

Autodesk App ストア: デスクトップ アプリ用の Entitlement API (権限 API)

Autodesk App ストアには、Autodesk App ストアに公開するデスクトップ アプリのために、単純なコピー保護機構を構築できる Entitlement API (権限 API) サービスが用意されています。Entitlement API サービスは、エンドユーザがアプリの使用権限を持っているかどうか識別するために使用する、REST ベースの “checkentitlement” API が公開されています。

API の詳細 API:

ベース URL: <https://apps.autodesk.com/>

エンド ポイント: webservices/checkentitlement

Http メソッド: GET

パラメータ: ?userid=***&appid=***

戻り値: JSON オブジェクト

まず、userid が示すユーザの ID を検証する必要があります。userid が、オートデスク製品内や Autodesk App ストアにログインするために使われるユーザ名とは異なる、内部的に使用される ID であることに注意してください。

Autodesk App ストアに記載したアプリからこの API を使用するためには、Entitlement API にアプリの一意な ID とオートデスク製品内で実行中のアプリに現在サインイン中のエンドユーザの userid を渡す HTTP (REST) 呼び出しをおこなう必要があります。Entitlement API は、エンドユーザがこのアプリを使用する ‘権限’ を持っているかどうか応答します (例、エンドユーザがアプリを購入していたかどうかを返す)。

開発者は、契約期間を持つ Subscription 形態のアプリからでも、Entitlement API を使用することが出来ます。Subscription アプリでは、ユーザの Subscription が期限切れかどうか依存した結果が返されます (例、この API は、Subscription 期間が有効な場合のみ、ユーザがアプリの利用権限を持つように応答します)。

Autodesk App ストアにアプリを提出したら、すぐに、アプリの一意な ID を取得することが出来ます (アプリの ID 識別で問題がある場合には、appsubmissions@autodesk.com までメールでお知らせください)。



次の URL からサンプルのリクエストを参照することが可能です:

<https://apps.autodesk.com/webservices/checkentitlement?userid=2N5FMZW9CCED&appid=2024453975166401172>

この例では、userid は “2N5FMZW9CCED”、appid は “2024453975166401172” になっています。

戻り値の JSON は次のようになります:

```
{"UserId": "2N5FMZW9CCED", "AppId": "2024453975166401172", "IsValid": false, "Message": "Ok"}
```

‘IsValid’ 値は、ユーザがアプリの使用権限を持っている場合に true を返します。その他の場合には、false を返します。

可能性のある戻り値は、次のうちの 1 つです:

