EXPLORING THE DYNAMO API WITH C#

Mark Thorley

Director

20th June, 2018

Join the conversation #AUCity #AU2018

AUTODESK. UNIVERSITY

INTRODUCTION
ABOUT ME

@markthorley1

mark.thorley@designtech.io

www.designtech.io

MARK THORLEY
DIRECTOR

- Previously worked at AECOM, Grimshaw, Spacezero in BIM Manager roles
- Passionate about computer science and its application in the field of Architecture and Construction
- Co-Founded designtech in 2016
- Specialise in Computational Workflows and Automation
- Work alongside some leading organisations including Heatherwick, Landsec, Arup, Mott MacDonald, NatWest and Nissan

About

designtech.io

At designtech we create custom workflows and software to enable businesses to develop their designs and manage their data while cutting down on costly inefficiencies and errors.

INTRODUCTION
IS THIS CLASS FOR ME?

New Horizons Beckon

Interested in stepping up your Dynamo game?

API Confusion

Heard the term API but unsure about what is actually means?

API Structure

Keen to learn how API's are structured?

C# Fundamentals

Want to grasp some basic C# fundamentals?

Personal Growth

Are you simply curious?

Free Goodies

Everyone loves a freebie…

INTRODUCTION

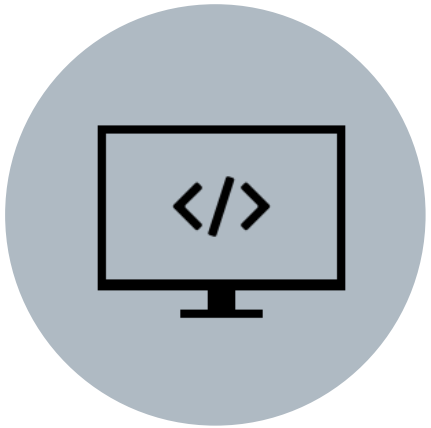
KEY LEARNING OBJECTIVES

1.

Learn what an ‘API’ is

2.

Learn how the Dynamo API is structured.



3.

Learn about the basic principles of C#.

4.

Learn about real use cases and applications

.

What is an API?

WHAT IS AN API?

NOT ANOTHER ACRONYM

Application Programming Interface

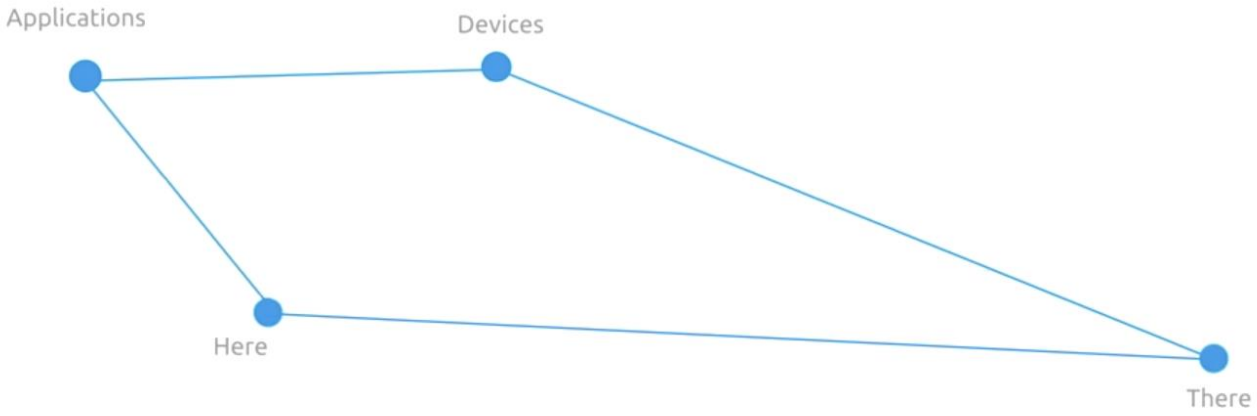
WHAT IS AN API?
EXPLICIT MEANING

'In computer programming, an application programming interface (API) is a set of subroutine definitions, protocols, and tools for building application software.'

WHAT IS AN API?
IN LAYMAN'S TERMS

'Basically an API is a set of commands that developers can use, alongside a description of what they do and how to use them.'

WHAT IS AN API?
WHY ARE THEY IMPORTANT



Communication

They allow the devices and applications to talk to one another.

Transfer

Allows the transfer of data between different systems

Leverage

Leverage others technologies for use in building applications.

Simplify

By only exposing particular objects and methods, code bases are simplified for developers.

WHAT IS AN API?
BENEFITS

Automation

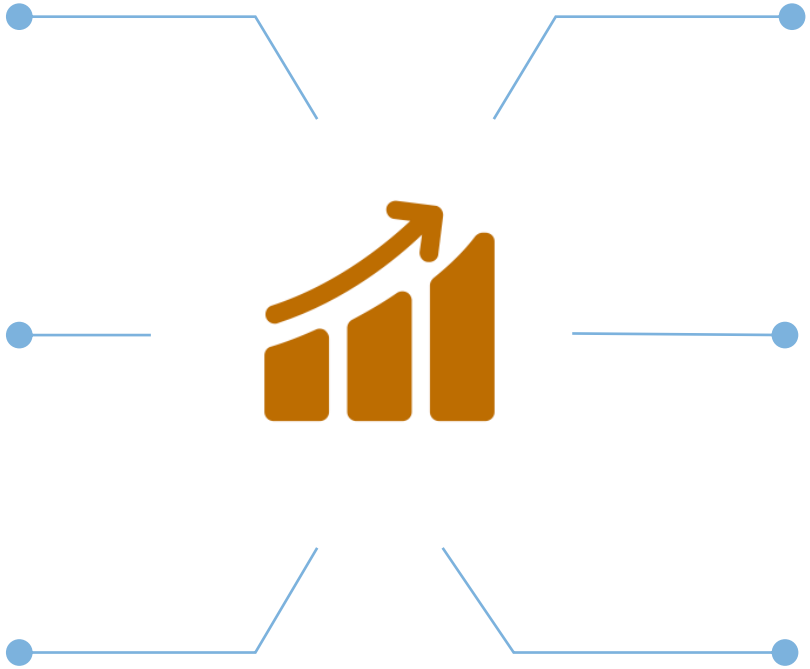
Computers rather than people manage the work making processes more efficient.

Integration

Content can be embedded from any application providing a smoother transfer of information.

Availability

API's allow for information to be readily available to the masses not just a select few..



Efficiency

Content can be shared and distributed easily.

Adaptation

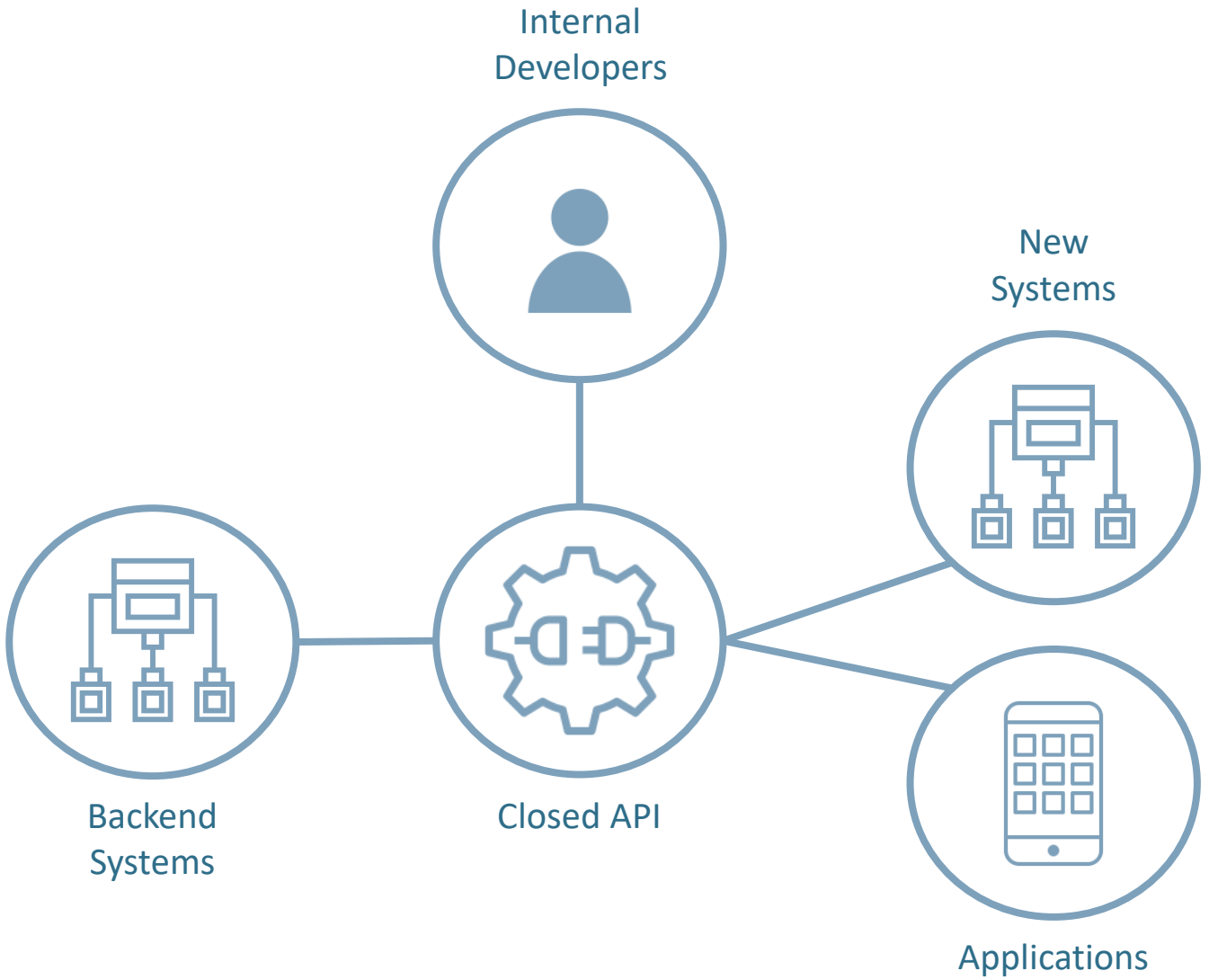
Changes over time are easier to anticipate and make.

Modular Programming

An API attempts to provide only tools a user would be expected to use

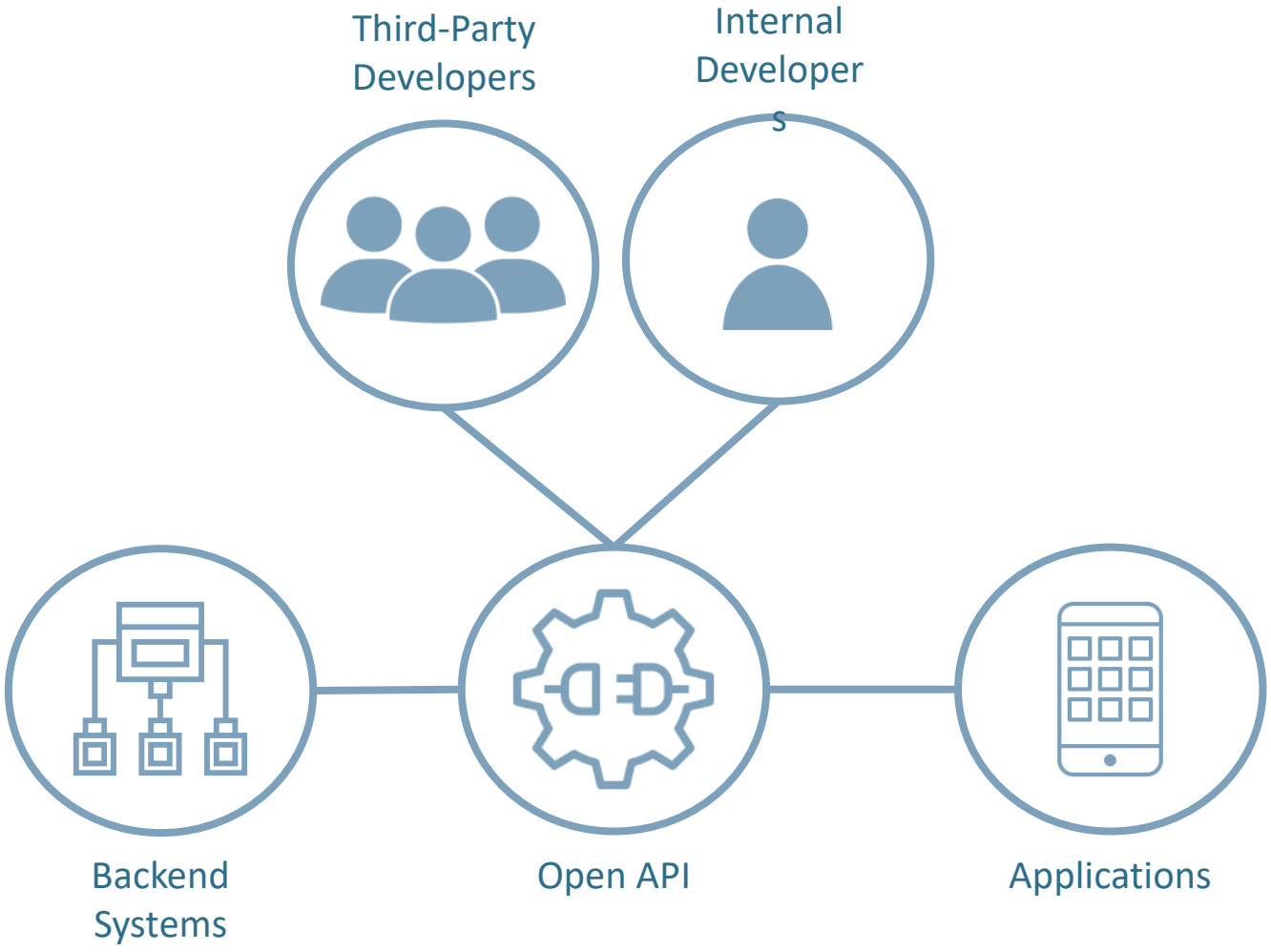
WHAT IS AN API?

CLOSED API

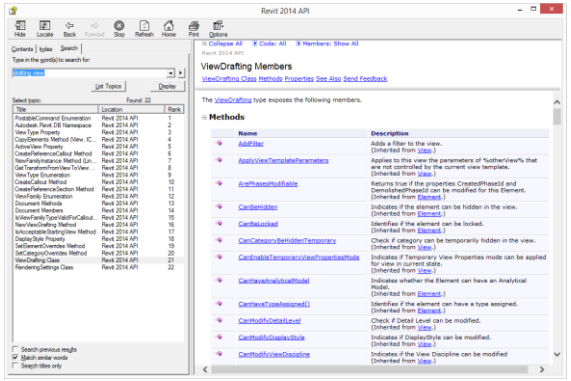


WHAT IS AN API?

