Getting started with Autodesk View and Data API

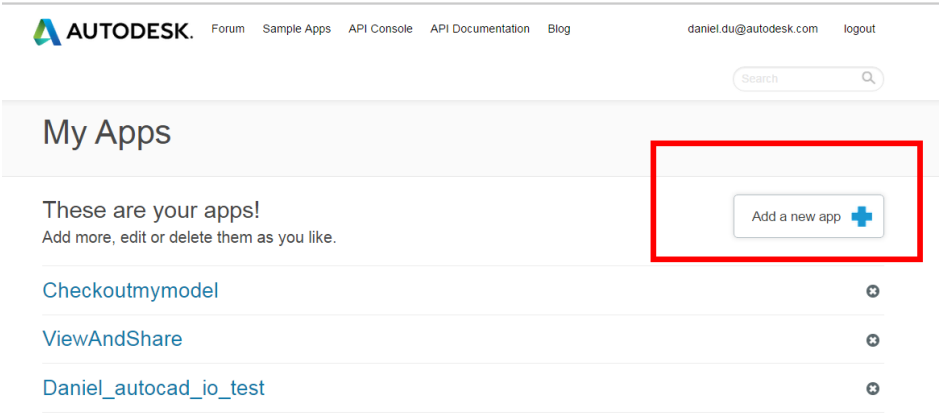
Pre-requisites

Install Visual Studio 2012

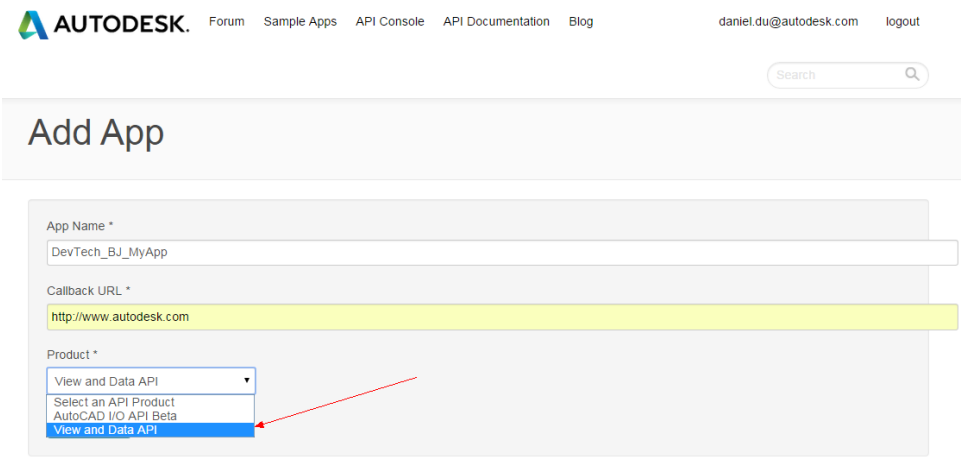
Get Keys

1. Log in http://developer.autodesk.com with your Autodesk ID. If you do not have an Autodesk ID, apply one in a minute.
2. Click [Access Key]

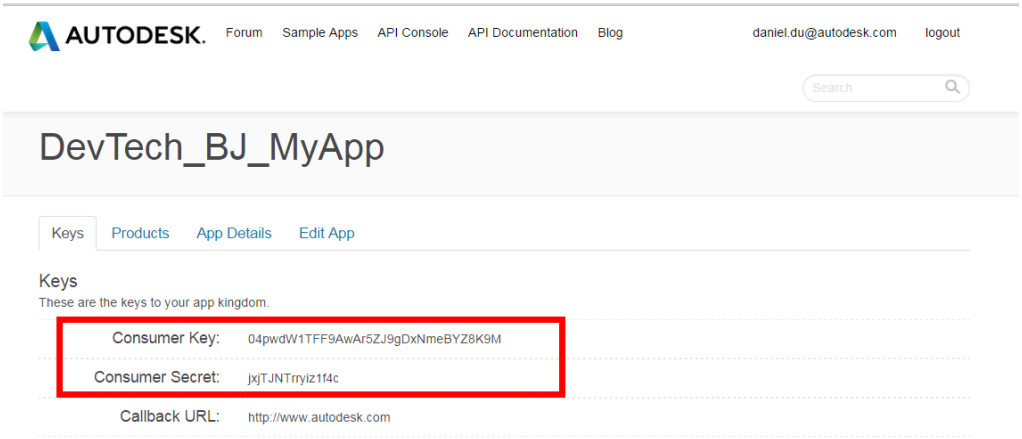
3. Register and Create an application



4. When creating an app, select product as “View and Data API”.
5. Fill in any URL for callback URL. It does not affect the registration currently.



