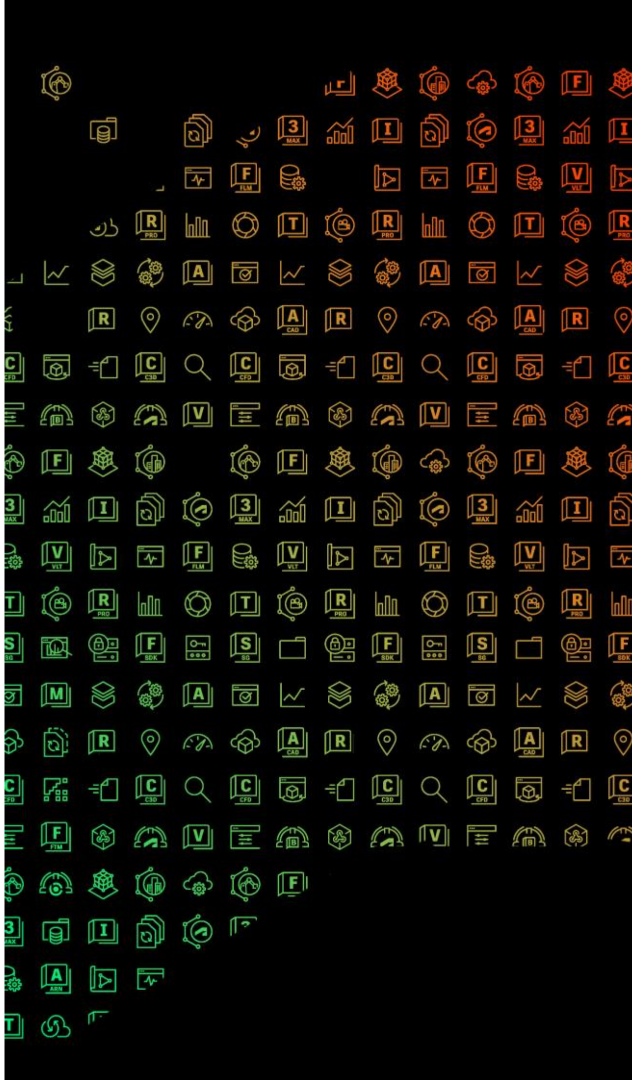
AUTODESK
DevCon

Autodesk Construction Cloud API – Deep Dive

Mikako Harada
Senor Manager, Developer Technical Services AEC

Safe Harbor Statement

Agenda



Shared Services in ACC

ACC/BIM 360 API Component Services & Structure

BIM Collaborate/Pro

- Model Coordination

Takeoff

- Takeoff*1

Build

- Assets
- Cost
- RFI
- Submittals*3
- Checklists*2
- Forms*1
- Photos*1
- Sheets*1
- AutoSpecs*1

Docs

- Custom attributes
- Permissions*3
- PDF export*3
- Naming standard

Shared Services

- Data Connector
- **Model Properties**
- **Relationships**
- Admin*3
- Issues*3
- Locations*3

Platform API

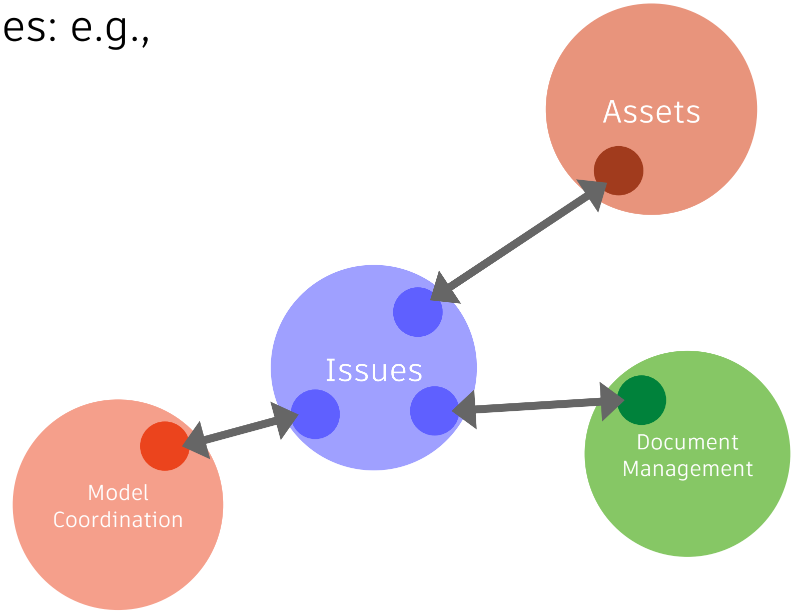
- Data Management
- Model Derivatives
- Webhooks
- Viewer
- OAuth