OPEN API



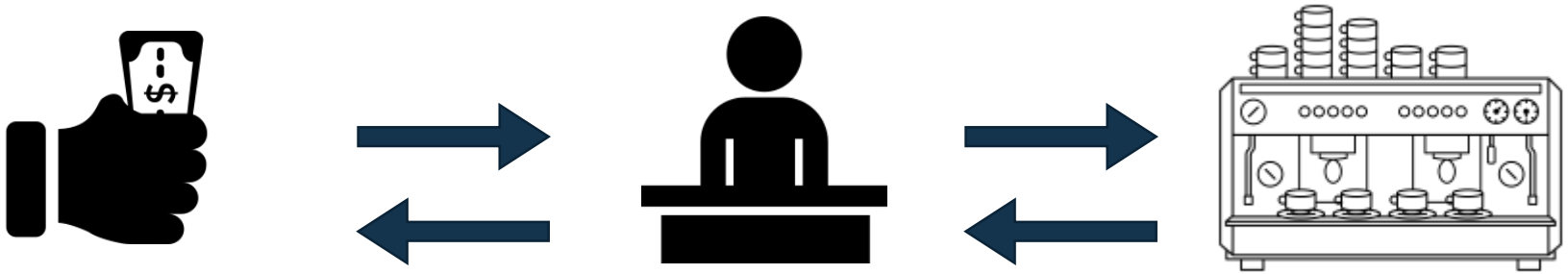
WHAT IS AN API?
DON’T GET CONFUSED



EXAMPLES OF API’S

EXAMPLES OF API'S

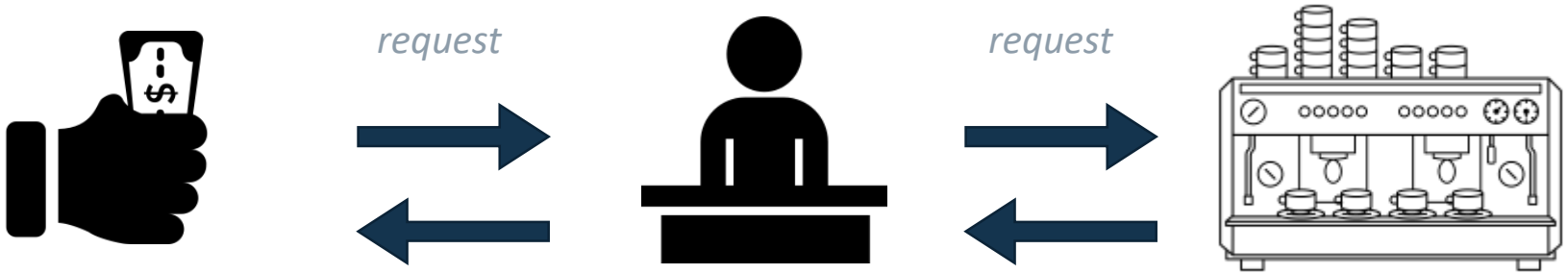
EVERYDAY LIFE



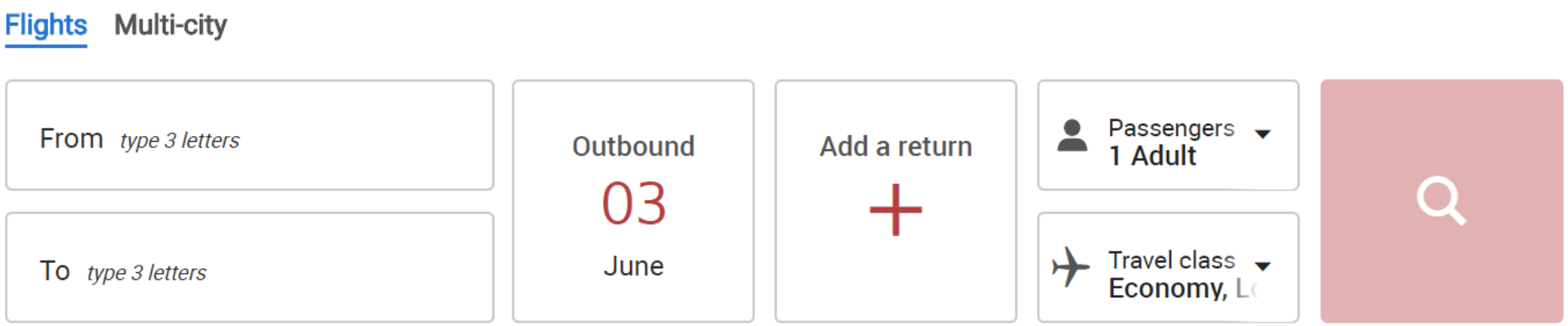
Customer Cashier Barista

EXAMPLES OF API'S

EVERYDAY LIFE



EXAMPLES OF API'S
BOOKING A FLIGHT



BRITISH AIRWAYS

EXAMPLES OF API'S

SKYSCANNER

API'S IN C# & .NET

BASIC PRINCIPLES

API'S IN C# & .NET

WHY C#

API'S IN C# & .NET

STRUCTURE

- Namespaces
- Types
- Members

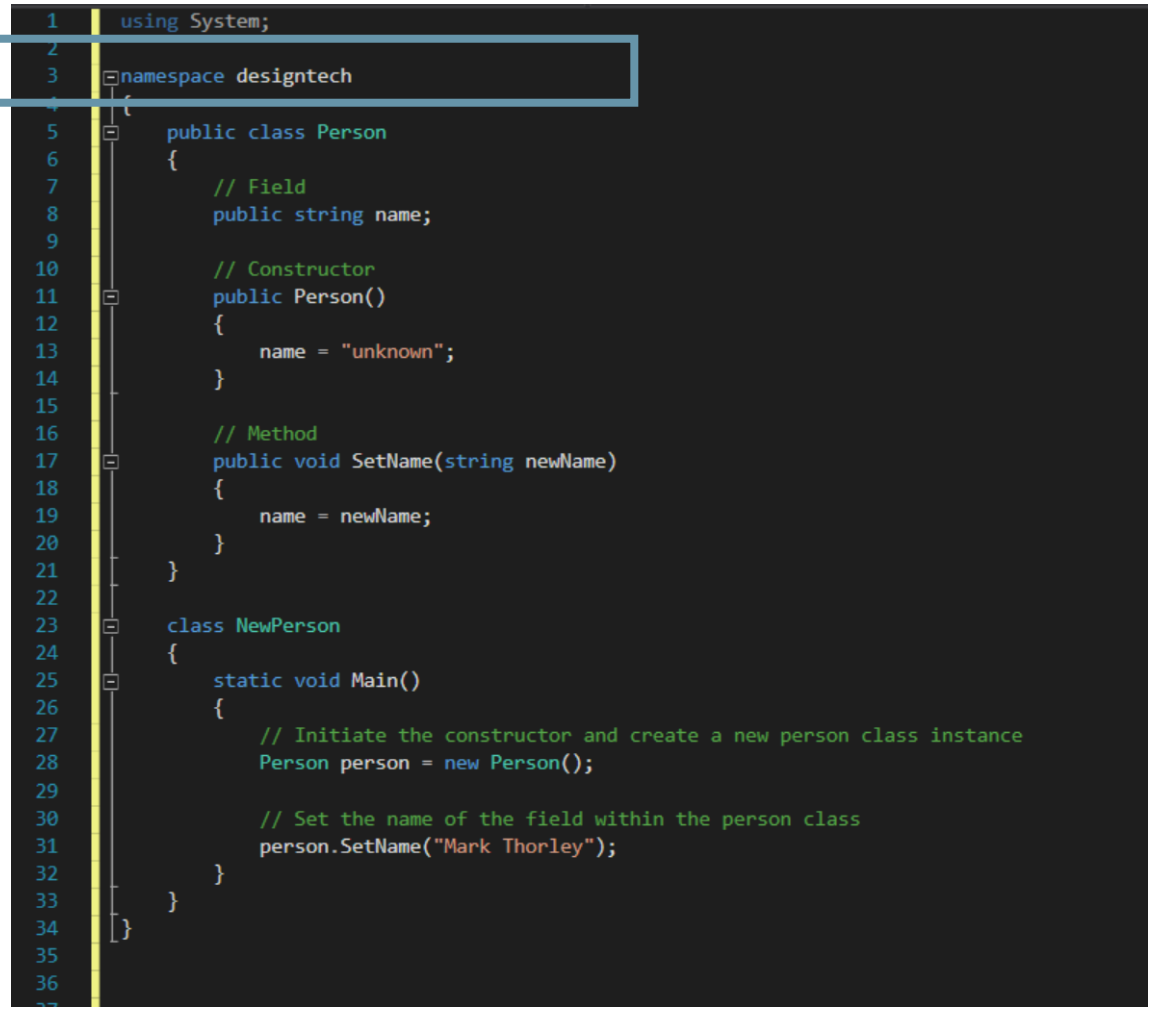
```
using System;

namespace designtech
{
    public class Person
    {
        // Field
        public string name;

        // Constructor
        public Person()
        {
            name = "unknown";
        }

        // Method
        public void SetName(string newName)
        {
            name = newName;
        }
    }

    class NewPerson
    {
        static void Main()
        {
            // Initiate the constructor and create a new person class instance
            Person person = new Person();

            // Set the name of the field within the person class
            person.SetName("Mark Thorley");
        }
    }
}
```

API'S IN C# & .NET

NAMESPACES

- **Namespaces**
- Types
- Members

Container to hold a group of unique identifiers
eg. Autodesk.Revit.DB



```
using System;

namespace designtech
{
    public class Person
    {
        // Field
        public string name;

        // Constructor
        public Person()
        {
            name = "unknown";
        }

        // Method
        public void SetName(string newName)
        {
            name = newName;
        }
    }

    class NewPerson
    {
        static void Main()
        {
            // Initiate the constructor and create a new person class instance
            Person person = new Person();

            // Set the name of the field within the person class
            person.SetName("Mark Thorley");
        }
    }
}
```

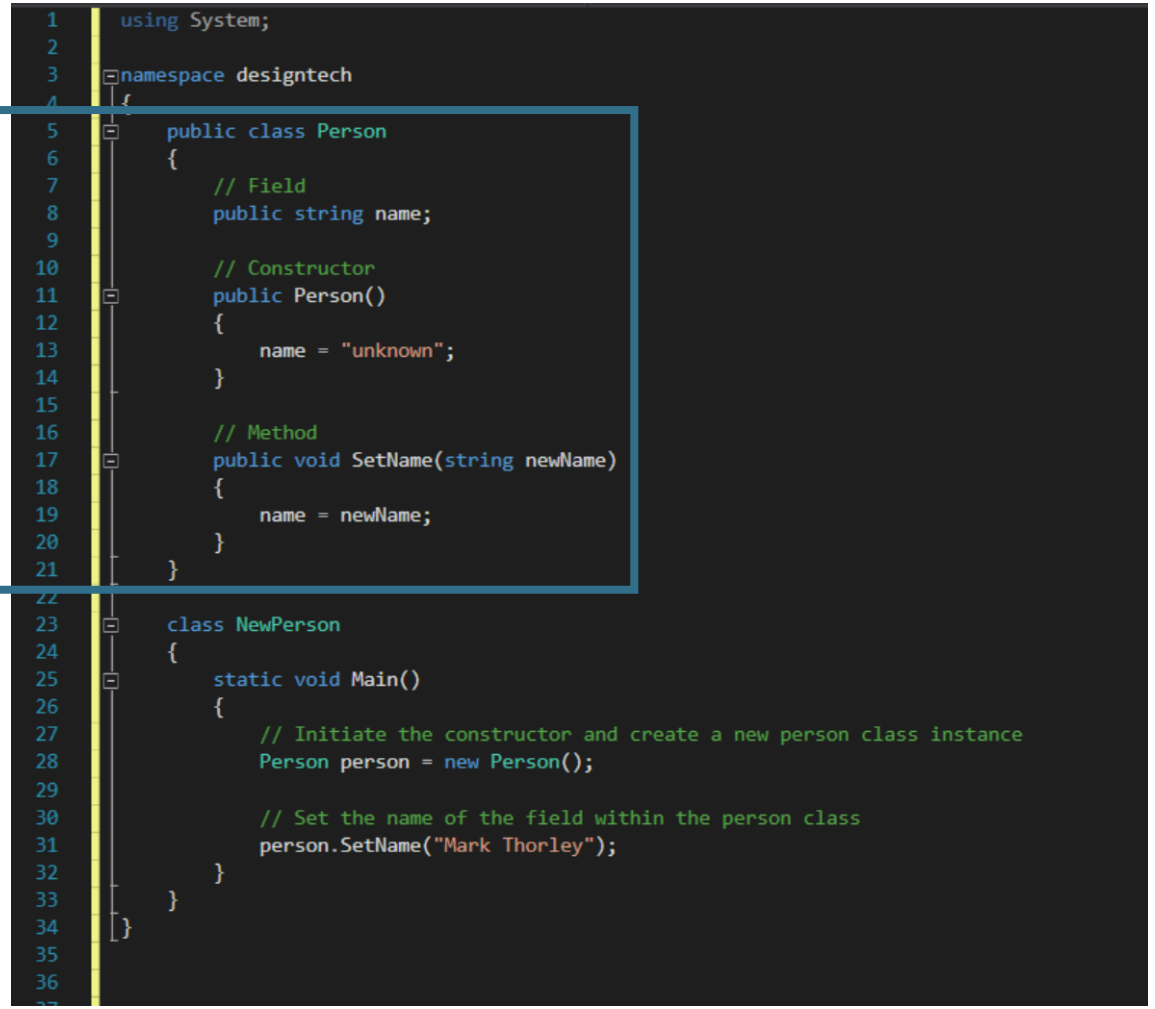
API'S IN C# & .NET

TYPES

- Namespaces
- **Types**
- Members

Includes:

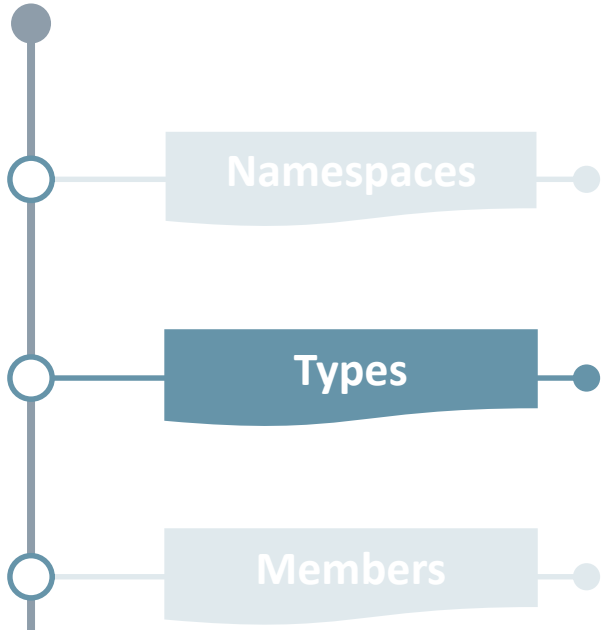
- Classes
- Structures
- Interfaces
- Enum's
- Delegates



```
using System;

namespace designtech
{
    public class Person
    {
        // Field
        public string name;

        // Constructor
        public Person()
        {
            name = "unknown";
        }

        // Method
        public void SetName(string newName)
        {
            name = newName;
        }
    }

    class NewPerson
    {
        static void Main()
        {
            // Initiate the constructor and create a new person class instance
            Person person = new Person();

            // Set the name of the field within the person class
            person.SetName("Mark Thorley");
        }
    }
}
```

API'S IN C# & .NET

CLASSES



Includes:

- Classes
- Structures
- Interfaces
- Enum's
- Delegates

Classes:

- The most fundamental of .Net types
- A data structure that combines members in a single unit
- Provides a definition for dynamically created instances of the class, also known as objects

API'S IN C# & .NET

MEMBERS

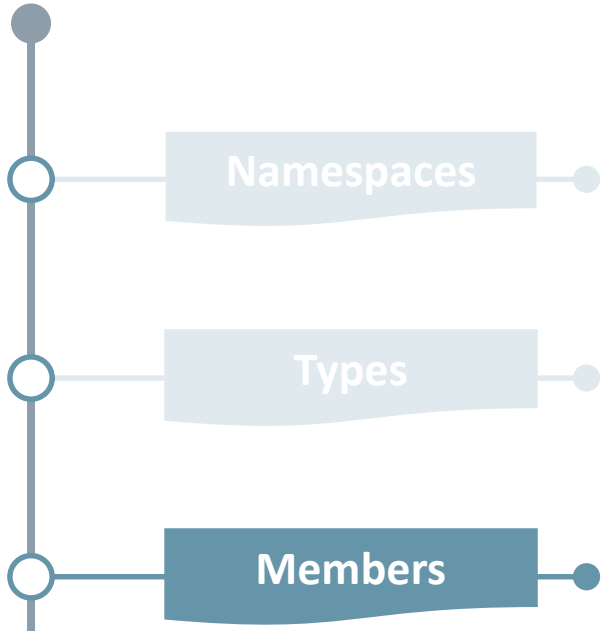
- Namespaces
- Types
- **Members**

Includes:

- Constants
- Fields
- Methods
- Constructors
- Properties
- Events

```
using System;

namespace designtech
{
    public class Person
    {
        // Field
        public string name;

        // Constructor
        public Person()
        {
            name = "unknown";
        }

        // Method
        public void SetName(string newName)
        {
            name = newName;
        }
    }

    class NewPerson
    {
        static void Main()
        {
            // Initiate the constructor and create a new person class instance
            Person person = new Person();

            // Set the name of the field within the person class
            person.SetName("Mark Thorley");
        }
    }
}
```

API'S IN C# & .NET
METHODS/CONSTRUCTORS/PROPERTIES



Includes:

- Constants
- Fields
- Methods
- Constructors
- Properties
- Events

Methods:

- Actions that can be performed by the class

Constructors:

- Methods that are called when the object is first created. They are often used to initialize the data of an object.

Properties:

- Actions associated with reading and writing named properties of the class

DYNAMO API

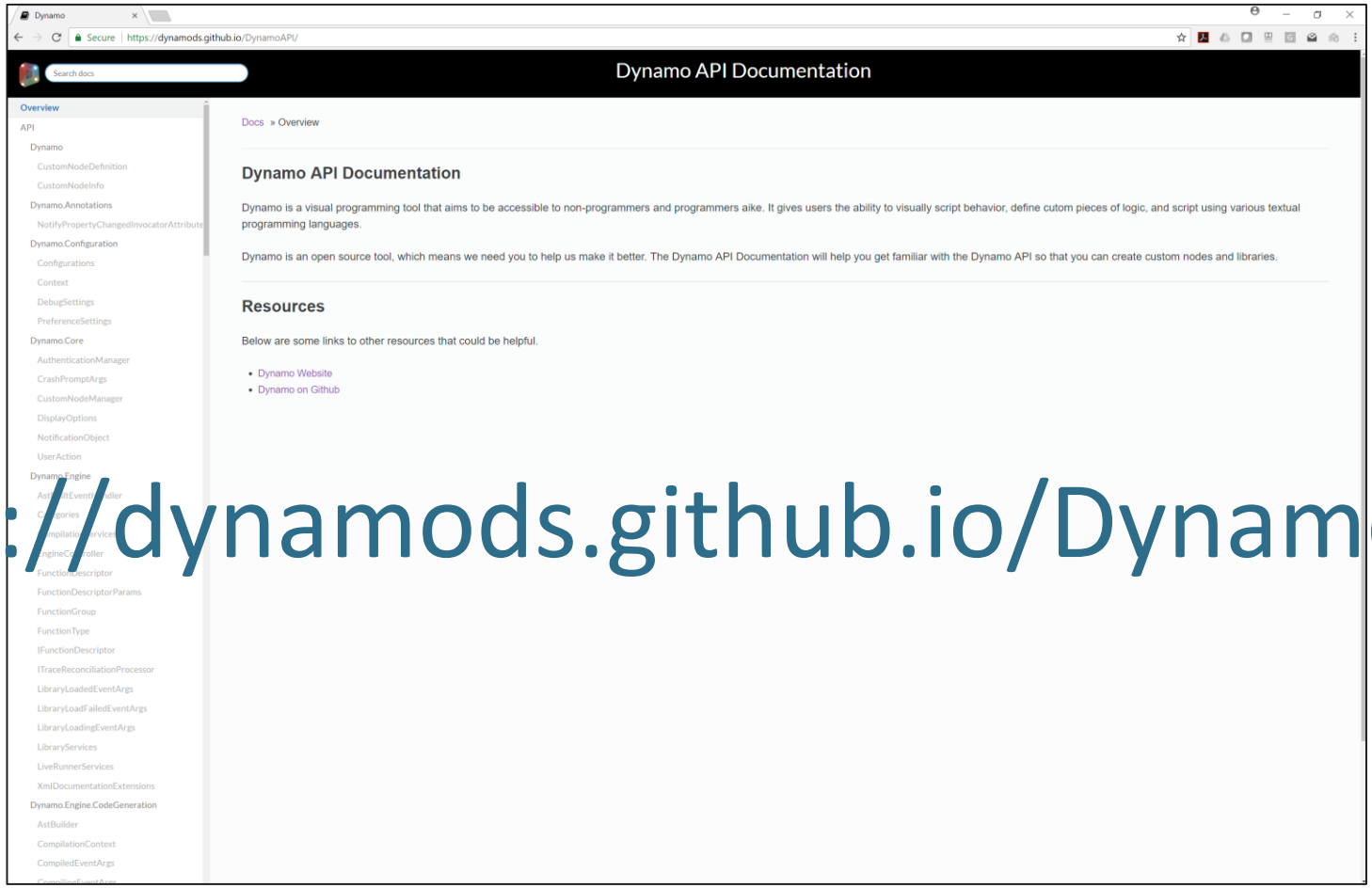
GETTING STUCK IN

DYNAMO API
WHAT IS IT?