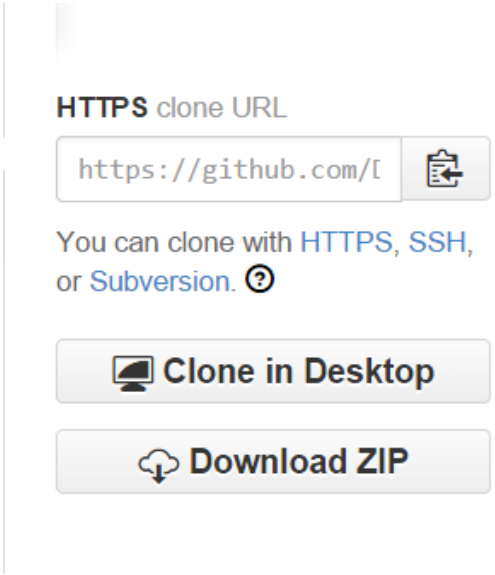
6. You will get the keys similar as below:



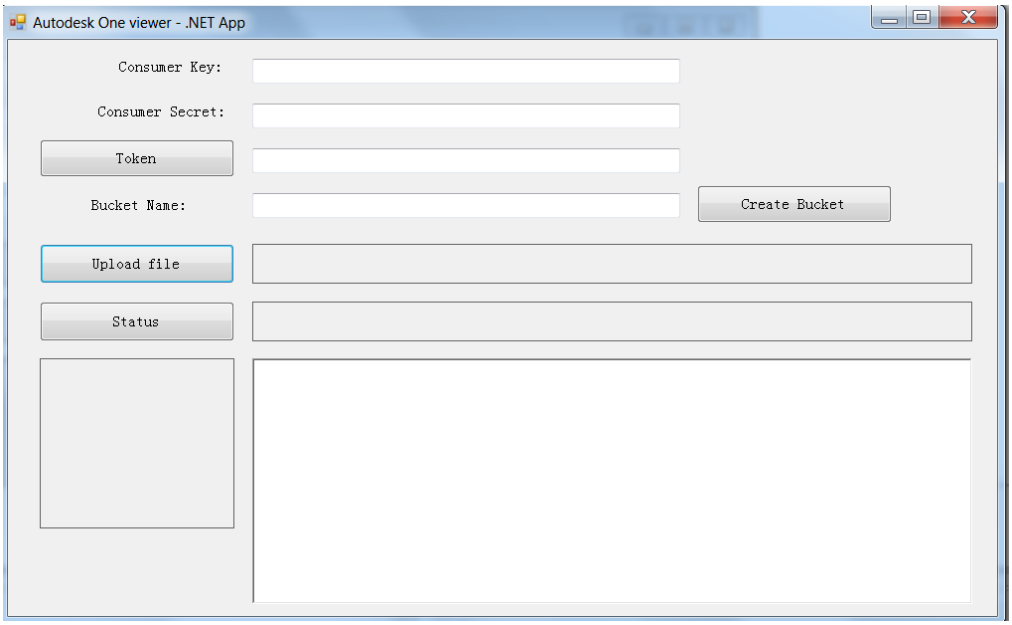
Prepare models with utility

To prepare viewable models, you need to upload and register your model with View and Data API, please refer to the [code samples] on github.com for the workflow. For simplicity of this exercise, you can use utility to do this.

1. Download utility tool from
   https://github.com/Developer-Autodesk/workflow-dotnet-winform-view.and.data.api/
   [Clone in Desktop]: if you have Github client on your computer
   [Download Zip]: if you do not have Github client. This allows you to download the project zip directly.



