよびサービスは現状のまま提供され、いかなる保証も適用されません。株式会社アクロディアは、適用される法律の許す限りにおいて、法律上の瑕疵担保責任、商品性の保証、特定目的への適合性の保証、権利の不侵害の保証を含むいかなる明示もしくは黙示の保証責任も一切負いません。
- 利用者は、本ドキュメント及び本ドキュメントが説明するソフトウェア・サービス・プログラムを取得もしくは利用する場合においては、利用者ご自身の判断および責任において行っていただくこと、また、その結果として発生するデータの損失または利用者のコンピューター・システムへの損傷などのいかなる損害もすべて利用者の責任となることを理解し、同意するものとします。
- 本ドキュメントに記載された仕様は、予告無く変更される場合があります。
- 特に明記されている場合を除き、Backendless サービスに関する全て（ソフトウェア、電子データ、画像、ドキュメント）の権利は米国 Backendless 社に帰属します。
- 本ドキュメントにて説明されている Backendless Unity SDK は Apache 2.0 ライセンスに基づき株式会社アクロディアが作成したものであり、また本ドキュメントの著作権は株式会社アクロディアに帰属します。
- 本ドキュメントの複製、頒布、展示、実演は、Creative Commons 表示- 継承 ライセンス（CC BYSA2.1 JP）に基づいて許可されます。<http://creativecommons.org/licenses/by-sa/2.1/jp/>
- 本ドキュメントが説明するソフトウェアのうち Backendless Unity SDK に関しては Apache License 2.0（<http://www.apache.org/licenses/LICENSE-2.0.html>）に基づいて再利用が可能です。