- Provides an interface to talk with Dynamo
- Allows you create custom nodes and libraries
- Customise the interface
- Build extensions

Dynamo

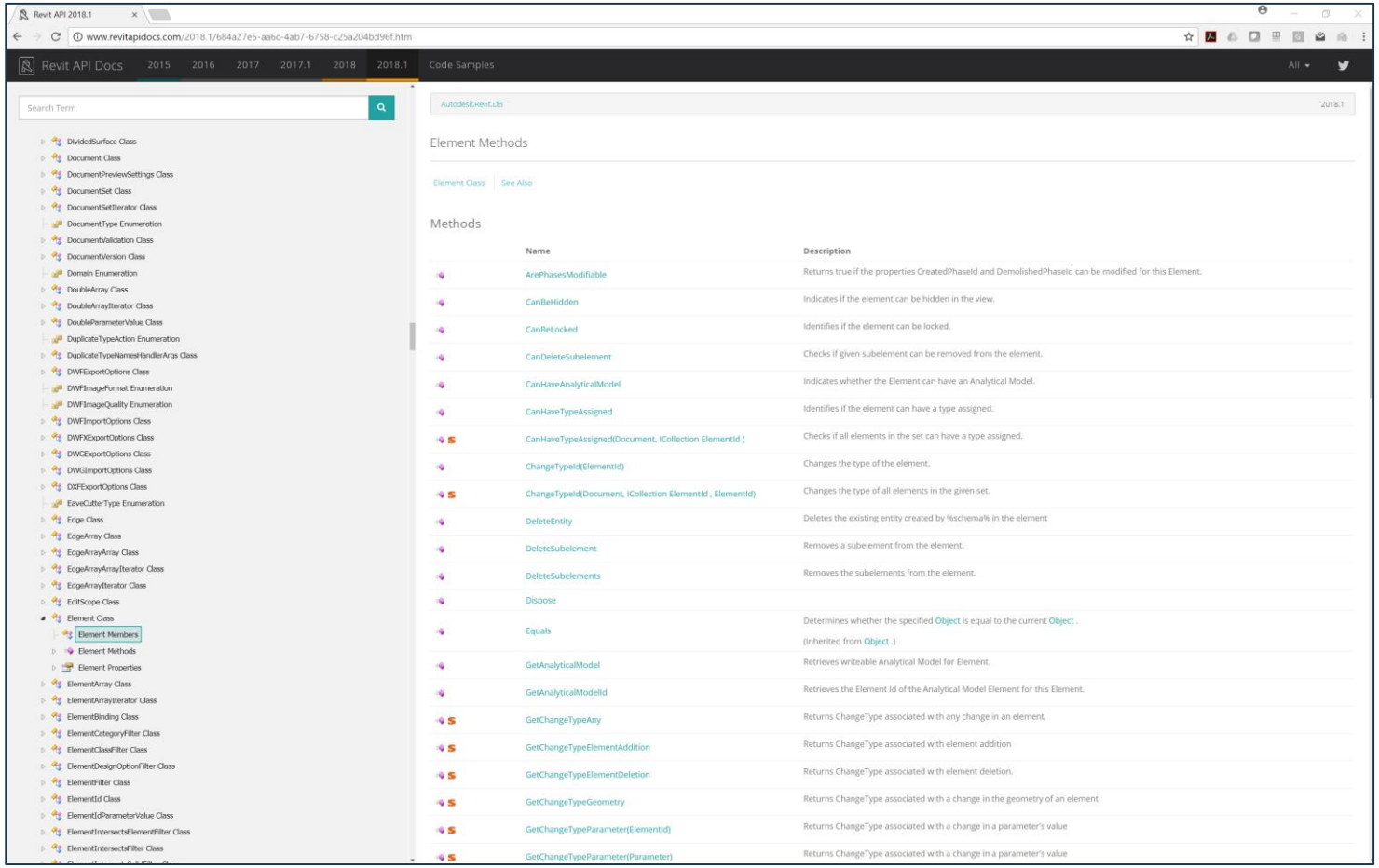
DYNAMO API
DOCUMENTATION



https://dynamods.github.io/DynamoAPI/

DYNAMO API

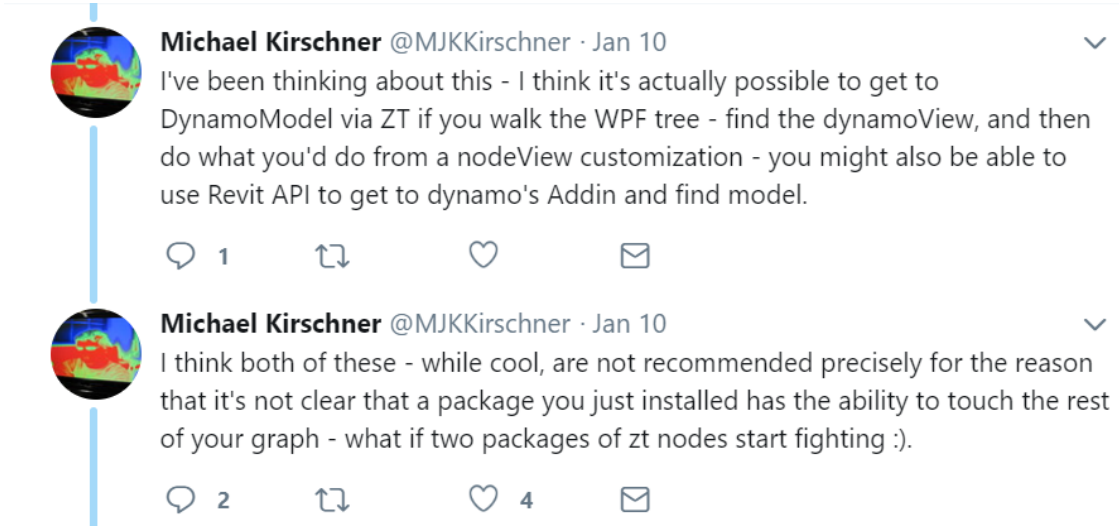
DYNAMOAPIDOCS.COM



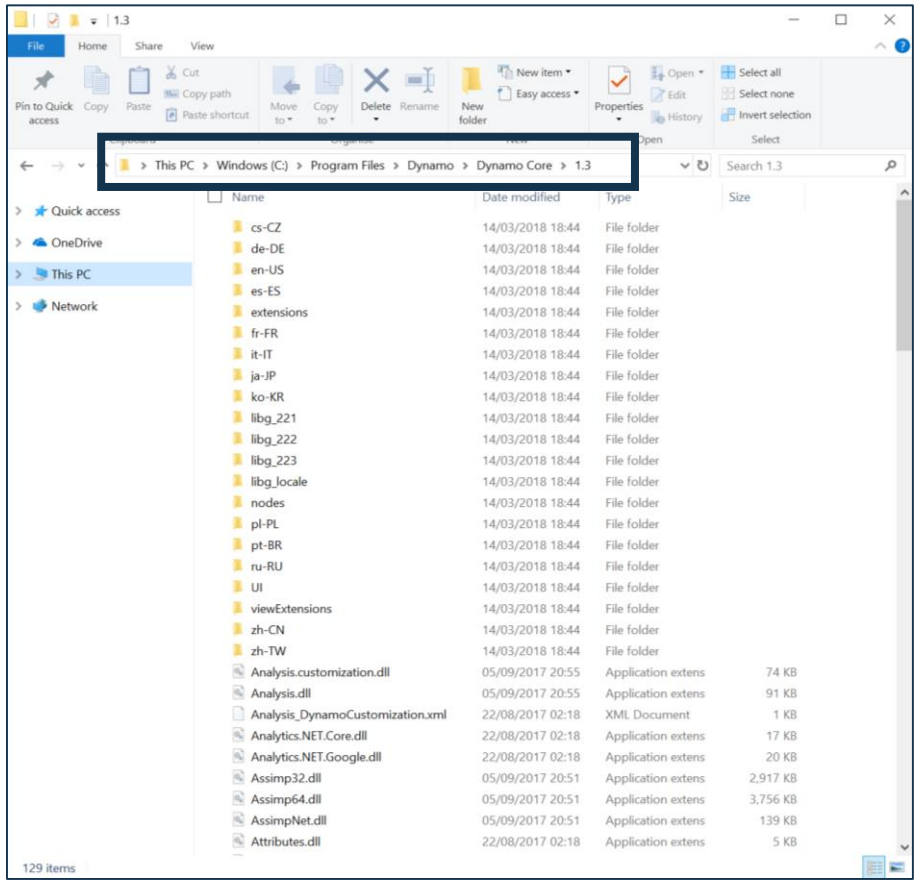
- RevitApiDocs.com
- Web-based, searchable, and extensible Revit API documentation
- Huge kudos to **Gui Talarico** for his work on this
- Dynamo:
  - content needs an upheaval
  - friendlier UI required
  - many items missing descriptions and lacking content

DYNAMO API

CAUTION



DYNAMO API
FILE LOCATION

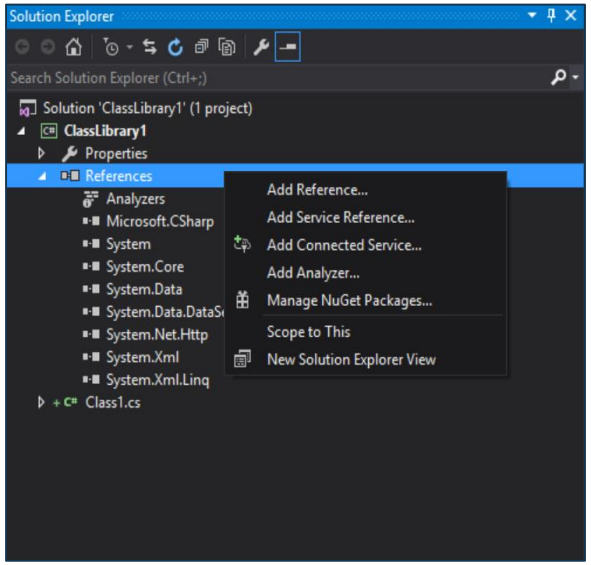


C:\Program Files\Dynamo\Dynamo Core\1.3

DynamoCore.dll

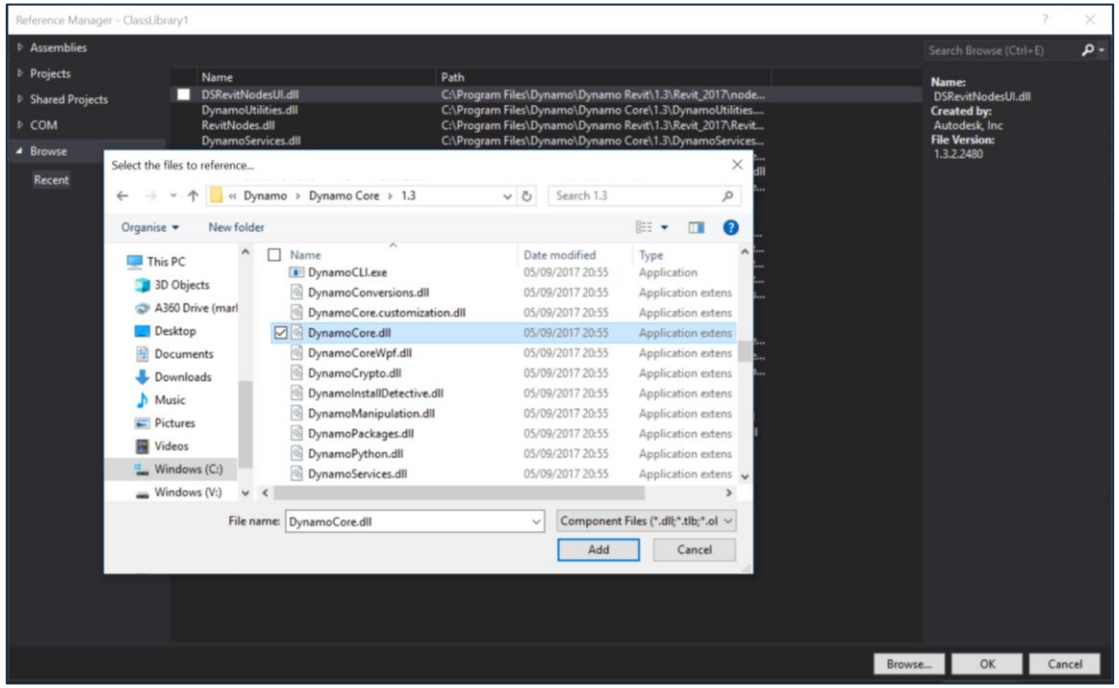
DYNAMO API

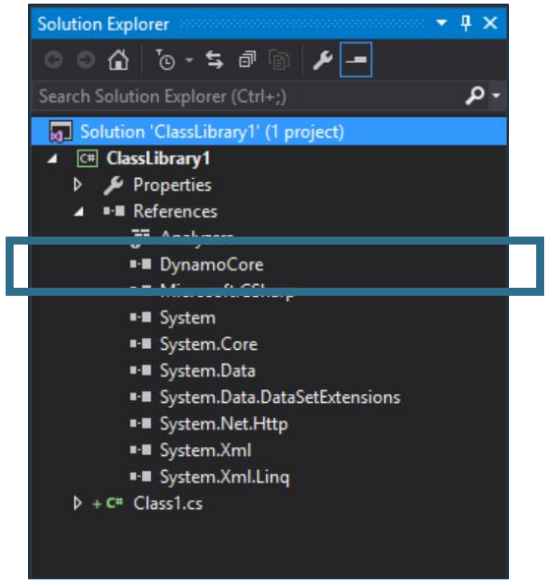
IMPORT LIBRARY INTO VISUAL STUDIO



1. Open up your solution and right click on References in the Solution Explorer and select Add Reference.

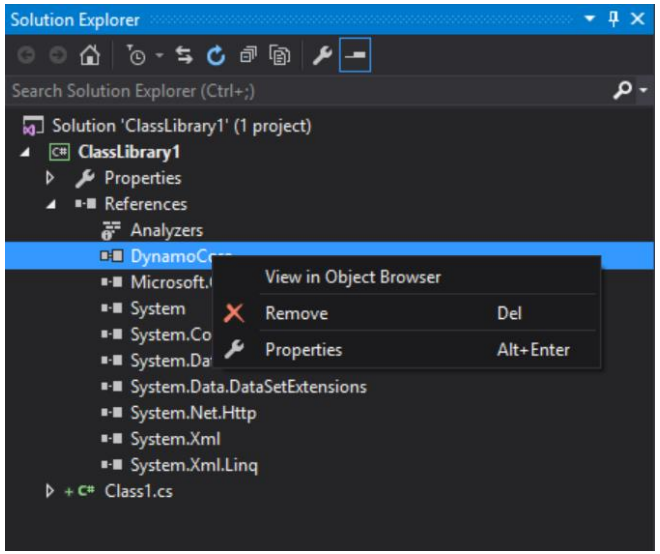
2. Click Browse, navigate to the location path on the previous slide and click Add. In the next window select OK.

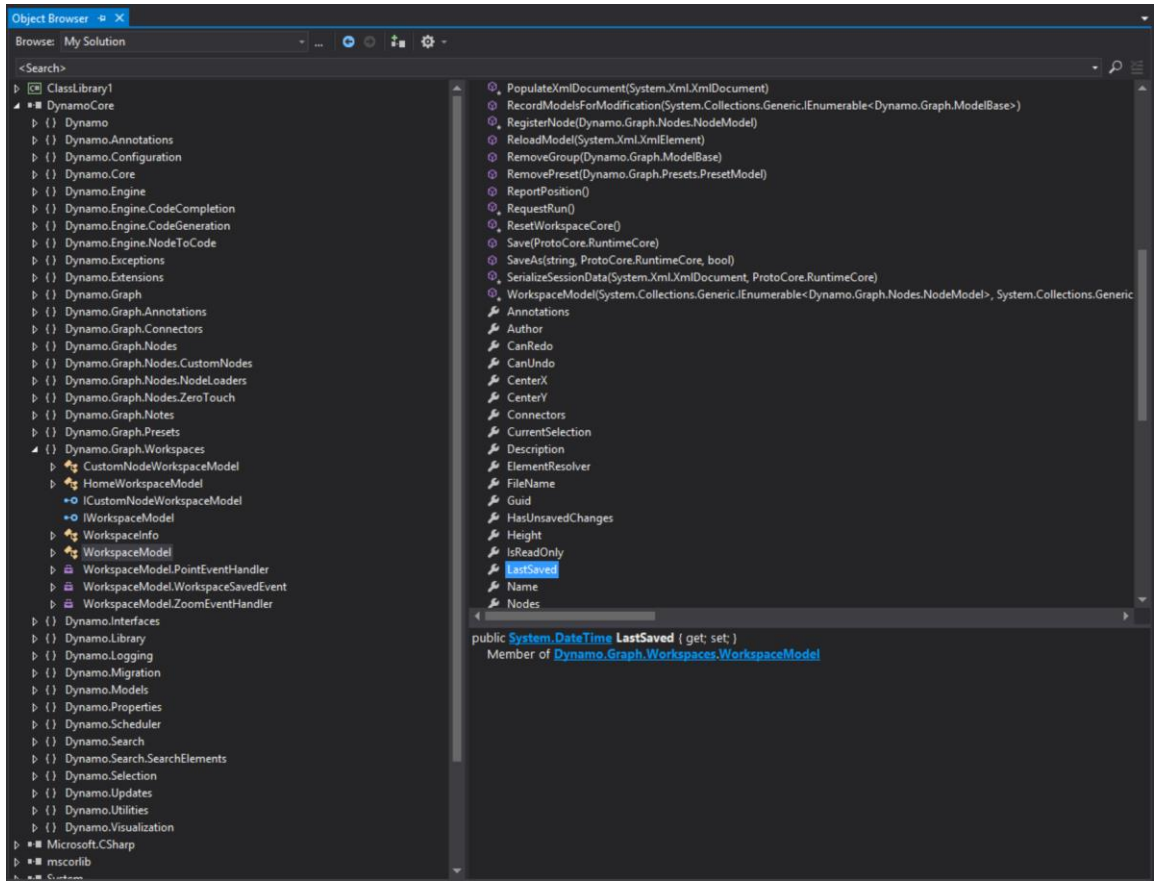




3. Once the Reference is added it should be visible in the Solution Explorer.

DYNAMO API
EXPLORING THROUGH OBJECT BROWSER





In the Solution Explorer under References right click on the DynamoCore item and select View in Object Browser

