

Autodesk App Store: Entitlement API for desktop Apps

The Autodesk App Store has an Entitlement API service with which you can build a simple copy protection system for your Autodesk App Store desktop Apps. The Entitlement API service exposes a REST based “**checkentitlement**” API that you can use to identify whether a user has an ‘entitlement’ to use your App or not.

Details on the API:

Base URL: <https://apps.autodesk.com>

End Point: `webservices/checkentitlement`

Http Method: GET

Parameters: `?userid=***&appid=***`

Return : Json object.

Here `userid` is the ID of the user whose entitlement needs to be verified. **Please note the `userid` is the internal ID, which is different from the username used to log into the store or into different Autodesk products.**

To use this API, from your App make a simple HTTP (REST) call to the Entitlement API, passing in the unique ID of your App, and the `userid` of the customer currently signed in to their Autodesk ID from the Autodesk product in which your App is running. The Entitlement API response will tell you whether the user has an ‘Entitlement’ to use your App (i.e. it tells you if this user has bought this App or not).

You can use the Entitlement API in your subscription Apps too. In subscription Apps, the result returned depends on whether the user’s subscription has expired or not. (i.e., this API will respond that the user has an entitlement for the App only while the subscription is valid). You can get the unique ID of your App once you submit the App in the Autodesk App Store (please let us know if you have any problem in identifying the id of your App).



Here is a sample request URL:

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

Here the `userid` is “2N5FMZW9CCED” and `appid` is “2024453975166401172”.

The return JSON is:

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

The ‘IsValid’ value will be true if the user has entitlement to the App. Otherwise, IsValid will be false. Possible return messages can be any one of the following):

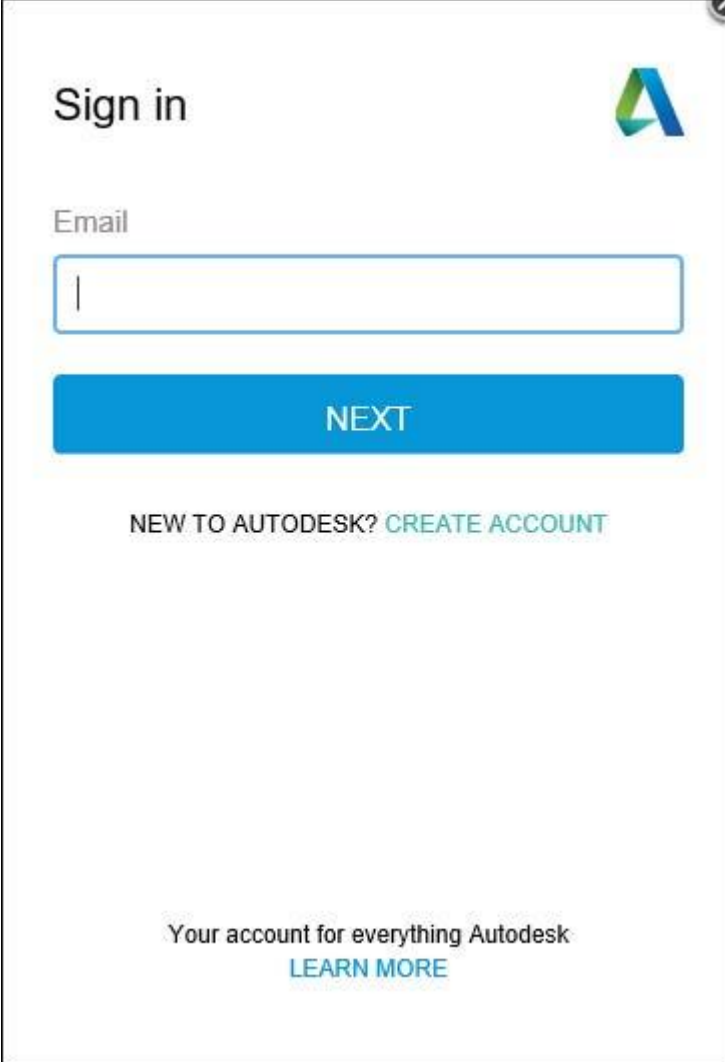
“OK” - current call is correct

“Invalid parameters(s)” – `userid` or `appid` is not set,

“Please use https” - the request is not using https.

How Does the Entitlement API work?

To download any App from the Autodesk App store users' needs to sign-in to the Autodesk App Store using an Autodesk account as shown below:



The image shows a screenshot of the Autodesk App Store sign-in interface. At the top left, it says "Sign in" next to the Autodesk logo. Below that is an "Email" label and a text input field containing a vertical bar. A blue "NEXT" button is positioned below the input field. Underneath the button, it says "NEW TO AUTODESK? CREATE ACCOUNT". At the bottom of the page, it says "Your account for everything Autodesk" with a "LEARN MORE" link below it.