2. Build and run it

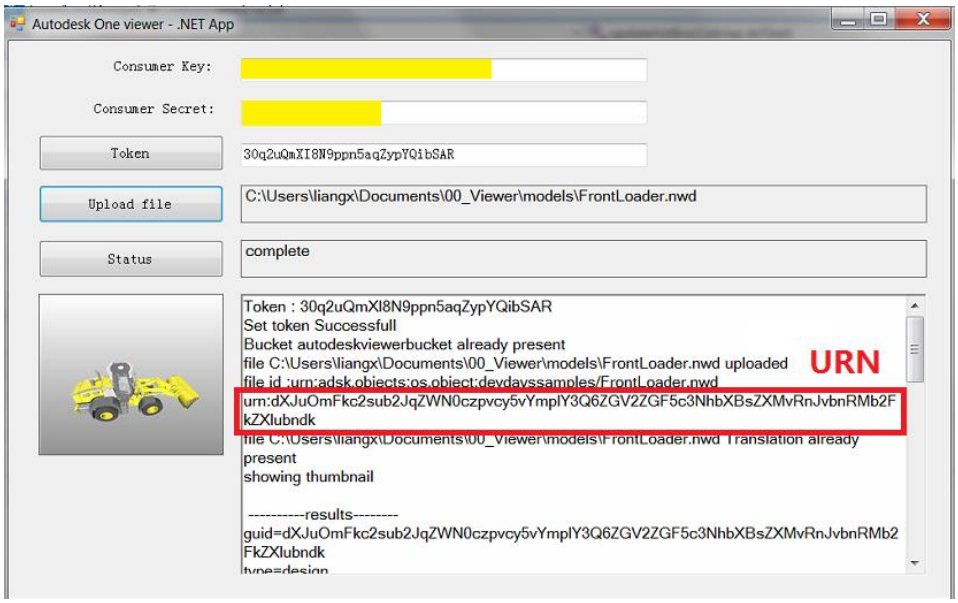


a) Input your own consumer key/secret key in the corresponding textboxes of the dialog
b) Click [Token] to get access token
c) Input a name for your bucket, it should be lower case characters and numbers, click [Create Bucket],
d) Click [Upload File] to select a local file
e) The log will appear in the status window. It will contain the URN of the uploaded file

Note： the URN should start with "urn:"

f) Click [Status] if you want to check the translating status.

g) If everything works well, the thumbnail will appear.



TASK 1 – complete the basic web application with viewer

1. Open the web application **FirstViewerWebApp.sln**
2. Go to **default.aspx**, examine the page layout

a) Add style sheet of “view and data web service” in <head> </head> tag:

```
<!-- viewer-->
<link type="text/css"
       rel="stylesheet"
       href=https://viewing.api.autodesk.com/viewingservice/v1/viewers/style.css
/>
```

b) Add JavaScript library, just before </body> :

```
<script type="text/javascript"
      src="https://developer.api.autodesk.com/viewingservice/v1/viewers/viewer3D.min.js">
      </script>
    <script src="Scripts/Viewer.js">
    </script>
```

3. Go to **/Scripts/Viewer.js**

h) Add following JavaScript in the function **Initialize**:

```
            var urn = $('#urn').val();
            if (urn =='') { //default urn, it can be will be the string you got in the step [prepare
viewable model]
            urn = 'urn:dXJuOmFkc2sub2JqZWN0czpvcy5vYmplY3Q6ZGFuaWVsX3RyYW5zbGF0ZV9idWNrZXQzL0R
                  yaWxsLmR3Zng=';
              }
           if (urn.substr(0,4) !=='urn:') {
            urn = 'urn:' + urn;
           }

        var options = {
            'document': urn,
            'getAccessToken': getToken,
            'refreshToken': getToken,
        };

       // get the canvas of the viewer.
        var viewerElement = document.getElementById('viewer');

        // initialize the viewer.
        viewer = new Autodesk.Viewing.Viewer3D(viewerElement, {});

        Autodesk.Viewing.Initializer(options, function () {
            viewer.start();
            loadDocument(viewer, options.document);
        });
```

i) Add more auxiliary JavaScript functions

```
    function loadDocument(viewer, documentId) {
        // Find the first 3d geometry and load that.
        Autodesk.Viewing.Document.load(documentId, function (doc) {// onLoadCallback
            var geometryItems = [];
            geometryItems = Autodesk.Viewing.Document.getSubItemsWithProperties(
                doc.getRootItem(), {
                'type': 'geometry',
                'role': '3d'}, true);

            if (geometryItems.length > 0) {
                viewer.load(doc.getViewablePath(geometryItems[0]));
            }
        }, function (errorMsg) {// onErrorCallback
            alert("Load Error: " + errorMsg);
        });
```

```
    }

    // This method returns a valid access token
    function getToken() {
        // This method should fetch a token from a service you create to provide authenticatio
n.
        // See the ADN Samples for examples of how to create such a service.  For example, see
        // https://github.com/Developer-Autodesk/workflow-aspnet-webform-view.and.data.api/blo
b/master/ViewAndShare/ViewAndShare/GetAccessToken.ashx.cs
        // This method might look something like:
        var xmlHttp = null;
        xmlHttp = new XMLHttpRequest();
        xmlHttp.open("GET", "GetAccessToken.aspx", false);
        xmlHttp.send(null);
        var res = xmlHttp.responseText;
        var newToken = JSON.parse(res);

        if (newToken.error) {
            console.log(newToken.error);
            return '';

        }
        else {
            return newToken.access_token;
        }

    }
```