DYNAMO API

CORE LIBRARIES OVERVIEW

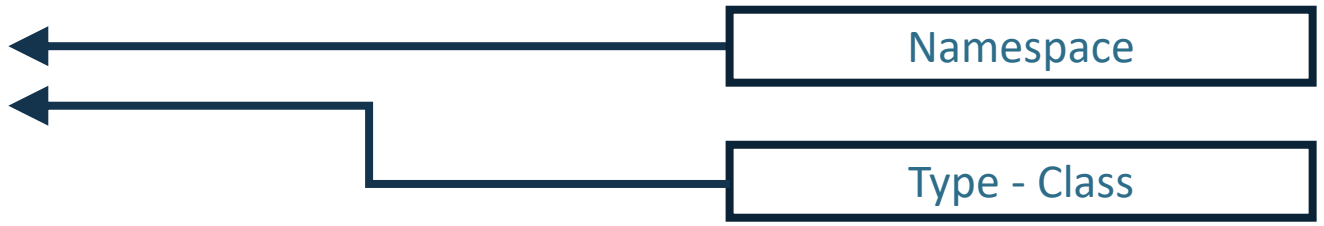
{} Dynamo.Graph
- NodeGraph

{} Dynamo.Graph.Workspaces
- WorkspaceModel

{} Dynamo.Graph.Nodes
- NodeModel

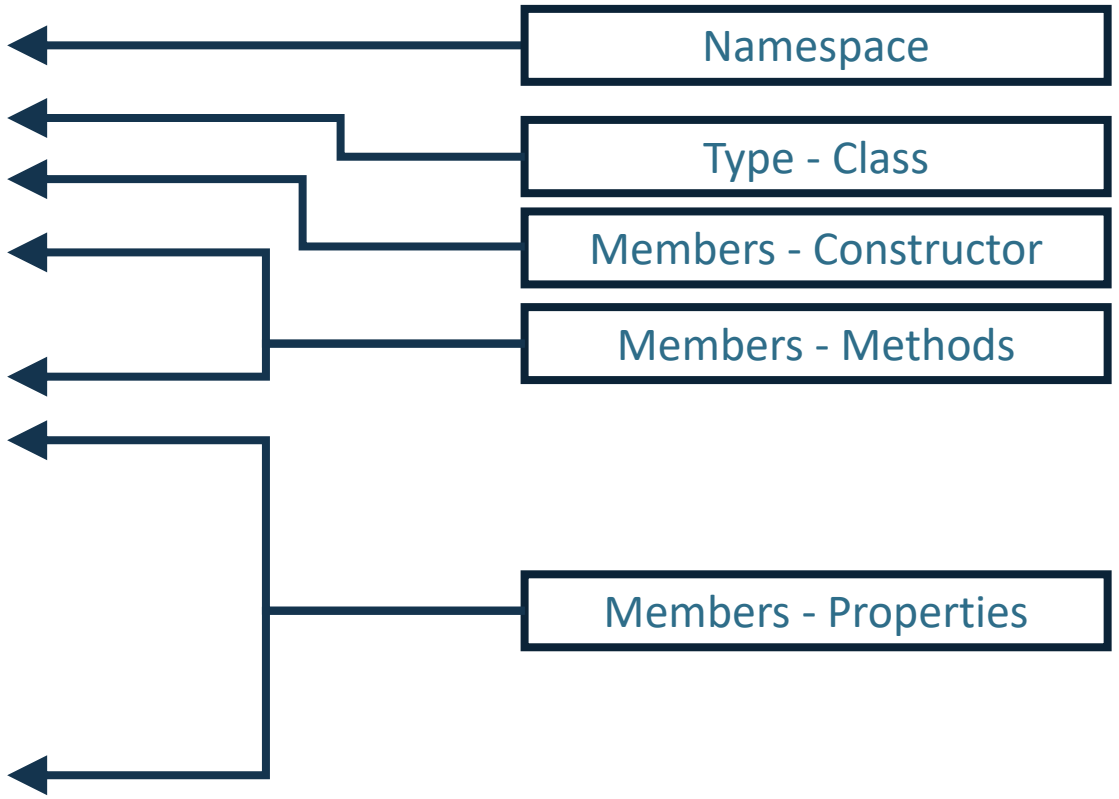
{} Dynamo.Graph.Notes
- NoteModel

{} Dynamo.Graph.Annotations
- AnnotationModel



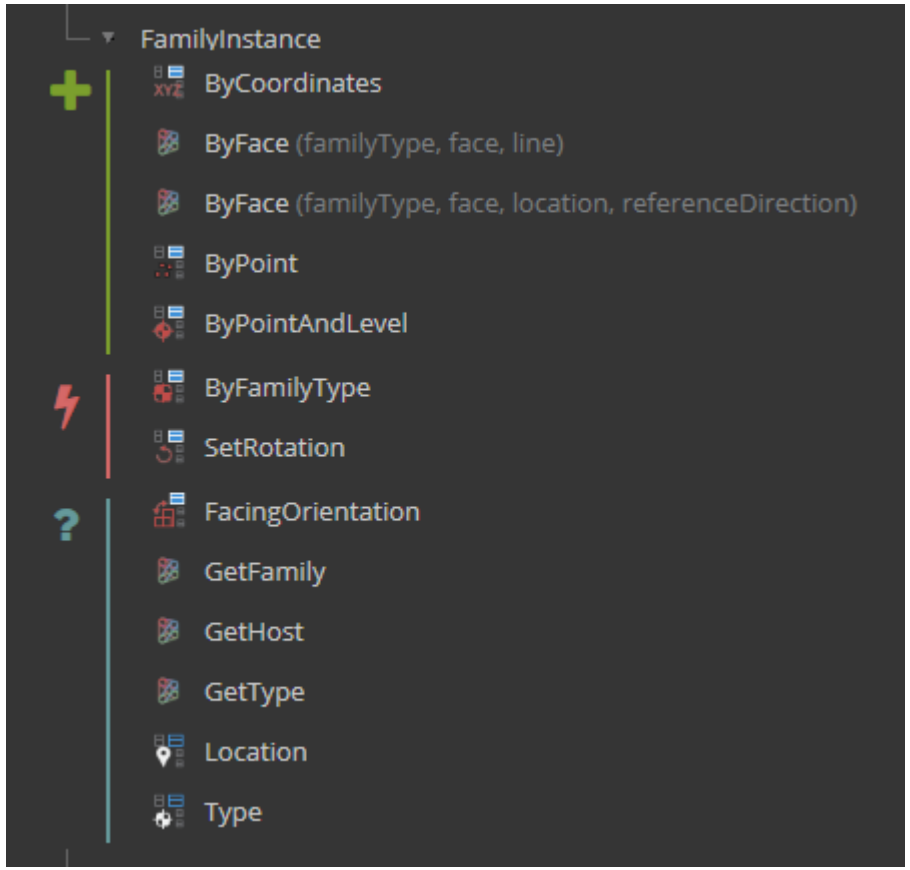
DYNAMO API
CORE LIBRARIES BREAKDOWN

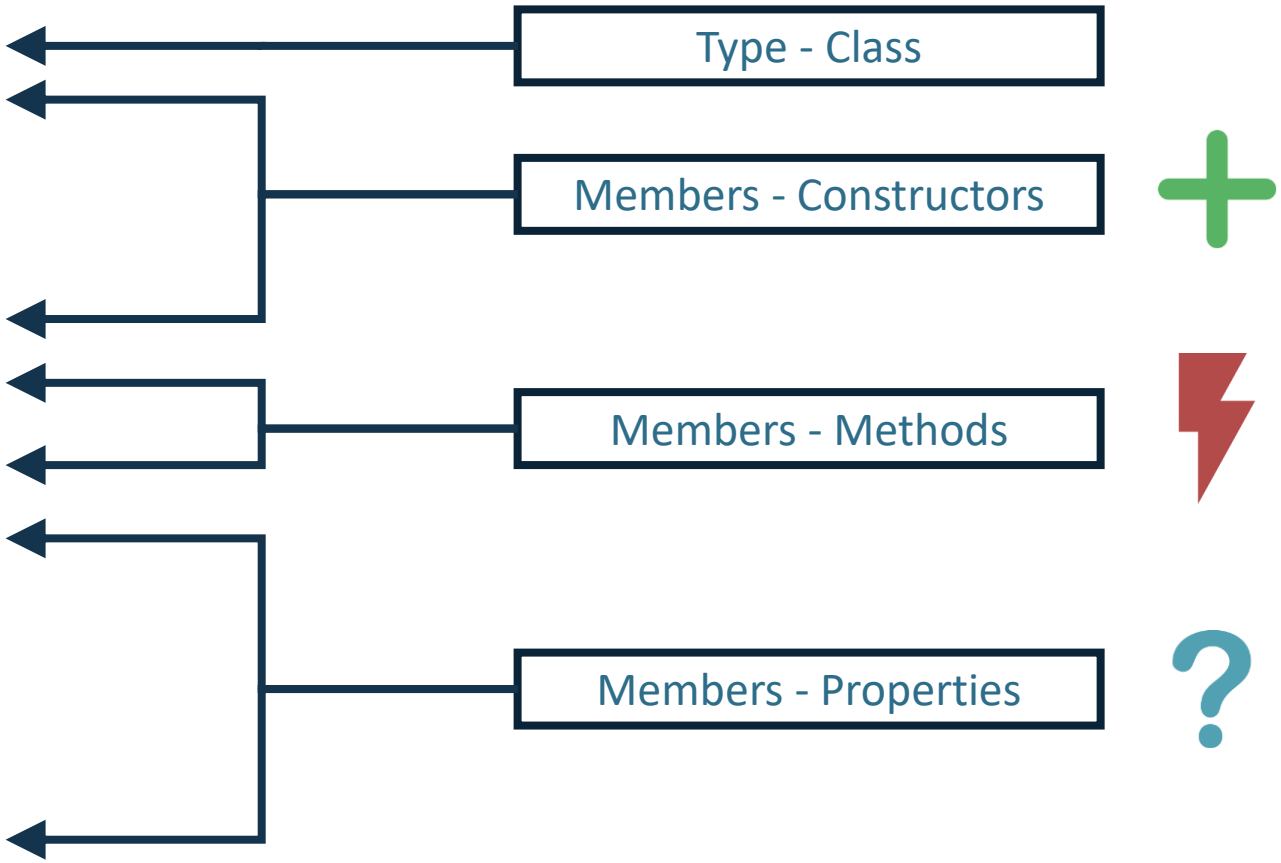
{} Dynamo.Graph
- NodeGraph
  - LoadGraphFromXml
  - LoadConnectorFromXml
  - LoadNodeFromXml
  - LoadPresetsFromXml
  - Annotations
  - Connectors
  - ElementResolver
  - Nodes
  - Notes
  - Presets



DYNAMO API

CORE LIBRARIES BREAKDOWN



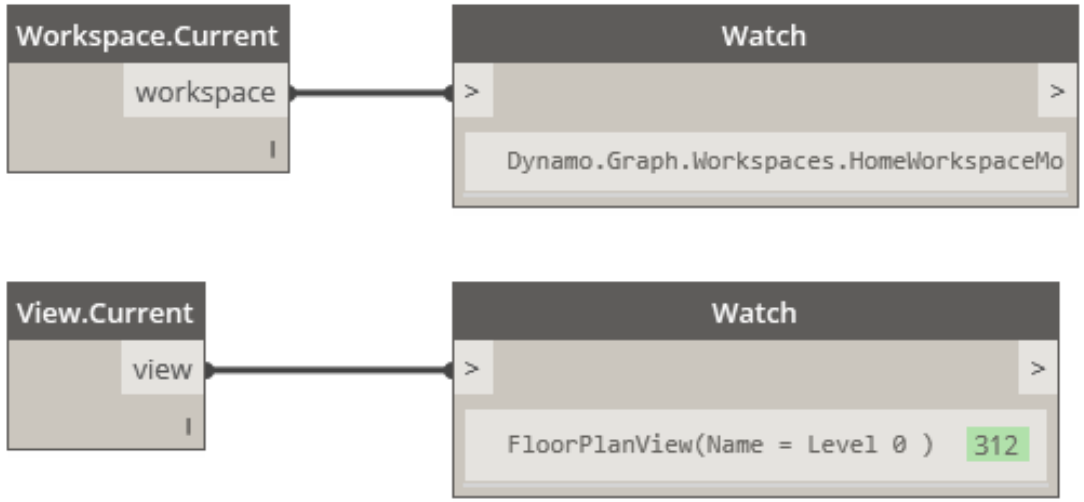


DYNAMO API
ACCESSING A GRAPH – LOCAL METHOD

```
//Accessing the current graph in Dynamo
public static Dynamo.Graph.Workspaces.WorkspaceModel Current()
   {
      var model = Dynamo.Applications.DynamoRevit.RevitDynamoModel;
      Dynamo.Graph.Workspaces.WorkspaceModel ws = model.CurrentWorkspace;
      return ws;
   }
```

```
//Accessing the current view in Revit
public static Revit.Elements.Element Current()
   {
      Document doc = RevitServices.Persistence.DocumentManager.Instance.CurrentDBDocument;
      return doc.ActiveView.ToDSType(true);
   }
```

DYNAMO API
ACCESSING A GRAPH – LOCAL METHOD



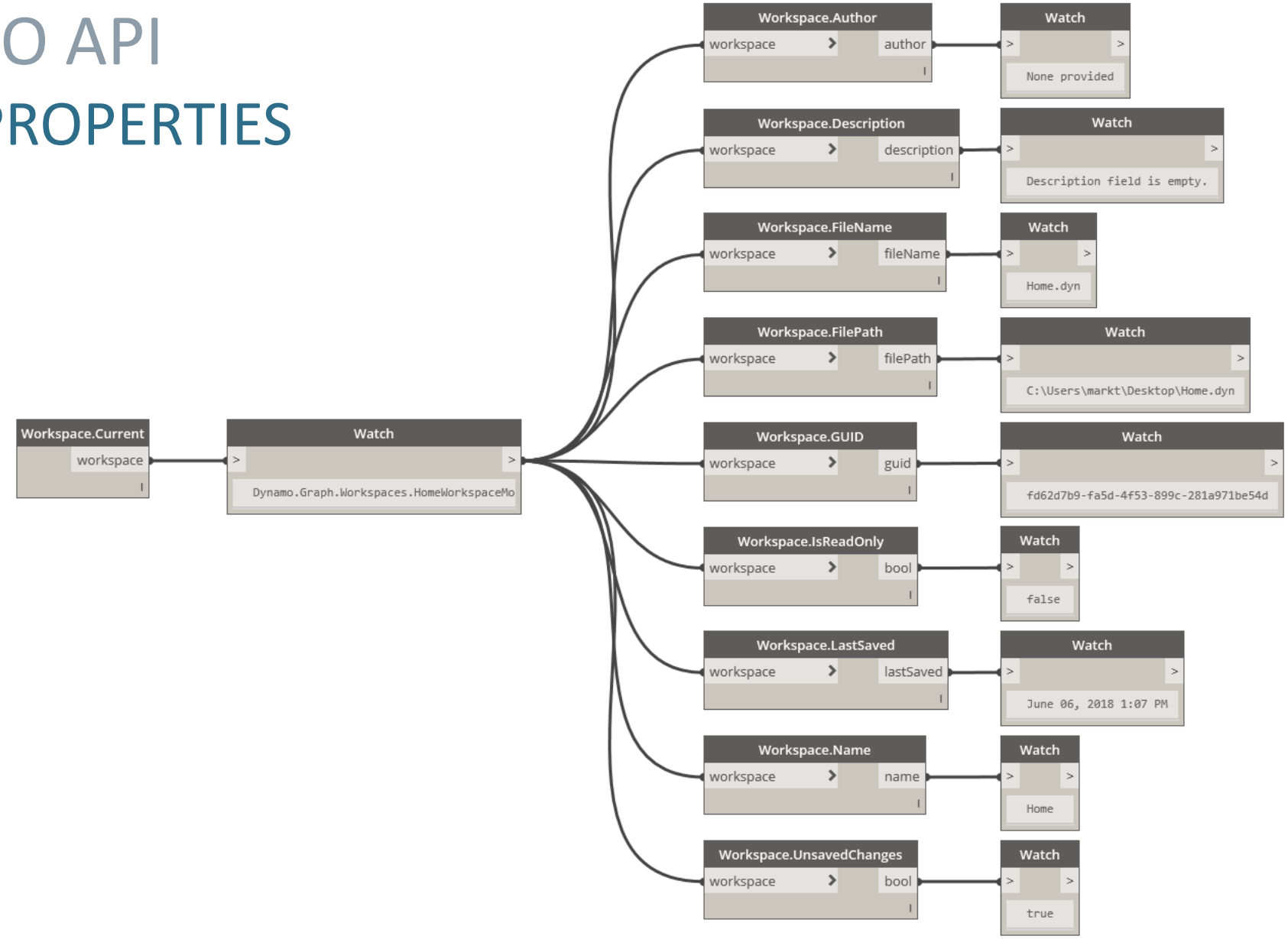
DYNAMO API
GRAPH PROPERTIES

```
//Reading when the current workspace was last saved
public static DateTime LastSaved(Dynamo.Graph.Workspaces.WorkspaceModel workspace)
    {
        return workspace.LastSaved;
    }


//Return a true or false statement based on whether the graph has unsaved changes
public static bool UnsavedChanges(Dynamo.Graph.Workspaces.WorkspaceModel workspace)
    {
        return workspace.HasUnsavedChanges;
    }
```

DYNAMO API
GRAPH PROPERTIES

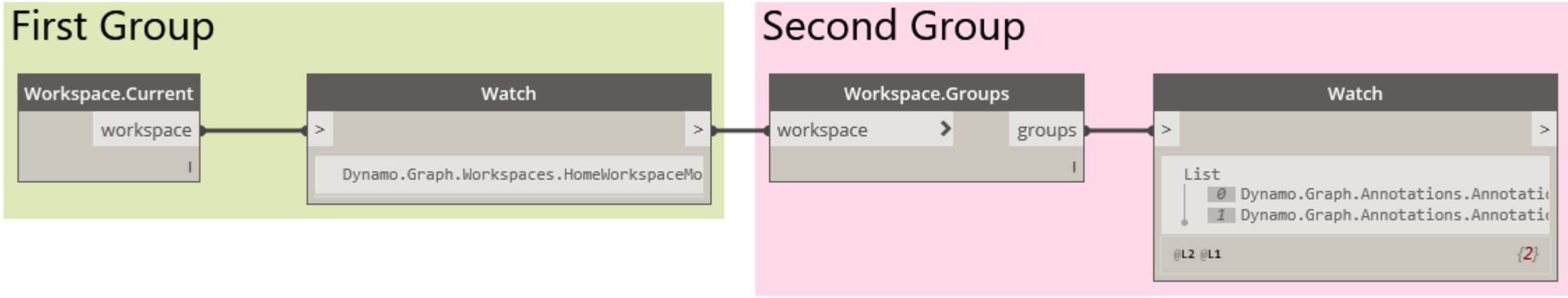
```
//Reading the current workspaces description
public static string Description(Dynamo.Graph.Workspaces.WorkspaceModel workspace)
    {
        var message = workspace.Description;
        if (message == null)
        {
            return "Description field is empty.";
        }
        else
        {
            return message;
        }
    }
```

