Autodesk App Store: Entitlement API for desktop Apps

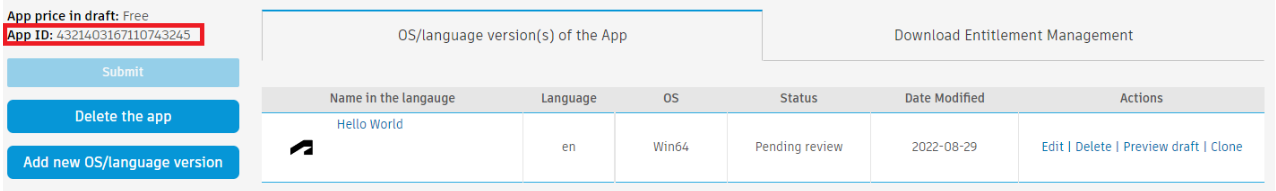
The Autodesk App Store has an Entitlement API service with which you can build a simple copy protection system for your Autodesk App Store desktop Apps. The Entitlement API service exposes a REST based “**checkentitlement**” API that you can use to identify whether a user has an ‘entitlement’ to use 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).



App price in draft: Free
App ID: 4321403167110743245

Submit
Delete the app
Add new OS/language version
OS/language version(s) of the App
Download Entitlement Management

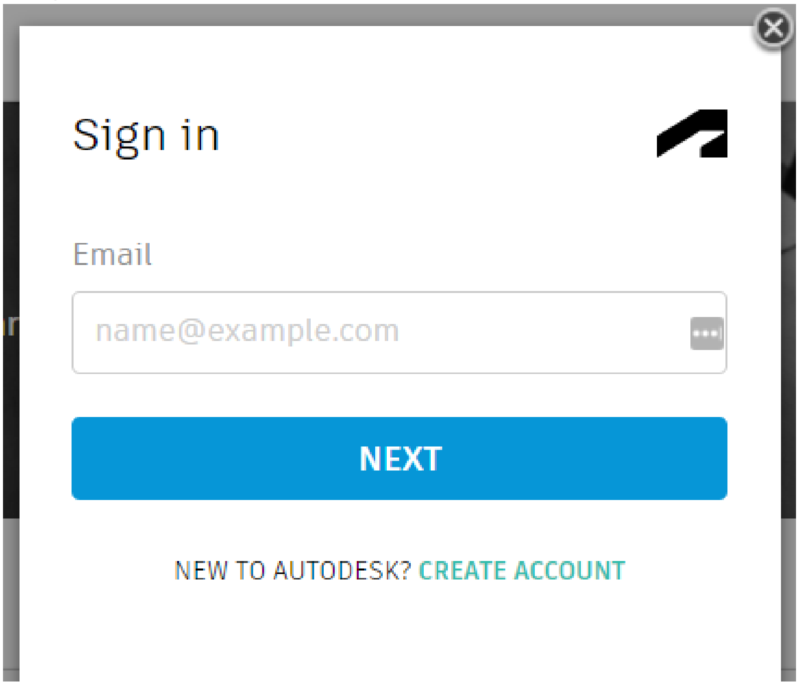
Name in the langauge	Language	OS	Status	Date Modified	Actions
Hello World	en	Win64	Pending review	2022-08-29	Edit | Delete | Preview draft | Clone

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:



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	Use API Application.LoginUserId.

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 AppId { 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 _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
                //<YOUR HANDLER CODE HERE>
            }

            //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.autodesk.com/webservices/checkentitlement
        ///
        /// Method: GET
        ///
        /// Sample response
        /// {
        /// "UserId":"2N5FMZW9CCED",
        /// "AppId":"202445397516640172",
        /// "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)
            {
                JsonDeserializer deserial = new JsonDeserializer();
    EntitlementResponse entitlementResponse =
    deserial.Deserialize<EntitlementResponse>(response);
    isValid = entitlementResponse.IsValid;
            }

            //
            return isValid;
        }
    [Serializable]
        public class EntitlementResponse
        {
            public string UserId { get; set; }
    public string AppId { get; set; }
    public bool IsValid { get; set; }             public
    string Message { get; set; }
        }
```

Inventor:

```
namespace Entitlement

{

    /// <summary>
    /// This is the primary AddIn Server class that implements the ApplicationAddInServer interface
    /// that all Inventor AddIns are required to implement. The communication between Inventor
and
    /// the AddIn is via the methods on this interface.
    /// </summary>

    [GuidAttribute("963308E2-D850-466D-A1C5-503A2E171552")]
    public class AddInServer : Inventor.ApplicationAddInServer
    {
        #region Data Members

        static readonly HttpClient httpClient = new HttpClient
        {
        BaseAddress = new Uri("https://apps.autodesk.com/webservices/checkentitlement")
        };

        //Inventor application object

        private Inventor.Application m_inventorApplication;

        #endregion

        public AddInServer()
        {
        }

        #region ApplicationAddInServer Members

        public async void Activate(Inventor.ApplicationAddInSite addInSiteObject, bool
firstTime)
        {
            try
            {
            //the Activate method is called by Inventor when it loads the addin
            //the AddInSiteObject provides access to the Inventor Application object
            //the FirstTime flag indicates if the addin is loaded for the first time

            //initialize AddIn members
      m_inventorApplication = addInSiteObject.Application;
            if (m_inventorApplication.LoggedIn) // check if user has logged in
            {

            //string username = m_inventorApplication.LoginUserName;
//returns the logged in username
            string userId = m_inventorApplication.LoginUserId; // returns the logged in
userID
            string appId = "<Enter Your APP ID here>";
```

```
                  string urlParameters = String.Format("?userid={0}&appid={1}", userId, appId);
                  string responseBody = await httpClient.GetStringAsync(urlParameters);
          EntitlementResponse entitlementResponse =
JsonSerializer.Deserialize<EntitlementResponse>(responseBody);

                  if (entitlementResponse.IsValid == true)
                  {
                  // user validated.. execute rest of the code
                  }

                  }
                  else
        {
                  MessageBox.Show("User not logged in");
        }
      }

                  catch(Exception e)
                  {
                  MessageBox.Show(e.ToString());
                  }
            }

            public void Deactivate()
            {
                  //the Deactivate method is called by Inventor when the AddIn is unloaded
                  //the AddIn will be unloaded either manually by the user or
                  //when the Inventor session is terminated

                  try
                  {
      //release inventor Application object
                        Marshal.ReleaseComObject(m_inventorApplication);
        m_inventorApplication = null;

                        GC.WaitForPendingFinalizers();
                        GC.Collect();
                  }
                  catch(Exception e)
                  {
                        MessageBox.Show(e.ToString());
                  }
            }

    public void ExecuteCommand(int CommandID)
    {
      //this method was used to notify when an AddIn command was executed
      //the CommandID parameter identifies the command that was executed

      //Note:this method is now obsolete, you should use the new
      //ControlDefinition objects to implement commands, they have
      //their own event sinks to notify when the command is executed
    }

    public object Automation
            {
                  //if you want to return an interface to another client of this addin,
                  //implement that interface in a class and return that class object
                  //through this property

                  get
```

```
                {
                return null;
                }
            }
            #endregion
    }

    public class EntitlementResponse
    {
            public string UserId { get; set; }
            public string AppId { get; set; }
            public bool IsValid { get; set; }
            public string Message { get; set; }
    }
```

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