This (signing by users) allows the Autodesk App Store to maintain a list of users who have downloaded the App from the store. The Entitlement API makes use of this list to tell you whether the user has an 'entitlement' to use your App or not.

How to use the Entitlement API from your App

As explained above, you need the information below to use the entitlement API

1. User ID
2. App ID

Your App ID can be obtained once you submit the App in the Autodesk App Store. This ID will remain the same for the life of your App in the Autodesk App Store.

Now, you can use the appropriate method/API depending upon the Autodesk product to get the user ID from your App.

Product	API
AutoCAD (&verticals)	Read system variable "ONLINEUSERID"
Revit	Use API Application.LoginUserId.
Inventor	<p>Inventor 2022 and previous versions: Use unmanaged "AdWebServices" API, "GetUserId" in .NET [DllImport("AdWebServices", EntryPoint = "GetUserId", CharSet = CharSet.Unicode)] private static extern int AdGetUserId(StringBuilder userid, int buffersize);</p> <p>Inventor 2023 and later versions: Use Application. LoginUserId</p>

Using User ID and App ID, make the REST call in your App to identify the entitlement of the user. Below is some sample code. Here we are using the RestSharp library to simplify the use of REST API in C#.

AutoCAD (& AutoCAD verticals)

```
[CommandMethod("CheckEntitleAutoCAD")]
```

```
static public void CheckEntitleAutoCAD()
```

```
{
```

```
    String _appID = "2024453975166401172";
```

```
    //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<EntitlementResult> resp = client.Execute<EntitlementResult>(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.
```

```
    }
```

```
}
```

```
[Serializable]
public class EntitlementResponse
{
    public string UserId { get; set; }
    public string Appld { get; set; }
    public bool IsValid { get; set; }
    public string Message { get; set; }
}
```

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 _appld = @"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(_appld, userId);
```

```
    if (isValid)
    {
        //The user has a valid entitlement
        //<YOUR HANDLER CODE HERE>
    }
```

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

```
    return Result.Succeeded;
}
```

```
///=====
/// URL: https://apps.autodesk.com/webservices/checkentitlement
```

```

///
/// Method: GET
///
/// Sample response
/// {
/// "UserId":"2N5FMZW9CCED",
/// "AppId":"2024453975166401172",
/// "IsValid":false,
/// "Message":"Ok"
/// }
/// =====

private bool Entitlement(string appld, 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", appld);

    //(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<EntitlementResponse>(response);
        isValid = entitlementResponse.IsValid;
    }

    //
    return isValid;
}
}
[Serializable]
public class EntitlementResponse
{

```

```

        public string UserId { get; set; }
        public string Appld { get; set; }
        public bool IsValid { get; set; }
        public string Message { get; set; }
    }

```

Inventor:

For Inventor 2022 and previous versions use the AdWebServices to get the login UserId: (see [Remarks](#) if your tool works with Inventor 2020)

```

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;
}
}

```

For Inventor 2023 and later versions use the LoginUserId to get the login UserId:
string _userID = Application.LoginUserId;
string userName= Application.LoginUserName;

```

//inside your command check for the entitlement
string _appID = "2024453975166401172";
//Steps to get the user id (below two lines are for Inventor 2022 and previous versions, from Inventor
2023 replace them with above two lines)
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<EntitlementResult> resp = client.Execute<EntitlementResult>(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.
}
[Serializable]
public class EntitlementResponse
{
    public string UserId { get; set; }
    public string AppId { get; set; }
    public bool IsValid { get; set; }
    public string Message { get; set; }
}

```

Remarks: In Inventor 2020, the login mechanism is upgraded from AdWebservice to AdSSO(single sign on) so that legacy AdWebservice needs to be initialized once to get the user Id, to workaround this you can call the Application.Login(hidden API) before calling the above entitlement API.

Best practices to use Entitlement API

As the Entitlement API is web-based service, users have to be connected to the internet to make calls to this service. If users of your App are offline (not connected to internet) then it is strongly recommended that you allow the user to use the App for certain amount of time before requiring them to connect to the internet. This can be done by adding code to storing a timestamp each time your App runs, and checking it on the next run.

Similarly, once your app has verified a user's entitlement, we recommend you store this on the local computer and allow the user to use your App offline (and then recheck the entitlement the next time they are online).

Frequently calling of Entitlement web service will have an impact on performance of your App – we recommend you call this once when your App starts up, and not before every time one of your commands is used during a session.

If you have any questions/doubts about building copy protection system using Entitlement web service, then please email to appsubmissions@autodesk.com