DYNAMO API
GRAPH PROPERTIES



DYNAMO API
ANNOTATIONS (GROUPS)

```
//Return the Annotation Model Class
public static IEnumerable<Dynamo.Graph.Annotations.AnnotationModel>
Groups(Dynamo.Graph.Workspaces.WorkspaceModel workspace)
    {
        return workspace.Annotations;
    }
```



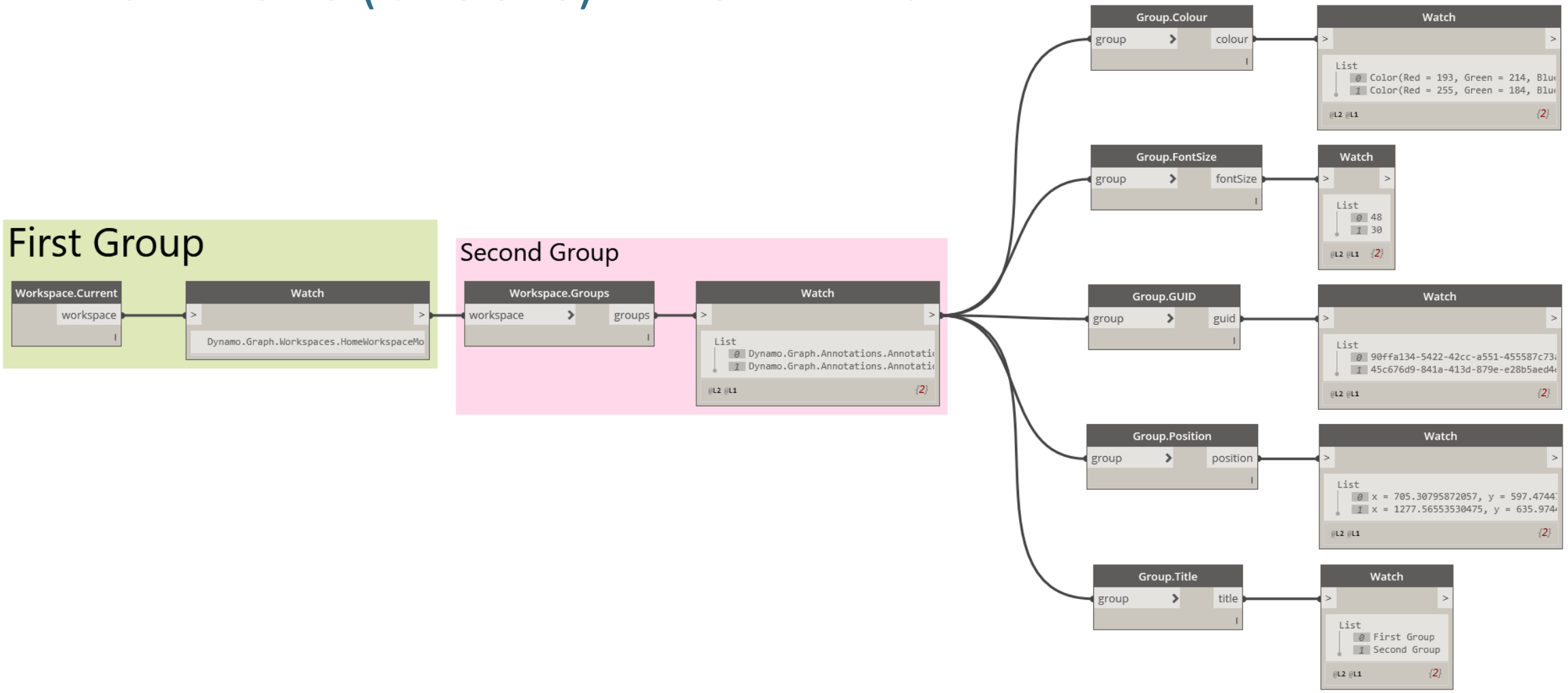
DYNAMO API

ANNOTATIONS (GROUPS) - PROPERTIES

```
//Get the location of the Group on the Workspace
public static string Position(Dynamo.Graph.Annotations.AnnotationModel group)
	{
	    double x = group.CenterX;
	    double y = group.CenterY;

	    string pos = "x = " + x.ToString() + ", y = " + y.ToString();
	    return pos;
	}

//Get the Font Size of the Group Title
public static double FontSize(Dynamo.Graph.Annotations.AnnotationModel group)
	{
	    return group.FontSize;
	}
```

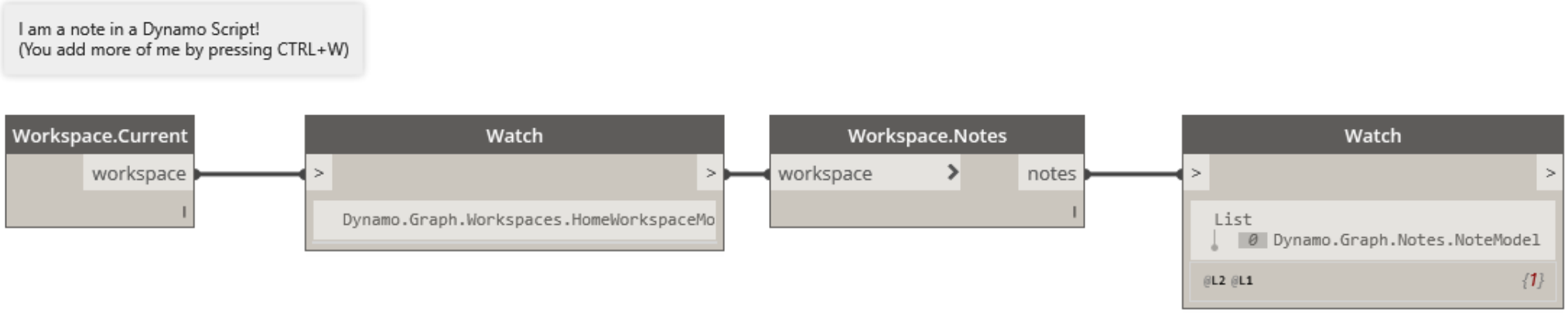
DYNAMO API

ANNOTATIONS (GROUPS) - PROPERTIES



DYNAMO API
NOTES

```
//Return the Note Model Class
public static IEnumerable<Dynamo.Graph.Notes.NoteModel> Notes(Dynamo.Graph.Workspaces.WorkspaceModel
workspace)
      {
         return workspace.Notes;
      }
```

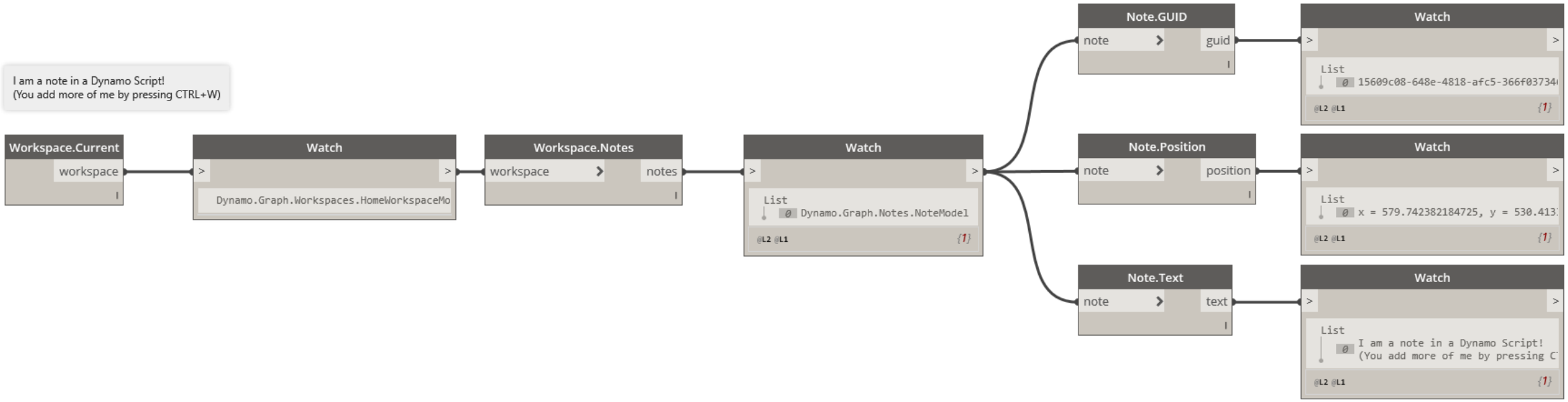


DYNAMO API
NOTES - PROPERTIES

```
//Get the GUID of the note
public static Guid GUID(Dynamo.Graph.Notes.NoteModel note)
      {
         return note.GUID;
      }

//Get the text of the note
public static string Text(Dynamo.Graph.Notes.NoteModel note)
      {
         return note.Text;
      }
```

DYNAMO API

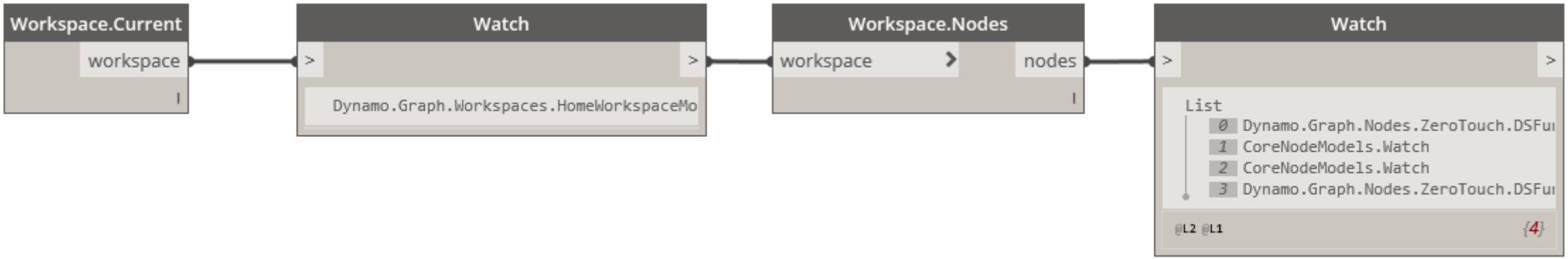
NOTES - PROPERTIES



DYNAMO API

NODES

```
//Return the Node Model Class
public static IEnumerable<Dynamo.Graph.Nodes.NodeModel> Nodes(Dynamo.Graph.Workspaces.WorkspaceModel
workspace)
      {
         return workspace.Nodes;
      }
```

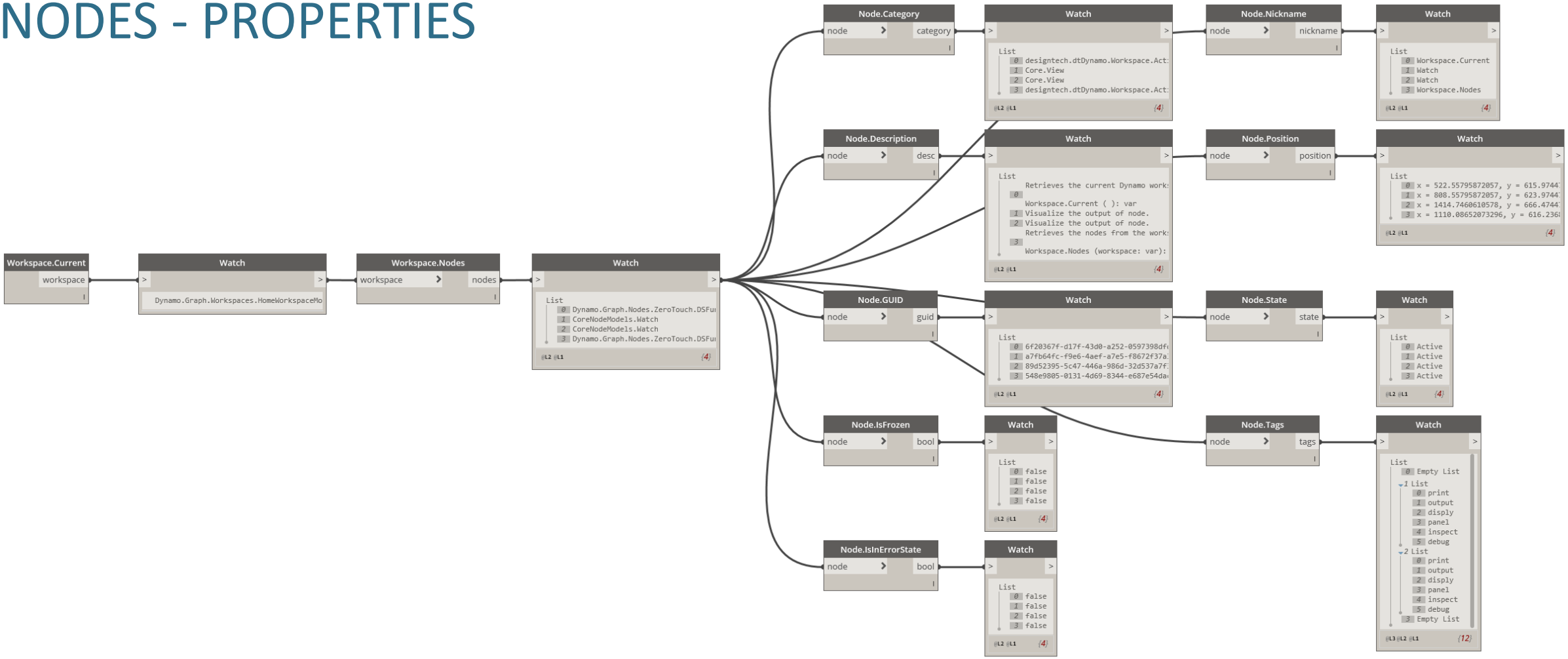


DYNAMO API
NODES - PROPERTIES

```
//Get the category of the node
public static string Category(Dynamo.Graph.Nodes.NodeModel node)
      {
         return node.Category;
      }


//Get the name of the node
public static string Nickname(Dynamo.Graph.Nodes.NodeModel node)
      {
         return node.NickName;
      }
```

DYNAMO API

NODES - PROPERTIES

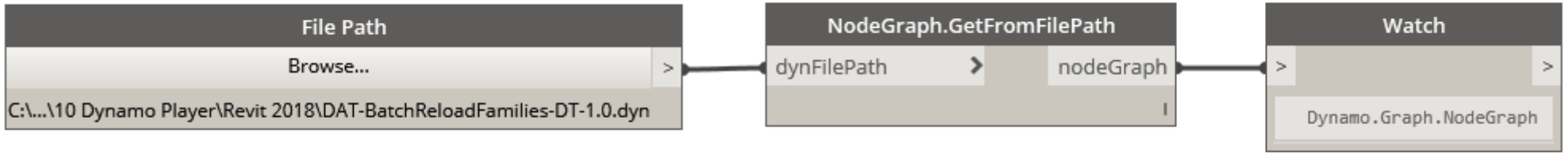


DYNAMO API

ACCESSING A GRAPH – EXTERNAL METHOD

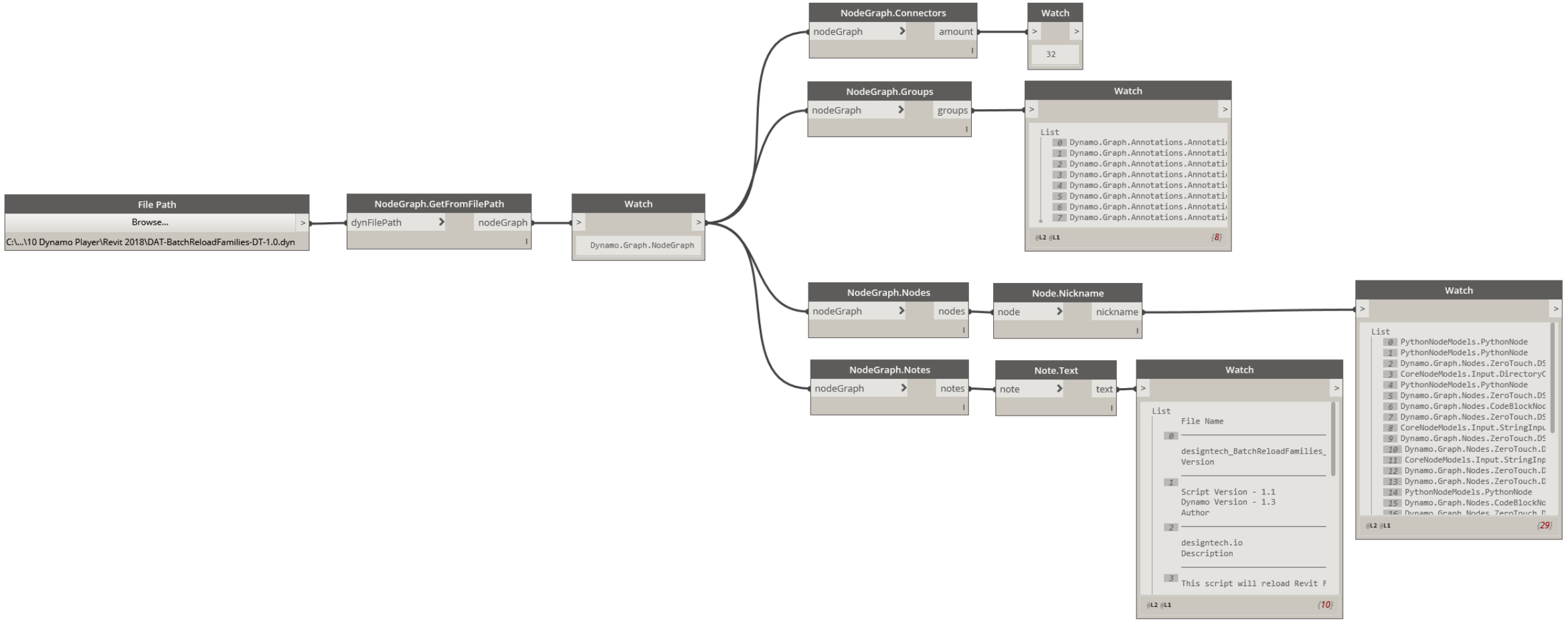
```
//Accessing an external graph in Dynamo
public static Dynamo.Graph.NodeGraph GetFromFilePath(string dynFilePath)
   {
      System.Xml.XmlDocument doc = new System.Xml.XmlDocument();
      doc.Load(dynFilePath);

      Dynamo.Graph.NodeGraph nG;
      Dynamo.Graph.Nodes.NodeLoaders.NodeFactory nF = new Dynamo.Graph.Nodes.NodeLoaders.NodeFactory();
      nG = Dynamo.Graph.NodeGraph.LoadGraphFromXml(doc, nF);

      return nG;
   }
```