*1) ACC only, *2) BIM 360 only, *3) Not compatible or partially compatible

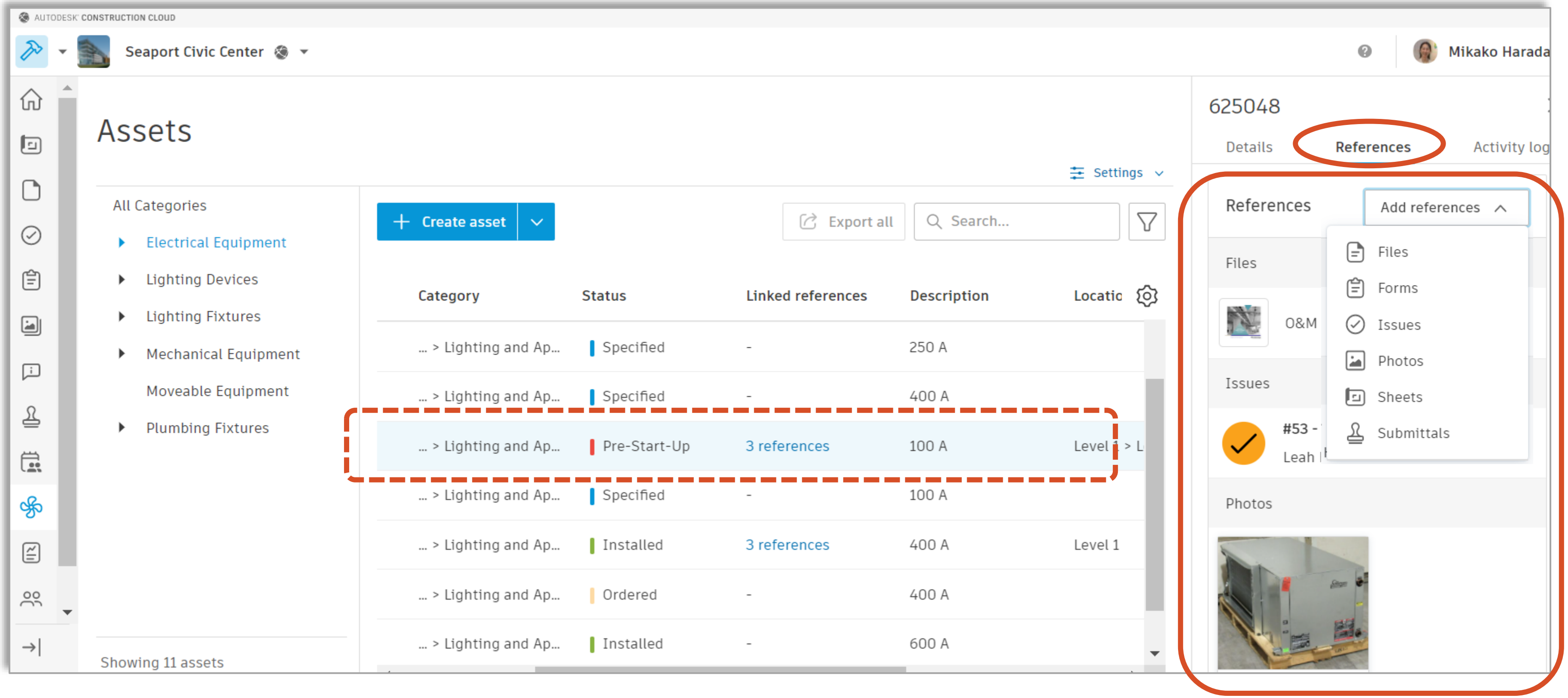
Relationships API

Relationships API

- Represent links between items (Entity) that reside in different functional components (Domain)
- Used in ACC and BIM 360 by various modules: e.g.,
  - Model Coordination, Assets, Forms, Photos, Cost
  - Older implementations may not be using it
- Read & Write
- Publicly available today



Relationships API



AUTODESK CONSTRUCTION CLOUD

Seaport Civic Center

Mikako Harada

Assets

Settings

All Categories
- Electrical Equipment
- Lighting Devices
- Lighting Fixtures
- Mechanical Equipment
- Moveable Equipment
- Plumbing Fixtures

Create asset | Export all | Search...

Category	Status	Linked references	Description	Locatic
... > Lighting and Ap...	Specified	-	250 A	
... > Lighting and Ap...	Specified	-	400 A	
... > Lighting and Ap...	Pre-Start-Up	3 references	100 A	Level 1 > L
... > Lighting and Ap...	Specified	-	100 A	
... > Lighting and Ap...	Installed	3 references	400 A	Level 1
... > Lighting and Ap...	Ordered	-	400 A	
... > Lighting and Ap...	Installed	-	600 A	