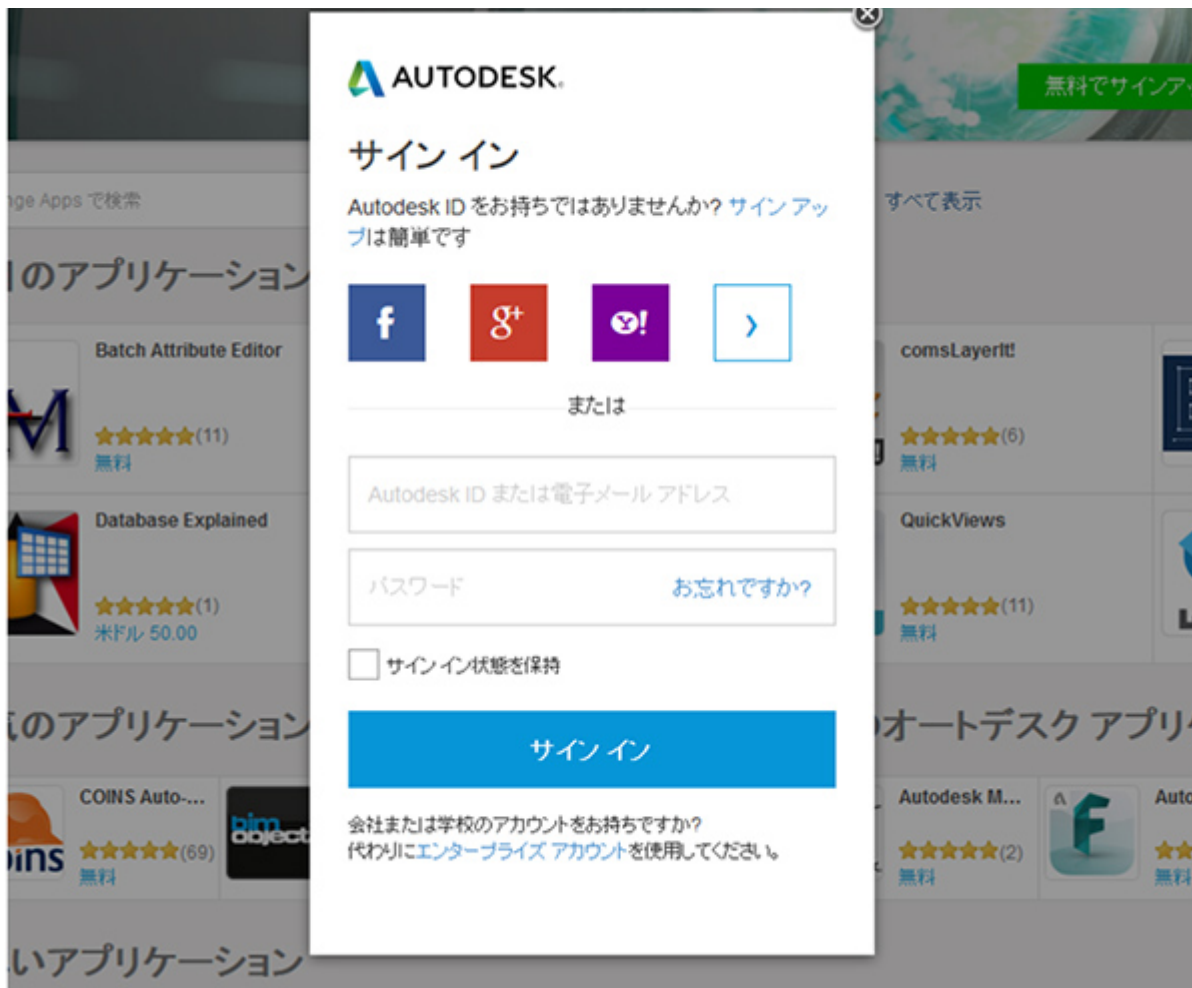
“OK” - 現在の呼び出しは正常

“Invalid parameters(s)” – User Id または App Id が未設定

“Please use https” – リクエストが https を使用していない

どのように Entitlement API が動作するのですか？

Autodesk App ストアからアプリをダウンロードするには、下図に示されているように、エンドユーザは Autodesk App ストアに Autodesk アカウント (Autodesk ID) を使ってサインインする必要があります:



エンドユーザによるサインインは、Autodesk App ストアからアプリをダウンロードしたユーザの一覧を保持しておくことを可能にします。Entitlement API は、アプリを使用する“権利”を持つかどうかを、この一覧を使って実現します。

どのようにアプリ内から Entitlement API を利用するのですか？

前述の通り、Entitlement API を使用するには、次の情報が必要となります。

User ID
App ID

Autodesk App ストアにアプリを提出することで、App ID を取得することが出来ます。この ID は、Autodesk App ストア内にアプリが記載され続ける限り、同じ値を維持します。

さて、オートデスク製品に依って、アプリ内からユーザ ID を取得する適切な方法/API を利用することが出来ます。

Product	API
AutoCAD(または、AutoCAD ベースの業種別製品)	読み取り専用システム変数 "ONLINEUSERID" を利用
Revit	Application.LoginUserId API を利用
Inventor	アンマネージ “AdWebServices” API か、.NET で “GetUserId” を利用 <pre>[DllImport("AdWebServices", EntryPoint = "GetUserId", CharSet = CharSet.Unicode)]</pre> <pre>private static extern int AdGetUserId(StringBuilder userid, int buffersize);</pre>