DYNAMO API
ACCESSING A GRAPH – EXTERNAL METHOD



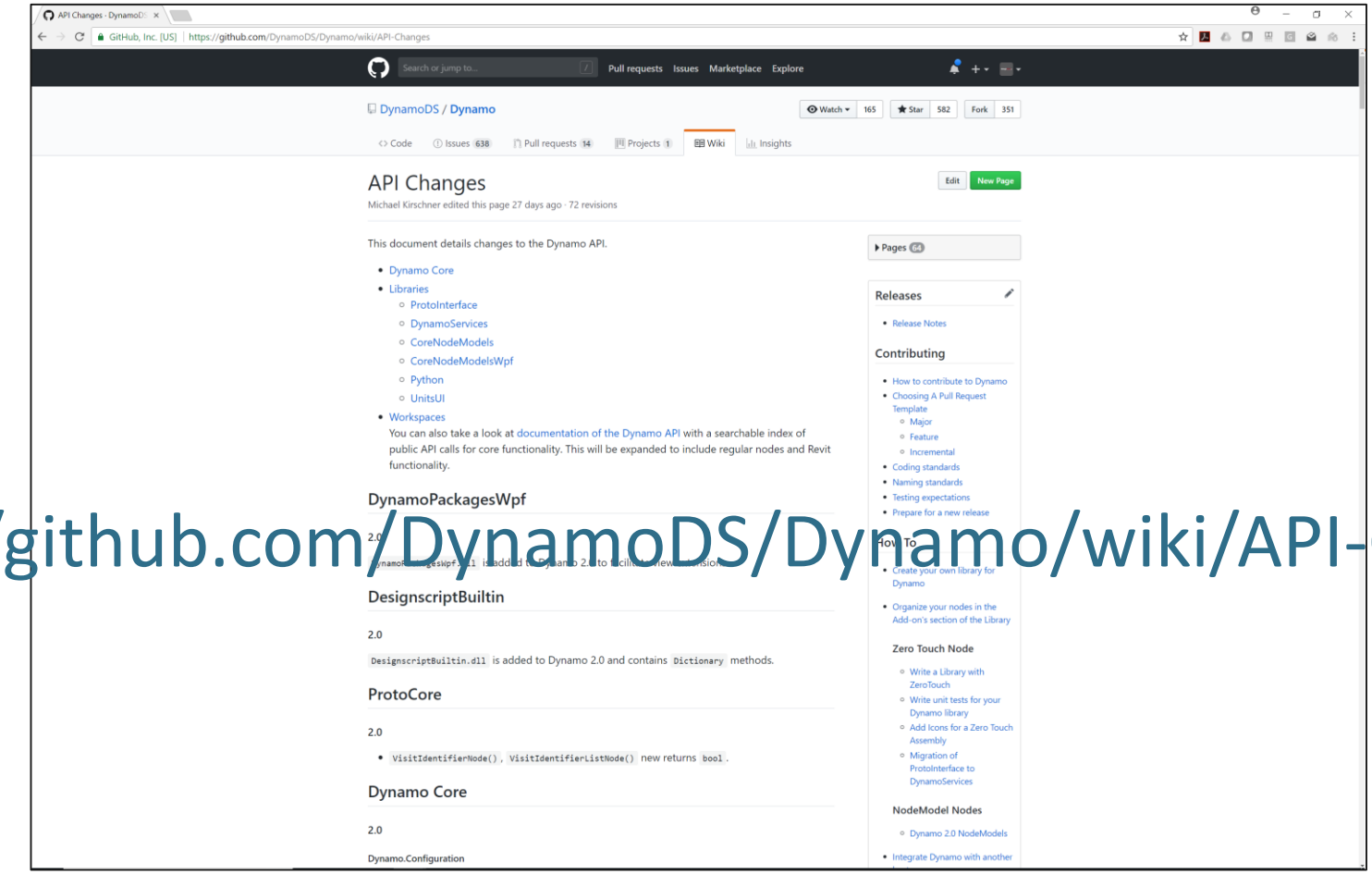
DYNAMO API

ACCESSING A GRAPH – EXTERNAL METHOD



DYNAMO API

CHANGES IN NEWER VERSIONS

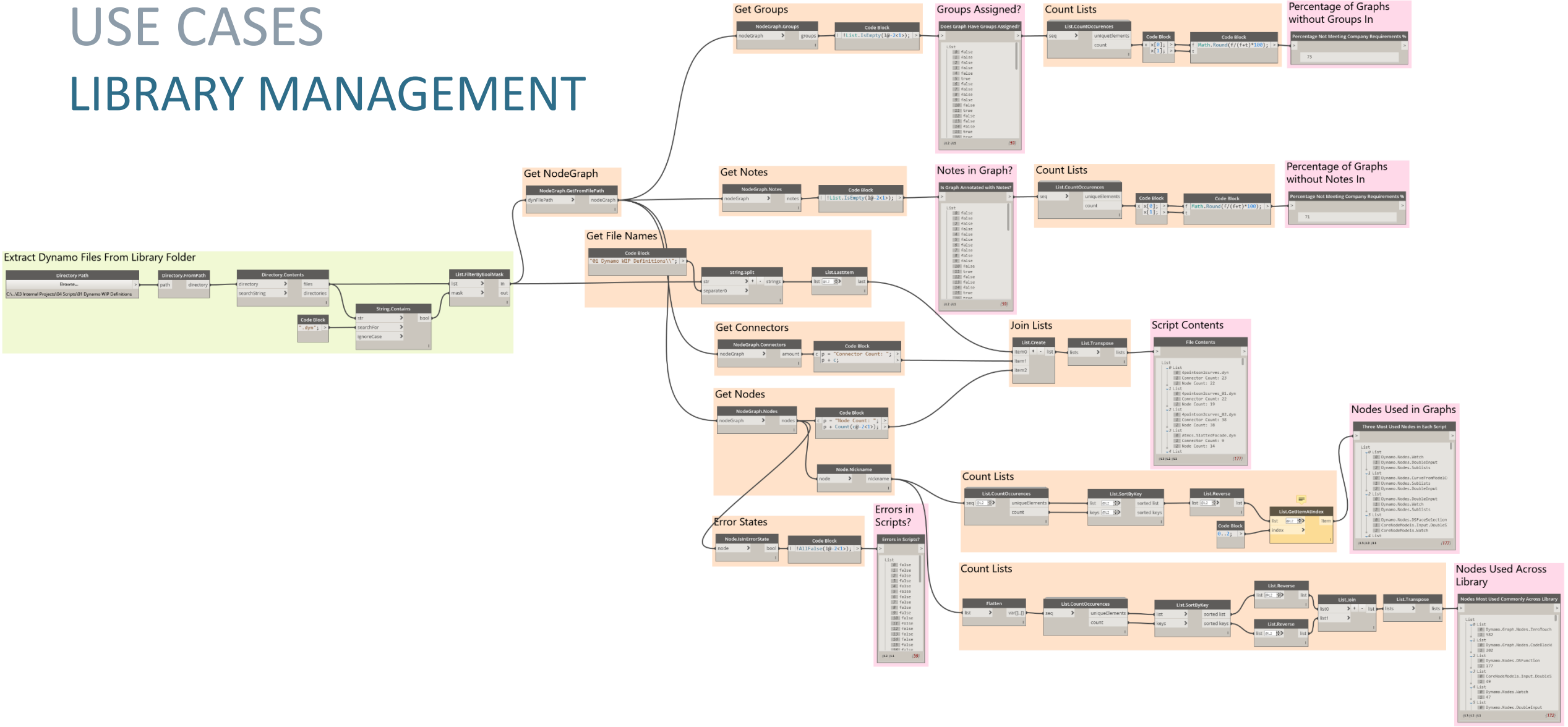


https://github.com/DynamoDS/Dynamo/wiki/API-Changes

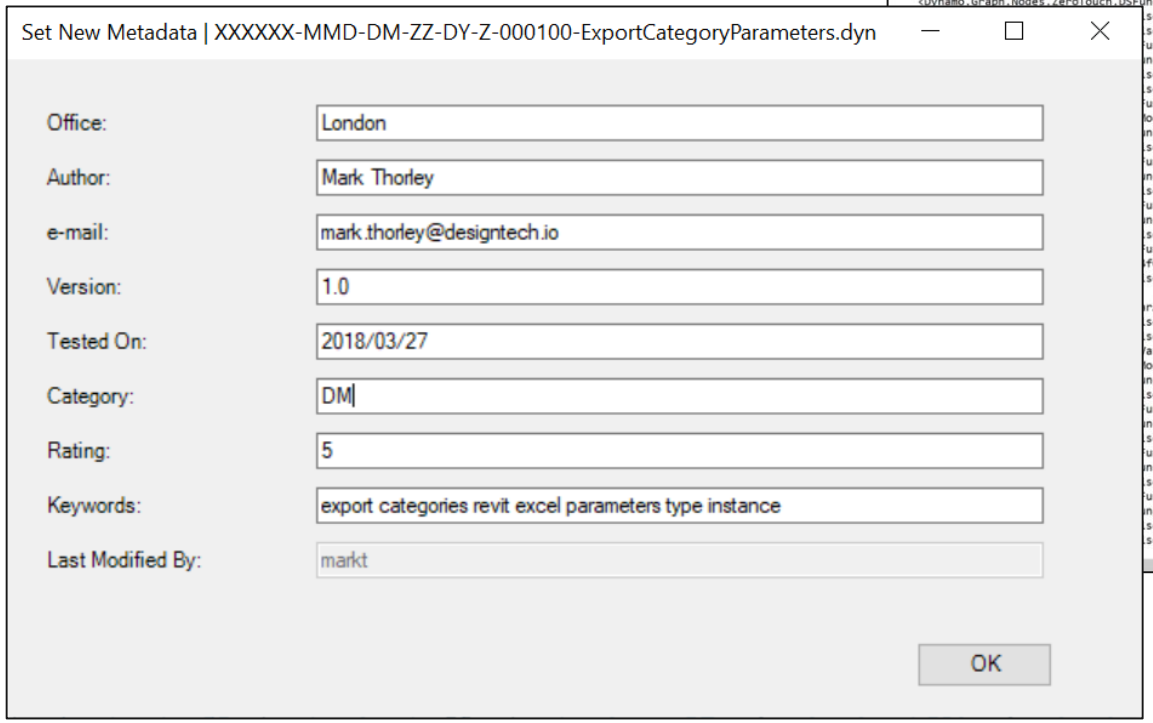
USE CASES

PRACTICAL USE

USE CASES
LIBRARY MANAGEMENT



USE CASES
ADDED METADATA



<Workspace Version="1.3.2.2480" X="961.43704140025" Y="79.1619050706823" zoom="0.761180597405121" ScaleFactor="1" Name="Home"
Description="{'1.office': 'London','2.author': 'Mark Thorley','3.email': 'mark.thorley@designtech.io','4.version': '1.0','5.tested on': '2018/03/27','6.category': 'DM','7.rating': '5','8.keywords': 'export categories revit excel parameters type instance','9.user': 'markt'}"
RunType="Manual" RunPeriod="1000" HasRunWithoutCrash="True">

USE CASES

USAGE TRACKER

Time			User	Machine					Application				
Start	End	Duration	Username	Name	Operating System	Processor	RAM	Version	Name	Version	Build Number	Language	Document Path
12/04/2018 14:31	12/04/2018 14:31	00:00.1	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 15:07	06/06/2018 15:07	00:00.2	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 15:07	06/06/2018 15:07	00:00.2	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 16:05	06/06/2018 16:05	00:19.7	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 16:07	06/06/2018 16:07	00:02.0	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 16:08	06/06/2018 16:08	00:00.1	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 16:08	06/06/2018 16:08	00:02.2	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 16:09	06/06/2018 16:09	00:00.3	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None
06/06/2018 16:09	06/06/2018 16:09	00:05.0	markt	DESKTOP-5PMK1JF	Microsoft Windows NT 6.2.9200.0	4	16	4.0.30319.42000	Autodesk Revit 2018	2018	20170223_1515(x64)	English_GB	None

Graph				
Name	Path	Last Saved	Node Count	Wire Count	Errors
XXXXXX-MMD-TM-ZZ-DY-Z-00000-DynamoTemplate.dyn	C:\Users\markt\Google Drive\designtech Server\02 Client Projects\Autodesk\Mott SCO14\Template\XXXXXX-MMD-TM-ZZ-DY-Z-00000-DynamoTemplate.dyn	12/04/2018 14:27	5	1	0
XXXXXX-MMD-DM-ZZ-DY-Z-000100-ExportCategoryParameters.dyn	C:\Users\markt\Google Drive\designtech Server\02 Client Projects\Autodesk\Mott SCO14\Script Library\XXXXXX-MMD-DM-ZZ-DY-Z-000100-ExportCategoryParameters.dyn	06/06/2018 15:06	62	72	0
XXXXXX-MMD-DM-ZZ-DY-Z-000100-ExportCategoryParameters.dyn	C:\Users\markt\Google Drive\designtech Server\02 Client Projects\Autodesk\Mott SCO14\Script Library\XXXXXX-MMD-DM-ZZ-DY-Z-000100-ExportCategoryParameters.dyn	06/06/2018 15:06	62	72	0
XXXXXX-MMD-PR-ZZ-DY-S-000201-CreateStructuralGaPlanViews.dyn	C:\Users\markt\Google Drive\designtech Server\02 Client Projects\Autodesk\Mott SCO14\Script Library\XXXXXX-MMD-PR-ZZ-DY-S-000201-CreateStructuralGaPlanViews.dyn	06/06/2018 16:04	265	362	0
XXXXXX-MMD-PR-ZZ-DY-S-000202-PlaceViewsOnSheets.dyn	C:\Users\markt\Google Drive\designtech Server\02 Client Projects\Autodesk\Mott SCO14\Script Library\XXXXXX-MMD-PR-ZZ-DY-S-000202-PlaceViewsOnSheets.dyn	06/06/2018 16:06	432	564	0
ANA-HighlightElementsWithWarnings-DT-1.0.dyn	C:\Users\markt\Google Drive\designtech Server\03 Internal Projects\04 Scripts\10 Dynamo Player\Revit 2018\ANA-HighlightElementsWithWarnings-DT-1.0.dyn	06/06/2018 16:08	26	28	0
ANA-OverrideMaskingRegions-DT-1.0.dyn	C:\Users\markt\Google Drive\designtech Server\03 Internal Projects\04 Scripts\10 Dynamo Player\Revit 2018\ANA-OverrideMaskingRegions-DT-1.0.dyn	06/06/2018 16:08	20	21	0
DAT-BatchSaveFamilies-DT-1.0.dyn	C:\Users\markt\Google Drive\designtech Server\03 Internal Projects\04 Scripts\10 Dynamo Player\Revit 2018\DAT-BatchSaveFamilies-DT-1.0.dyn	06/06/2018 16:09	37	42	0
DAT-FamiliesToExcel-DT-1.0.dyn	C:\Users\markt\Google Drive\designtech Server\03 Internal Projects\04 Scripts\10 Dynamo Player\Revit 2018\DAT-FamiliesToExcel-DT-1.0.dyn	06/06/2018 16:09	78	96	2

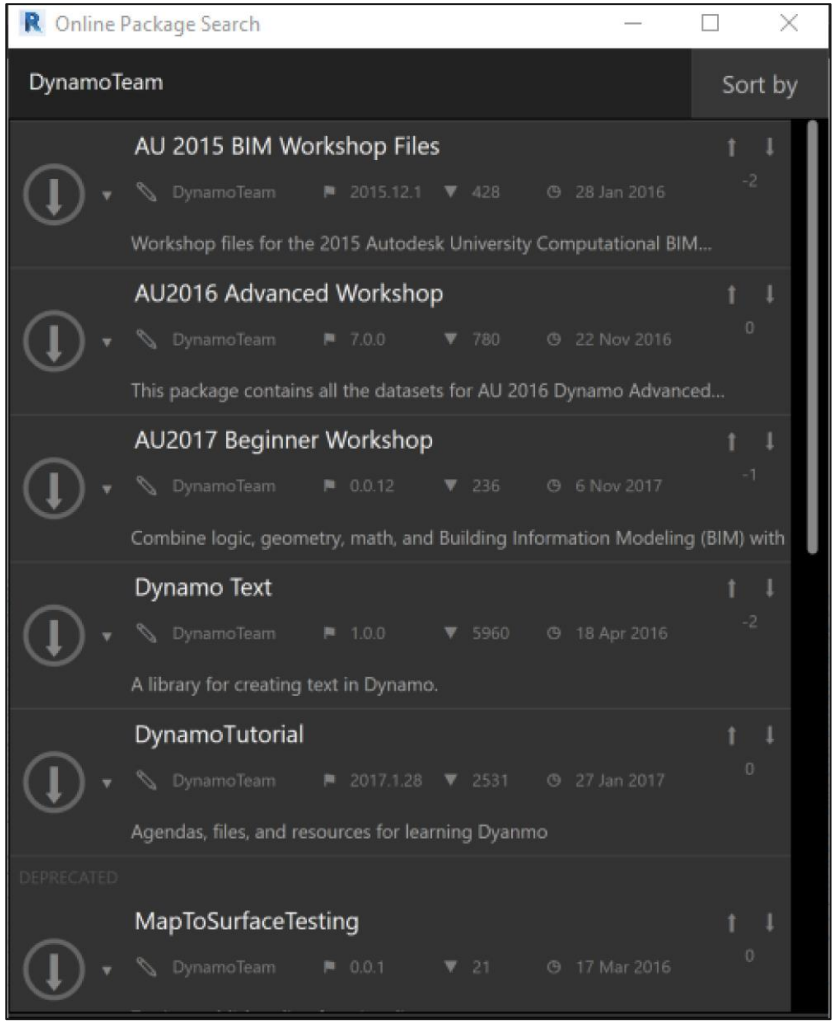
Company								
Office	Graph Author	Author Email	Version	Tested On	Category	Rating	Keywords	Last Modified By
London, UK	Mark Thorley	mark.thorley@designtech.io	1	27/03/2018	TM	5	template, usage, tracker	markt
London	Mark Thorley	mark.thorley@designtech.io	1	27/03/2018	DM	5	export categories revit excel parameters type instance	markt
London	Mark Thorley	mark.thorley@designtech.io	1	27/03/2018	DM	5	export categories revit excel parameters type instance	markt
London	Mark Thorley	mark.thorley@designtech.io	2	27/03/2018	PR	5	structure create views plan general arrangement revit	markt
LDN	Antonio Rodrigu	antonio.rodriguez@designtech.io	1.3	27/03/2018	PR	5	place views sheets revit	markt
London	Sol Amour	sol.amour@designtech.io	2	27/03/2018	ANA	4	revit highlight isolate warnings validation check error	markt
London	Antonio Rodrigu	antonio.rodriguez@designtech.io	3	27/03/2018	ANA	3	revit masking regions override validation	markt
London	Antonio Rodrigu	antonio.rodriguez@designtech.io	3	27/03/2018	DAT	5	export batch save family families revit folder	markt
London	Mark Thorley	mark.thorley@designtech.io	1	27/03/2018	DAT	2	families parameters export excel revit family instance types	markt

EXTENSIONS

NON-COWBOY METHOD

EXTENSIONS
WHAT ARE THEY?