4. Go to **Credentials.cs** , replace with your consumer key and secret:

```
        //replace your consumer key
        public static string CONSUMMER_KEY = "your-consumer-key";
        public static string SECRET_KEY = "your-secret-key";
```

5. Launch the web project; fill in your URN (you get it in step- [prepare viewable model]) and click “load model” button to load the viewer
6. Debug and fix any issue.
7. If everything works well, you should see the model is loaded in the viewer.

TASK 2 – Create a simple extension

Target: Create a viewer extension to output the properties of selected element on viewer

1. Add a new Js file in Script folder, name as BasicExtension.js

2. Copy the Extension skeleton snippet:

```
'use strict';

//declare your namespaces
AutodeskNamespace('MyCommpany.Extensions');


MyCommpany.Extensions.BasicExtension = function (viewer) {

    Autodesk.Viewing.Extension.call(this, viewer);

}

MyCommpany.Extensions.BasicExtension.prototype = Object.create(Autodesk.Viewing.Extension.prot
otype);
MyCommpany.Extensions.BasicExtension.prototype.constructor = MyCommpany.Extensions.BasicExtens
ion;

//extension load
MyCommpany.Extensions.BasicExtension.prototype.load = function () {

    console.log('MyCommpany.Extensions.BasicExtension is loaded');
    //get the viewer object
     var oViewer = this.viewer;
     //examine the viewer object in console
    console.dir(oViewer);

    //TODO: Add event listener for ‘selection’

    //TODO: output the properties of selected element

    }

    //extension unload
    MyCommpany.Extensions.BasicExtension.prototype.unload = function () {

    }

    /**
     * Register the extension with the extension manager.
     */
    Autodesk.Viewing.theExtensionManager.registerExtension('BasicExtension', MyCommpany.Extens
    ions.BasicExtension);
```

3. Go to Scripts**viewer.js** , around line 23, change to following snippet, the differences is in bold.

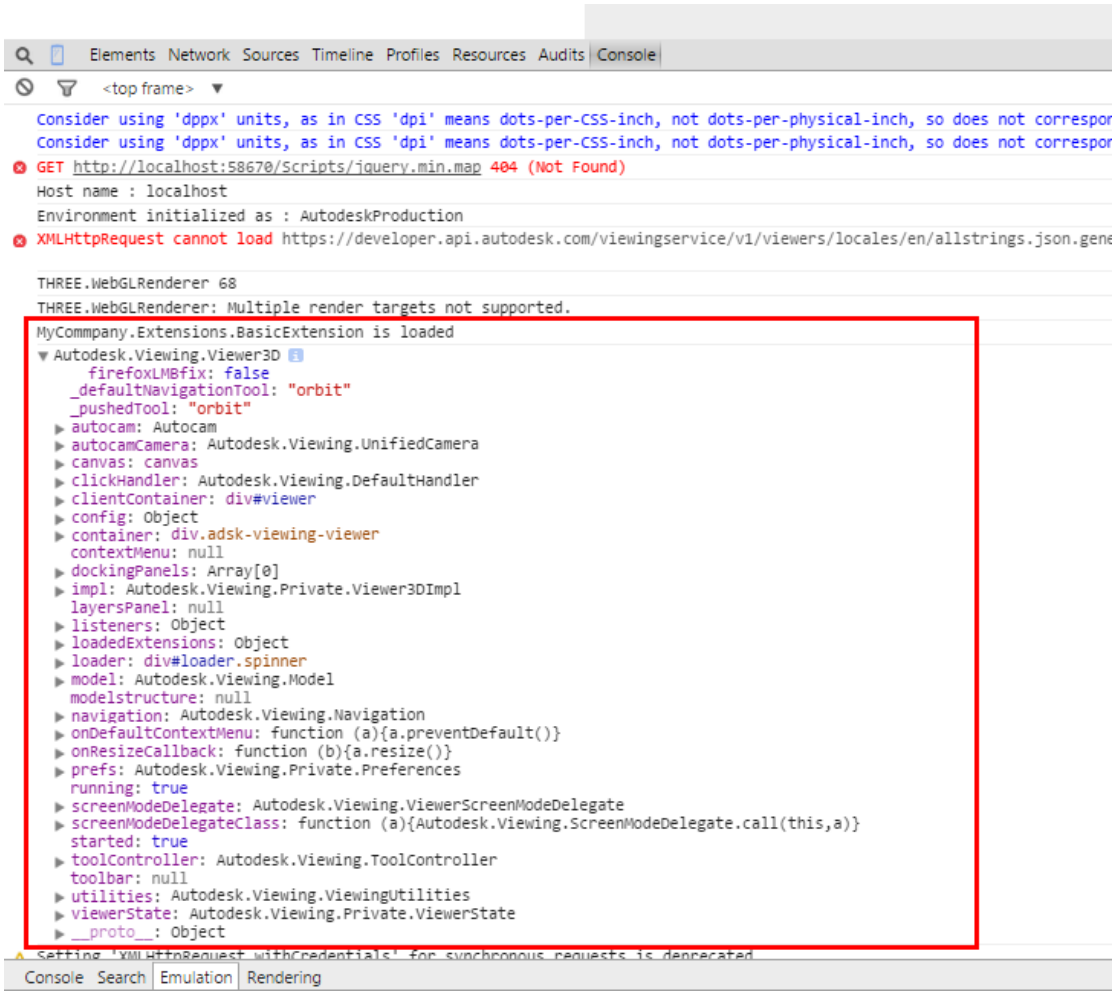
```
viewer = new Autodesk.Viewing.Viewer3D(viewerElement, { extensions: ['BasicExtension'] });
```