ユーザの権限を識別するために、アプリ内から User ID と App ID を使って REST 呼び出しをおこないます。下記のそのサンプルコードです。ここでは、C# での REST API の使用を簡素化する目的で、[RestSharp](#) ライブラリを使っています。

AutoCAD(または、AutoCAD ベースの業種別製品)

```
[CommandMethod("CheckEntitleAutoCAD")]
static public void CheckEntitleAutoCAD()
{
    String _appId = "appstore.exchange.autodesk.com:screenshot:en";
    //Steps to get the user id
    String _userId = Application.GetSystemVariable("ONLINEUSERID") as String;
    //Not logged in with Autodesk Id, hence we can not get user id
    if (_userId.Equals(""))
    {
        return;
    }
    //check for online entitlement
    RestClient client = new RestClient("https://apps.autodesk.com");
    RestRequest req = new RestRequest("webservices/checkentitlement");
    req.Method = Method.GET;
    req.AddParameter("userid", _userId);
    req.AddParameter("appid", _appId);
    ServicePointManager.ServerCertificateValidationCallback += (sender, certificate, chain, sslPolicyErrors) => true;

    IRestResponse resp = client.Execute(req);

    if (resp.Data != null && resp.Data.IsValid)
    {
        //User has downloaded the App from the store and hence is a valid user...
    }
    else
    {
        //Not a valid user. Entitlement check failed.
    }
}
```

Revit

```
//Set values specific to the environment
public const string _baseApiUrl = @"https://apps.autodesk.com/";
//This is the id of your app.
public const string _appId = @"2024453975166401172";

//Command to check an entitlement
public Autodesk.Revit.UI.Result Execute(ExternalCommandData commandData, ref string message,
Autodesk.Revit.DB.ElementSet elements)
{
    //Get the top elements
    UIApplication uiApp = commandData.Application;
    Application rvtApp = uiApp.Application;

    //Check to see if the user is logged in.
    if(!Application.IsLoggedIn) {
        TaskDialog.Show("Entitlement API", "Please login to Autodesk 360 first\n");
        return Result.Failed;
    }

    //Get the user id, and check entitlement
    string userId = rvtApp.LoginUserId;
    bool isValid = Entitlement(_appId, userId);

    if (isValid)
    {
        //The user has a valid entitlement
        //
    }
}
```

```

    }

    //For now, just display the result
    string msg = "userId = " + userId
    + "\nappId = " + _appId
    + "\nisValid = " + isValid.ToString();
    TaskDialog.Show("Entitlement API", msg);

    return Result.Succeeded;
} ///=====
    /// URL: https://apps.exchange.autodesk.com/webservices/checkentitlement
    ///
    /// Method: GET
    ///
    /// Sample response
    /// {
    /// "UserId":"2N5FMZW9CCED",
    /// "AppId":"2024453975166401172",
    /// "IsValid":false,
    /// "Message":"Ok"
    /// }
    /// =====
    private bool Entitlement(string appId, string userId)
    {
        //REST API call for the entitlement API.
        //We are using RestSharp for simplicity.
        //You may choose to use another library.

        //(1) Build request
        var client = new RestClient();
        client.BaseUrl = new System.Uri(_baseApiUrl);

        //Set resource/end point
        var request = new RestRequest();
        request.Resource = "webservices/checkentitlement";
        request.Method = Method.GET;

        //Add parameters
        request.AddParameter("userid", userId);
        request.AddParameter("appid", appId);

        //(2) Execute request and get response
        IRestResponse response = client.Execute(request);

        //Get the entitlement status.
        bool isValid = false;
        if (response.StatusCode == HttpStatusCode.OK)
        {
            JsonSerializer deserial = new JsonSerializer();
            EntitlementResponse entitlementResponse = deserial.Deserialize(response);
            isValid = entitlementResponse.IsValid;
        }

        //
        return isValid;
    }
}