- A powerful development tool in the Dynamo Ecosystem
- Create custom functionality based on Dynamo interactions and logic
- Use when you want to control the creation/removal of nodes in graphs
- Use when you want to add new UI functionality



EXTENSIONS
TWO TYPES

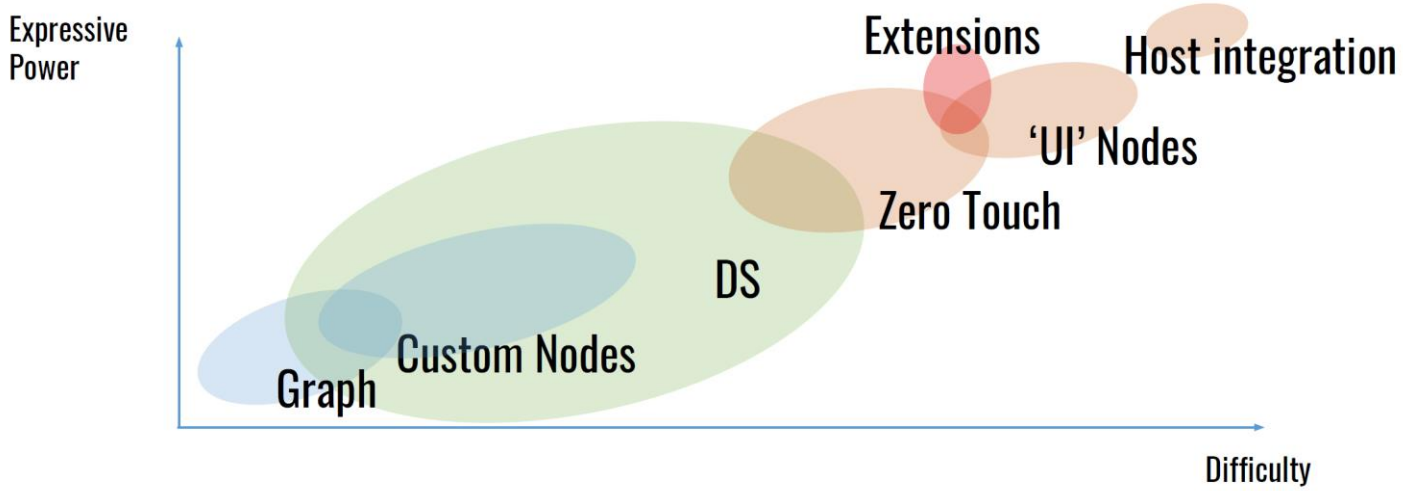
Regular Extensions

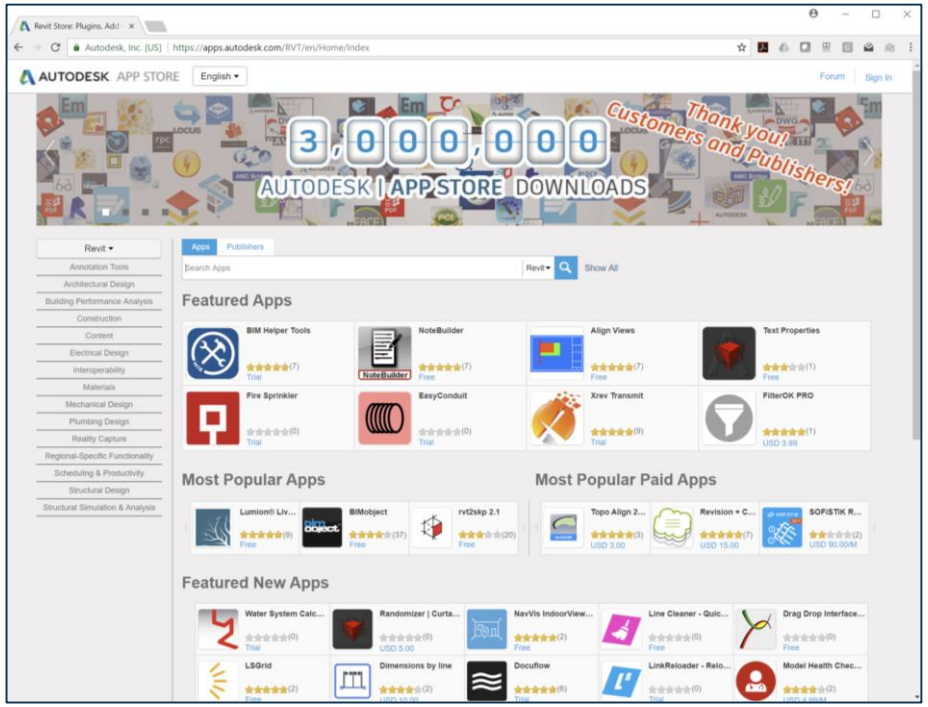
- Extend the functionality of Dynamo without the need for additional UI work
- An example would be logging information to the Dynamo console.

View Extensions

- The View Extension Framework allows you to extend the Dynamo UI by creating custom display items.
- The Package Manager and Notification window is an example of a View Extension

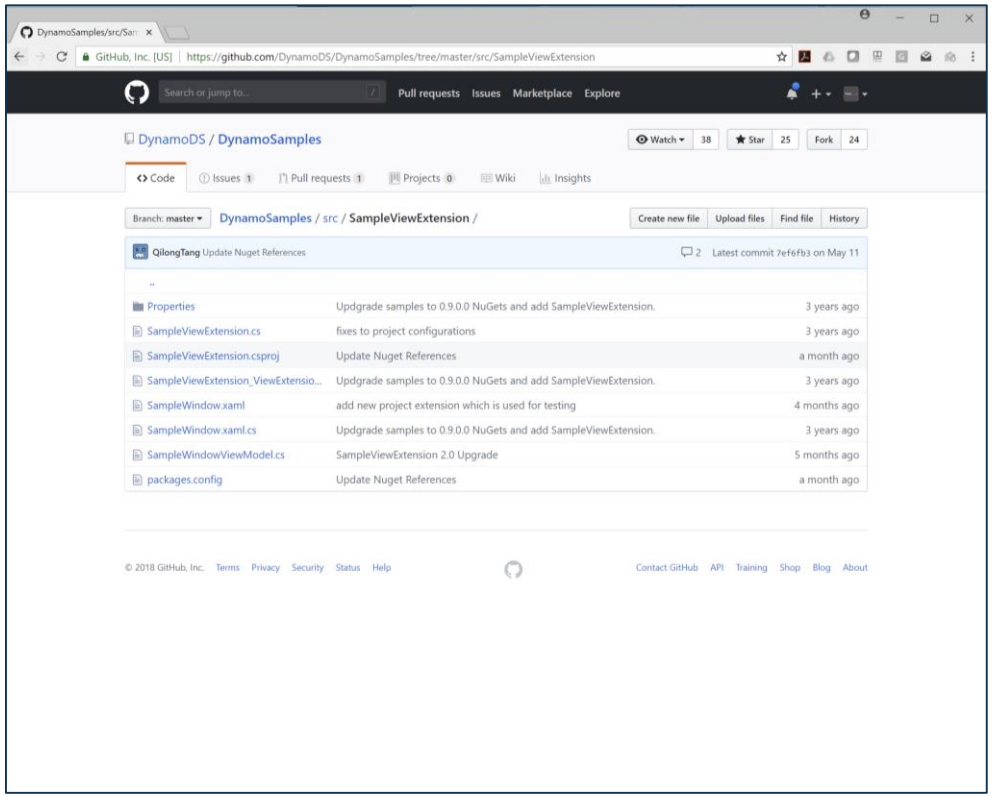
EXTENSIONS
PROBLEMS/POSSIBILITIES



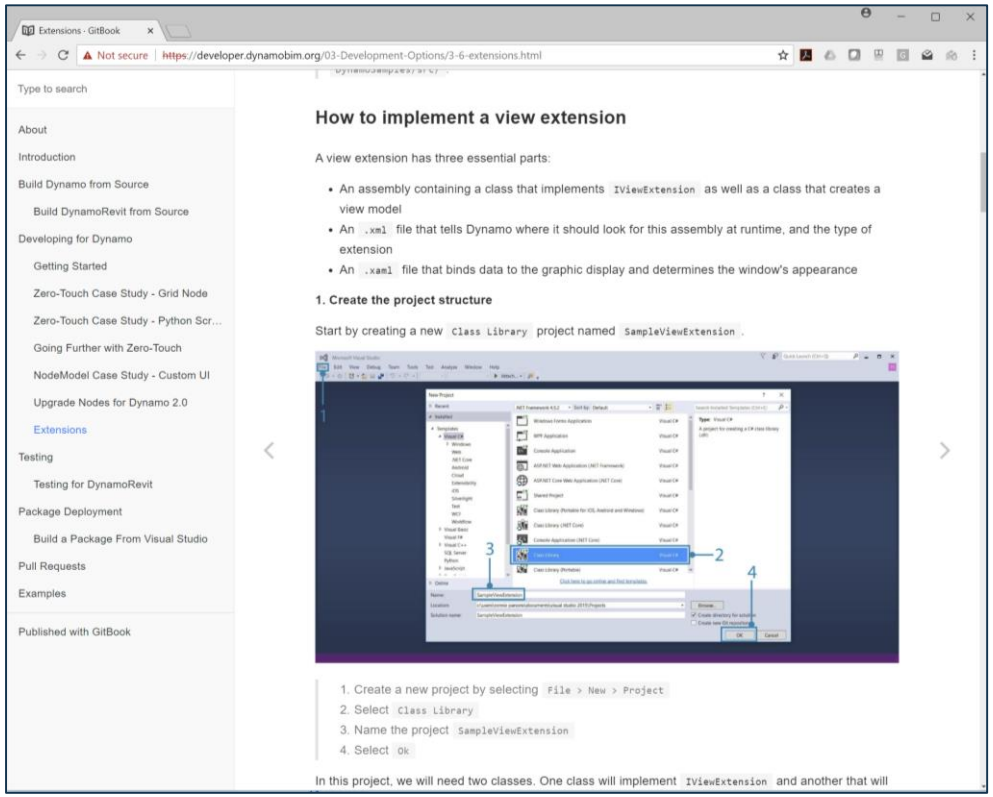


EXTENSIONS

GETTING STARTED



https://github.com/DynamoDS/DynamoSamples/tree/master/src/SampleViewExtension

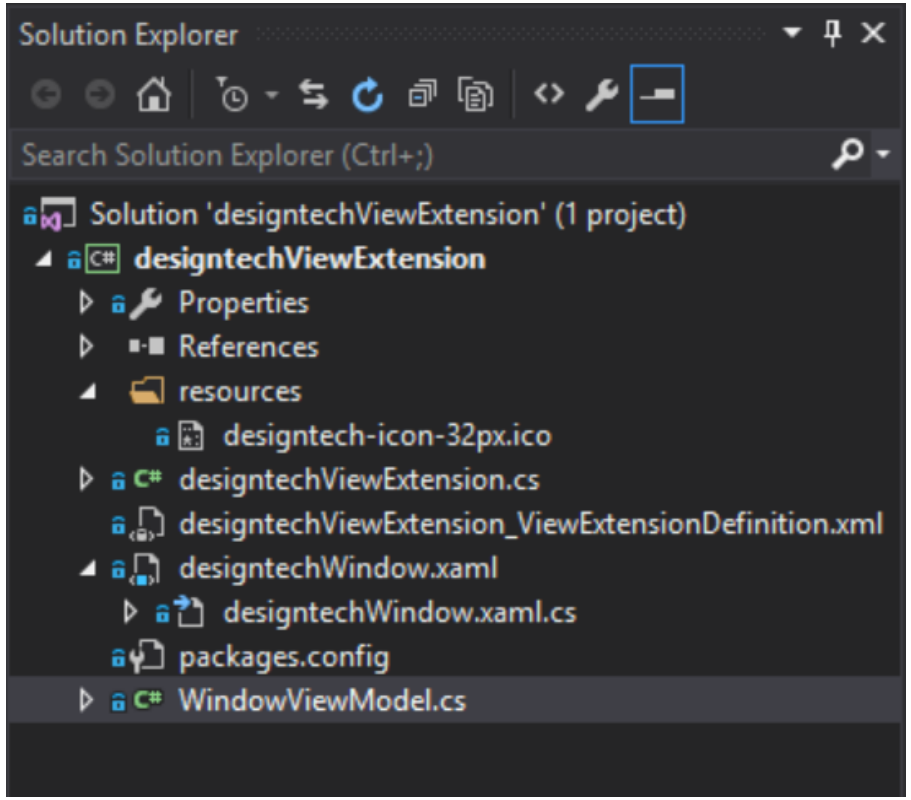


https://developer.dynamobim.org/03-Development-Options/3-6-extensions.html

EXTENSIONS
GETTING STARTED

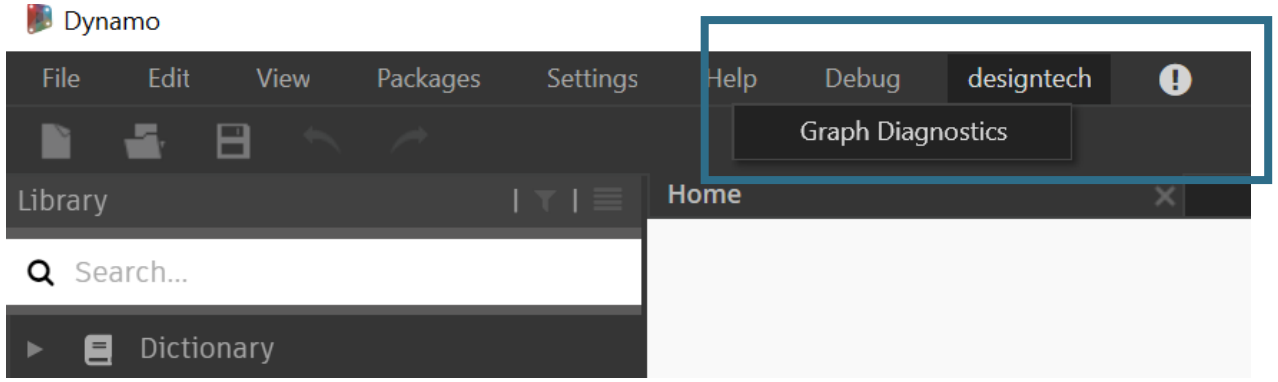
Three essential parts:

- An assembly containing a class that implements IViewExtension as well as a class that creates a view model
- An .xaml file that binds data to the graphic display and determines the window's appearance
- An .xml file that tells Dynamo where it should look for this assembly at runtime, and the type of extension



EXTENSIONS
IViewExtension

```
namespace designtechViewExtension
{
   public class designtechViewExtension : IViewExtension
   {
      private MenuItem sampleMenuItem;
      public void Dispose()
      {
      }
      public void Startup(ViewStartupParams p)
      {
      }
      public void Loaded(ViewLoadedParams p)
      {
         designtechMenuItem = new MenuItem { Header = "designtech" };
         var VM = p.DynamoWindow.DataContext as DynamoViewModel;
         designtechGraphDiagnosticsMenuItem = new MenuItem { Header = "Graph Diagnostics" };
         designtechGraphDiagnosticsMenuItem.Click += (sender, args) =>
         {
            var viewModel = new WindowViewModel(p);
```



EXTENSIONS

NotificationObject

```
//ReadyParams
public WindowViewModel(ReadyParams p)
{
    readyParams = p;
    p.CurrentWorkspaceChanged += CurrentWorkspaceModel_GraphSaved;
    p.CurrentWorkspaceModel.NodeAdded += CurrentWorkspaceModel_NodeCount;
    p.CurrentWorkspaceModel.NodeRemoved += CurrentWorkspaceModel_NodeCount;
    p.CurrentWorkspaceModel.ConnectorAdded += CurrentWorkspaceModel_WireCount;
    p.CurrentWorkspaceModel.ConnectorDeleted += CurrentWorkspaceModel_WireCount;
    p.CurrentWorkspaceModel.NodeAdded += CurrentWorkspaceModel_NodesChanged;
    p.CurrentWorkspaceModel.NodeRemoved += CurrentWorkspaceModel_NodesChanged;
}

//ChangeProperty
private void CurrentWorkspaceModel_GraphSaved(Dynamo.Graph.Workspaces.IWorkspaceModel
obj)
{
    RaisePropertyChanged("GraphLastSaved");
}
```

EXTENSIONS
AFTER BUILD EVENTS

.xml

copy "$(ProjectDir)bin\Release*.xml" "C:\Users\markt\Documents\GitHub\Dynamo\bin\AnyCPU\Debug\viewExtensions"