4. Go to **Default.aspx**, add following script reference before </body>

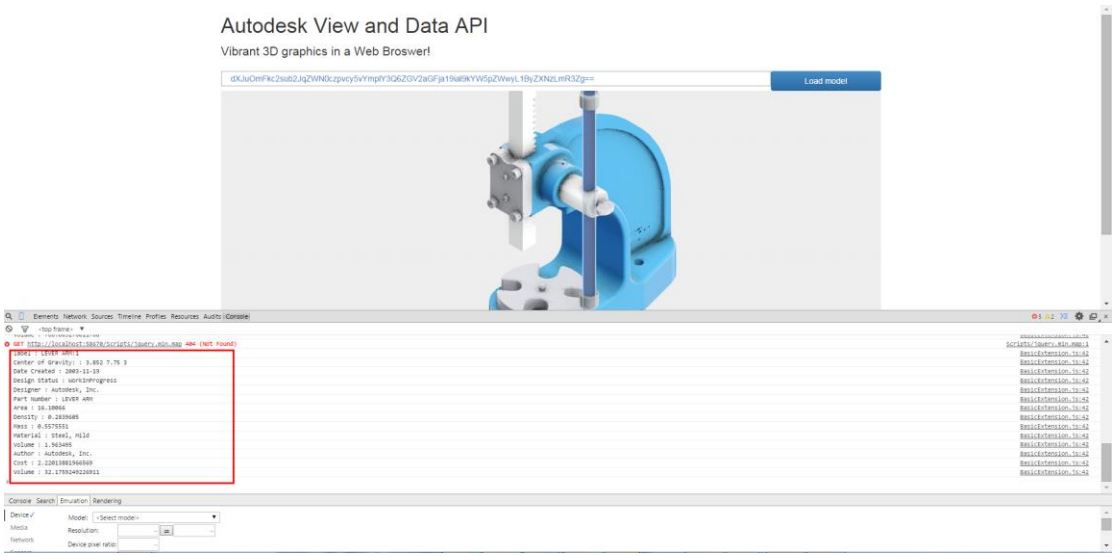
```
<script src="Scripts/BasicExtension.js"></script>
```

5. Debug and fix any issue.

If everything goes OK, go to Developer tools of Chrome, you should see the message is logged: "MyCommpany.Extensions.BasicExtension is loaded"



Challenge: Select one element on model, display the properties in Chrome console:



Solution for reference:

```
    oViewer.addEventListener('selection', function (event) {
        var dbIdArray = event.dbIdArray;

        for (var i = 0; i < dbIdArray.length; i++) {

            var dbId = dbIdArray[i];

            oViewer.getProperties(dbId, function (result) {
                if (result.properties) {

                    for (var i = 0; i < result.properties.length; i++) {

                        var prop = result.properties[i];

                        console.log(prop.displayName + ' : ' + prop.display
Value);

                    }
                }
            });
        }
    });
```

Help Documents:

http://developer-dev.api.autodesk.com/documentation/v1/index.html