```

Inventor

```

class WebServicesUtils
{
    [DllImport("AdWebServices", EntryPoint = "GetUserId", CharSet = CharSet.Unicode)]
    private static extern int AdGetUserId(StringBuilder userid, int buffersize);

    [DllImport("AdWebServices", EntryPoint = "IsWebServicesInitialized")]
    private static extern bool AdIsWebServicesInitialized();
}

```

```

[DllImport("AdWebServices", EntryPoint = "InitializeWebServices")]
private static extern void AdInitializeWebServices();

[DllImport("AdWebServices", EntryPoint = "IsLoggedIn")]
private static extern bool AdIsLoggedIn();

[DllImport("AdWebServices", EntryPoint = "GetLoginUserName", CharSet = CharSet.Unicode)]
private static extern int AdGetLoginUserName(StringBuilder username, int buffersize);

internal static string _GetUserId()
{
    int buffersize = 128; //should be long enough for userid
    StringBuilder sb = new StringBuilder(buffersize);
    int len = AdGetUserId(sb, buffersize);
    sb.Length = len;

    return sb.ToString();
}

internal static string _GetUserName()
{
    int buffersize = 128; //should be long enough for username
    StringBuilder sb = new StringBuilder(buffersize);
    int len = AdGetLoginUserName(sb, buffersize);
    sb.Length = len;

    return sb.ToString();
}

public static string GetUserId(out string userName)
{
    AdInitializeWebServices();

    if (!AdIsWebServicesInitialized())
        throw new Exception("Could not initialize the web services component.");

    if (!AdIsLoggedIn())
        throw new Exception("User is not logged in.");

    string userId = _GetUserId();
    if (userId == "")
        throw new Exception("Could not get user id.");

    userName = _GetUserName();
    if (userName == "")
        throw new Exception("Could not get user name.");

    return userId;
}
}
//inside your command check for the entitlement
String _appId = "2024453975166401172";
//Steps to get the user id
string userName;
string _userID = WebServicesUtils.GetUserId(out userName);
//Not logged in with Autodesk Id, hence we can not get user id
if (_userID.Equals(""))
{
    return;
}
//Check for online entitlement
RestClient client = new RestClient("https://apps.autodesk.com");
RestRequest req = new RestRequest("webservices/checkentitlement");
req.Method = Method.GET;
req.AddParameter("userid", _userID);
req.AddParameter("appid", _appId);

```

```
ServicePointManager.ServerCertificateValidationCallback += (sender, certificate, chain, sslPolicyErrors) => true;

IRestResponse resp = client.Execute(req);

if (resp.Data != null && resp.Data.IsValid)
{
    //User has downloaded the App from the store and hence is a valid user...
}
else {
    //Not a valid user. Entitlement check failed.
}
}
```

Entitlement API を使用するベスト プラクティス

Entitlement API は Web-ベースのサービスであるため、エンドユーザがこのサービスを呼び出できるように、インターネットに接続しておく必要があります。アプリ ユーザがオフラインになった場合には（インターネット接続の瞬断を想定）、インターネットへの接続を要求する前に、数十分のアプリの使用期間（猶予期間）を設定することをお勧めします。これは、アプリを実行した各時間のタイムスタンプ文字列を記録して、次回の実行時にそれをチェックするコードを追加することで実現できます。

同様に、アプリがユーザの権限の有効性をチェックしたら、使用中のローカル コンピュータ内に、その情報を書き込み、ユーザがオフラインでもアプリを使用することをかのようにすることもお勧めしています（もちろん、オンラインになった時点で権限を再チェックする必要もあります）。

Entitlement API は Web サービスであるため、頻繁な呼び出しは、アプリのパフォーマンスに影響を与える可能性もあります。このため、アプリが実装するコマンドの実行時に毎回呼び出しをするのではなく、ロード時や初回の実装コマンドの実行時のみなど、アプリのスタート時に呼び出しをおこなうことを推奨しています。

Entitlement Web サービス API を使ったコピー保護機構の構築について、ご質問等があれば、appsubmissions@autodesk.com までメールでお問い合わせください。