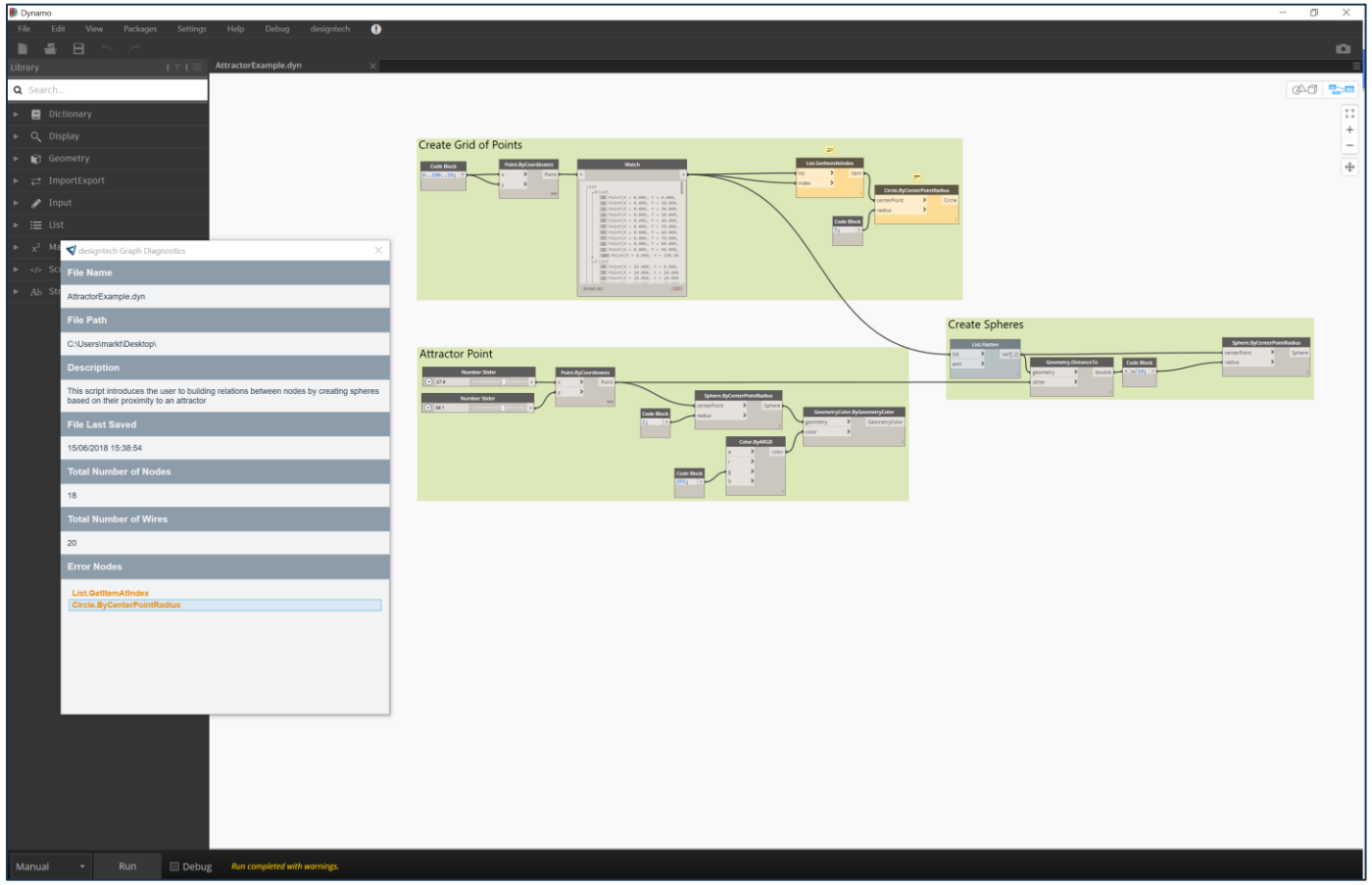
C:\Program Files\Dynamo\Dynamo Core\2.0\viewExtensions

.dll

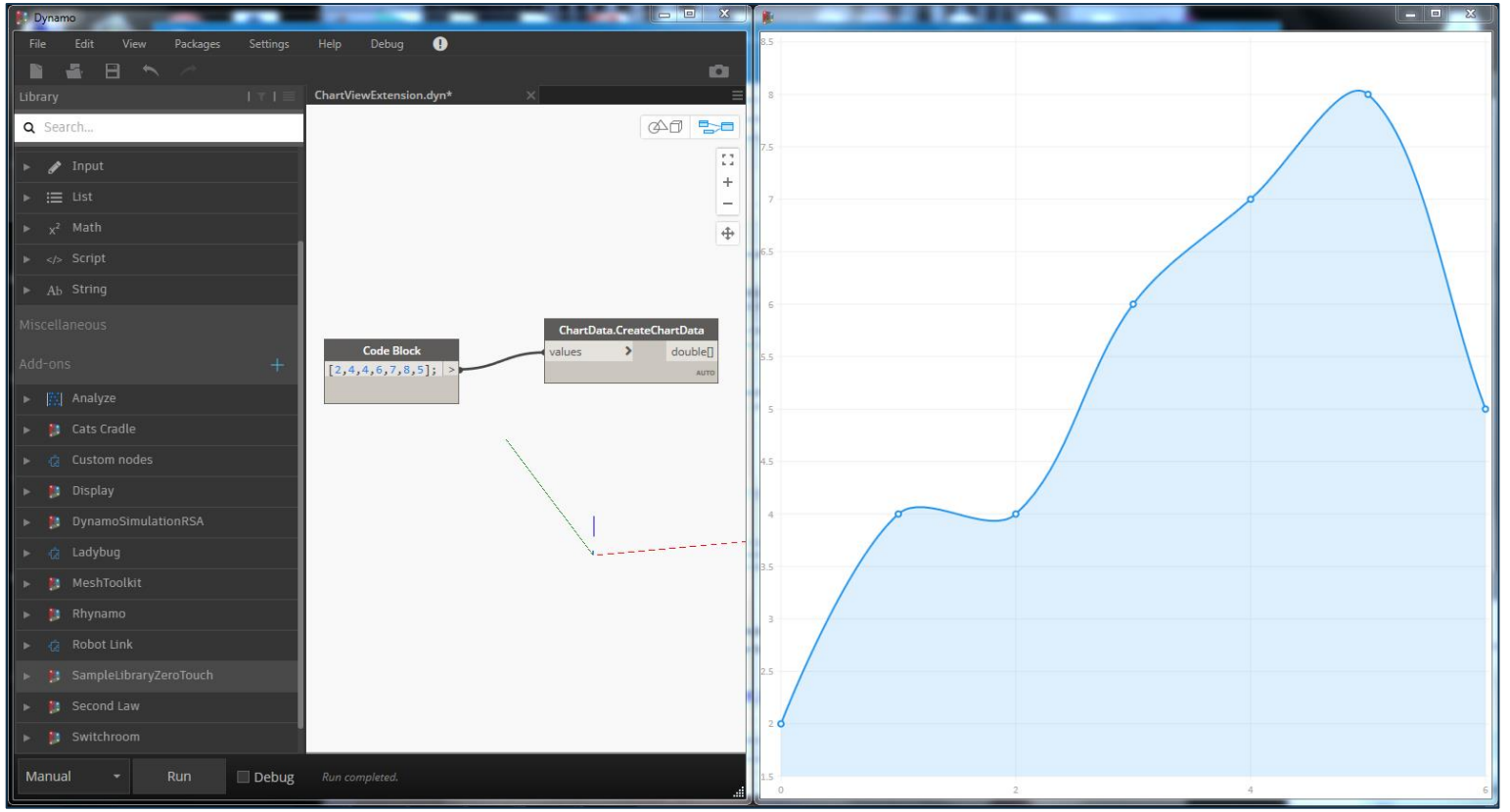
copy "$(ProjectDir)bin\Release*.dll" "C:\Users\markt\Documents\GitHub\Dynamo\bin\AnyCPU\Debug\"

C:\Program Files\Dynamo\Dynamo Core\2.0

EXTENSIONS
EXAMPLES

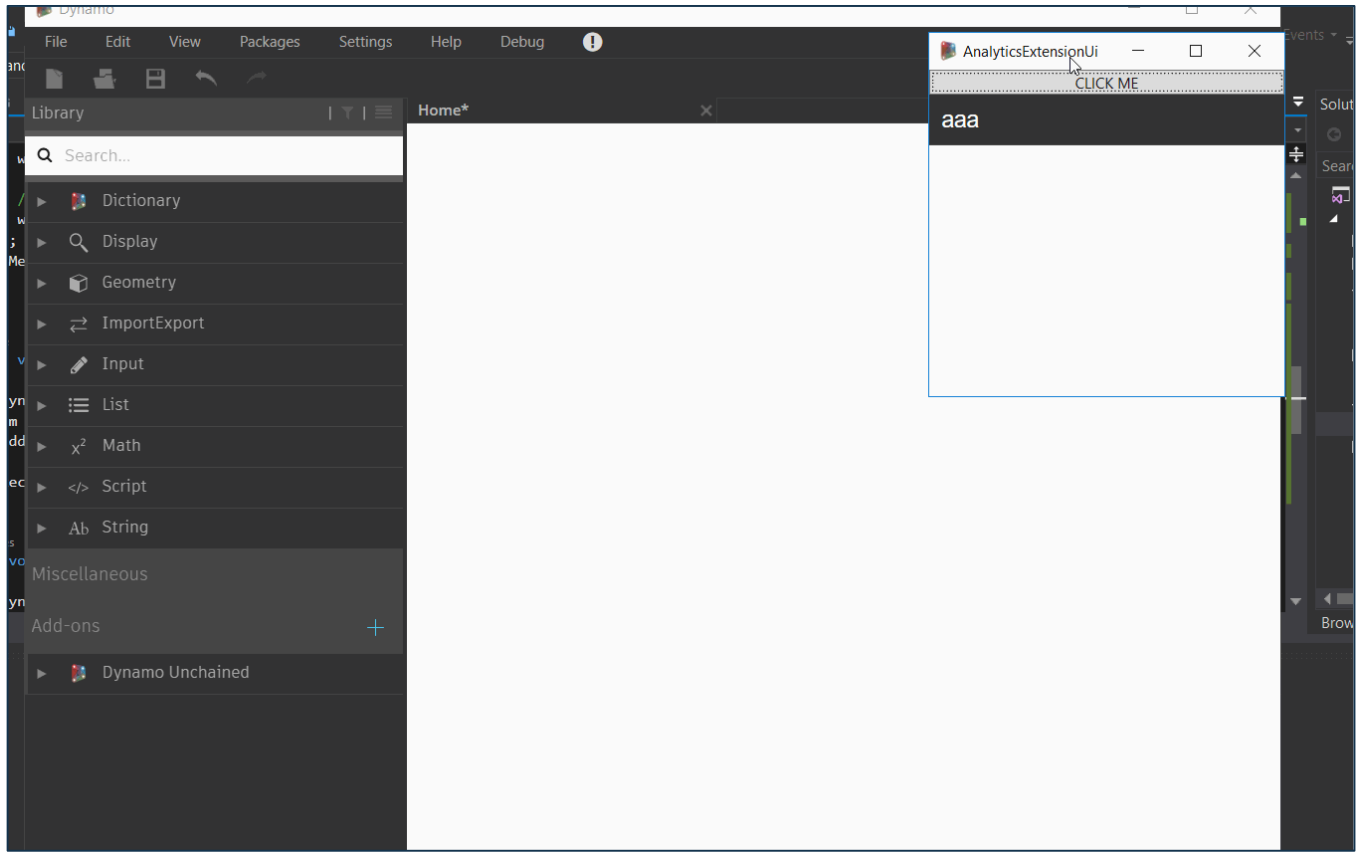


EXTENSIONS
EXAMPLES



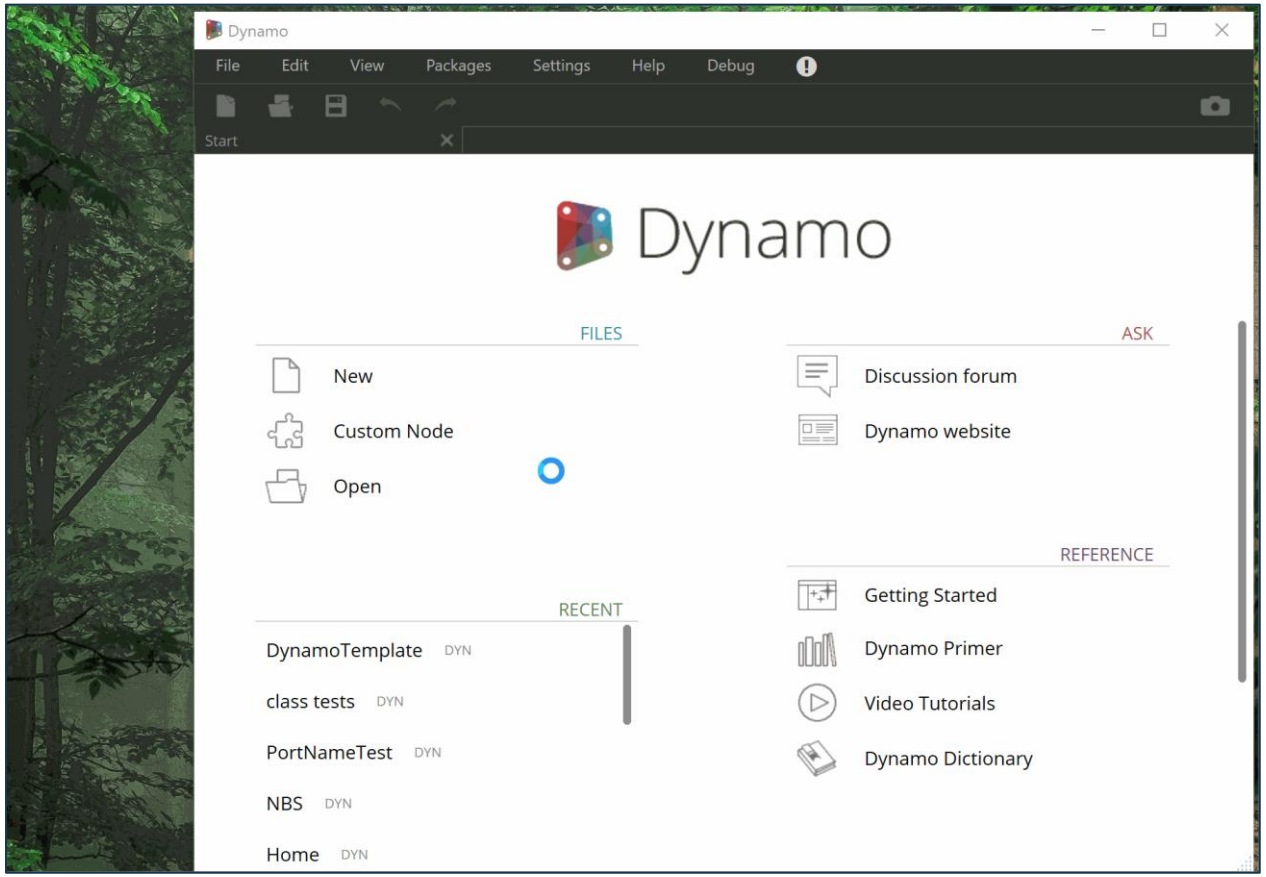
Credit: Alex Balham

EXTENSIONS
EXAMPLES



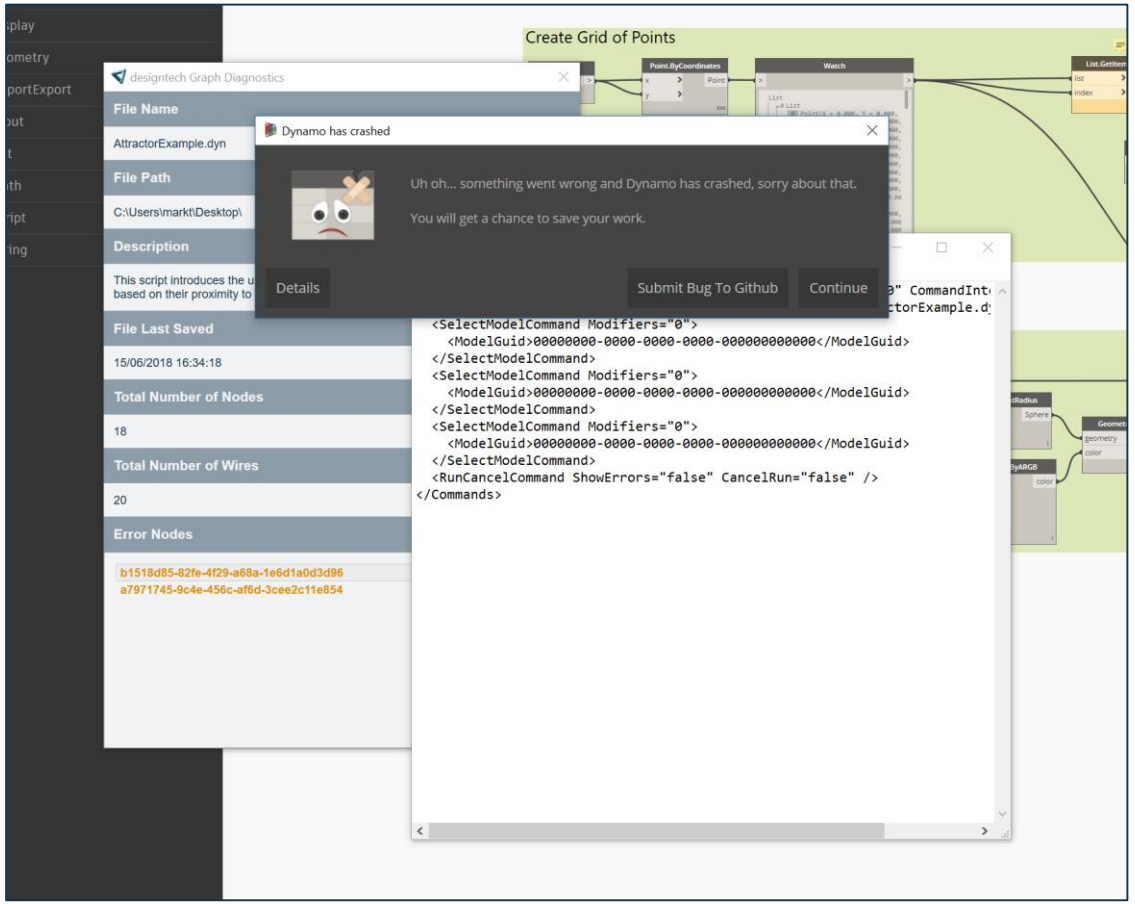
Credit: Matteo Cominetti

EXTENSIONS
EXAMPLES



Credit: Radu Gidei

EXTENSIONS
PATIENCE NEEDED



QUESTIONS

IS ANYONE STILL AWAKE…

CONTACT
FUTURE ITEMS

Source code freely available on GitHub

https://github.com/MarkThorley/designtechDynamo

https://github.com/MarkThorley/designtechViewExtension

Additional Learning

https://developer.dynamobim.org/03-Development-Options/3-6-extensions.html

https://www.youtube.com/watch?v=qLGsRcIOwzc&feature=youtu.be

https://github.com/DynamoDS/DynamoSamples/tree/master/src/SampleViewExtension

Feel free to reach out on:

@markthorley1

mark.thorley@designtech.io

www.designtech.io

AUTODESK®

Make anything.