Showing 11 assets

625048

Details | References | Activity log

References

Add references
- Files
- Forms
- Issues
- Photos
- Sheets
- Submittals

Files

O&M

Issues

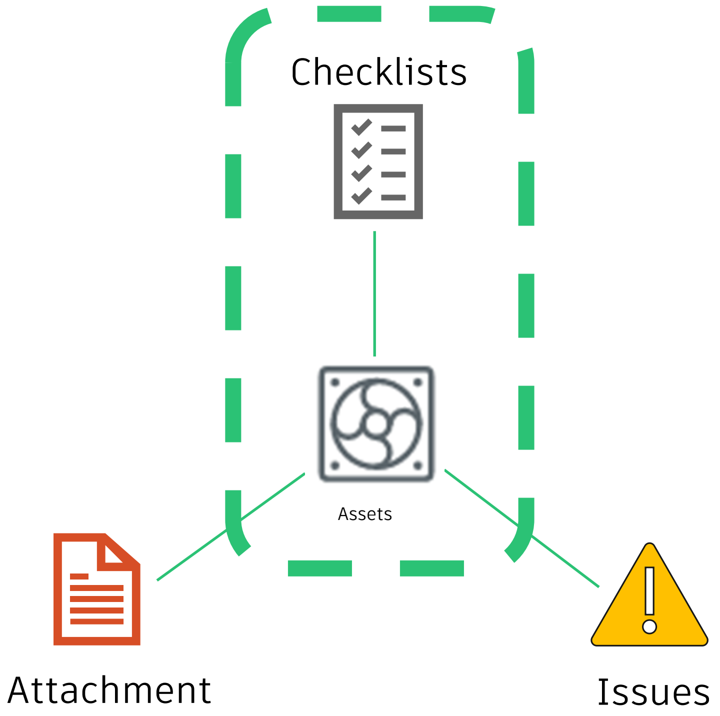
#53 -

Leah

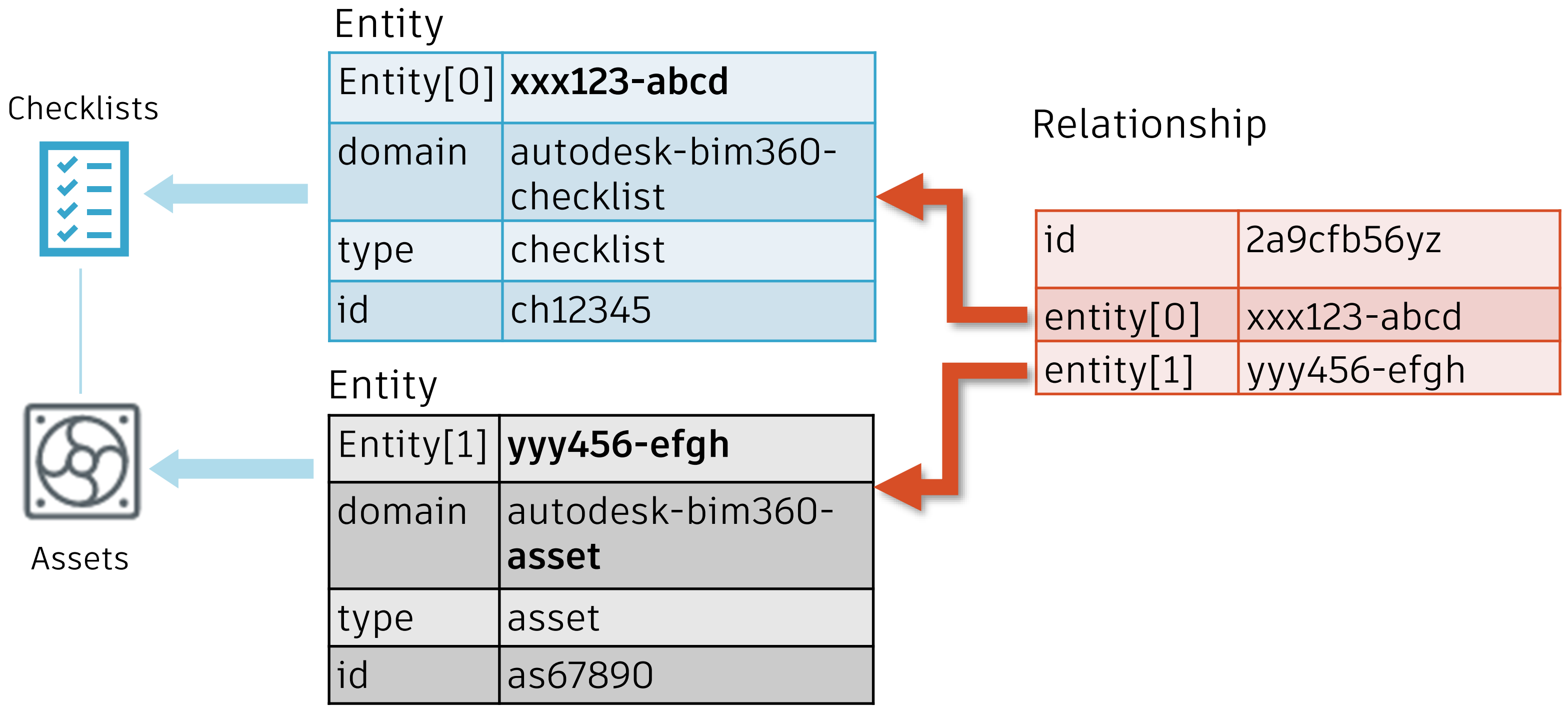
Photos

Relationships API

Example



Relationships API



Entity

Entity[0]	**xxx123-abcd**
domain	autodesk-bim360-checklist
type	checklist
id	ch12345

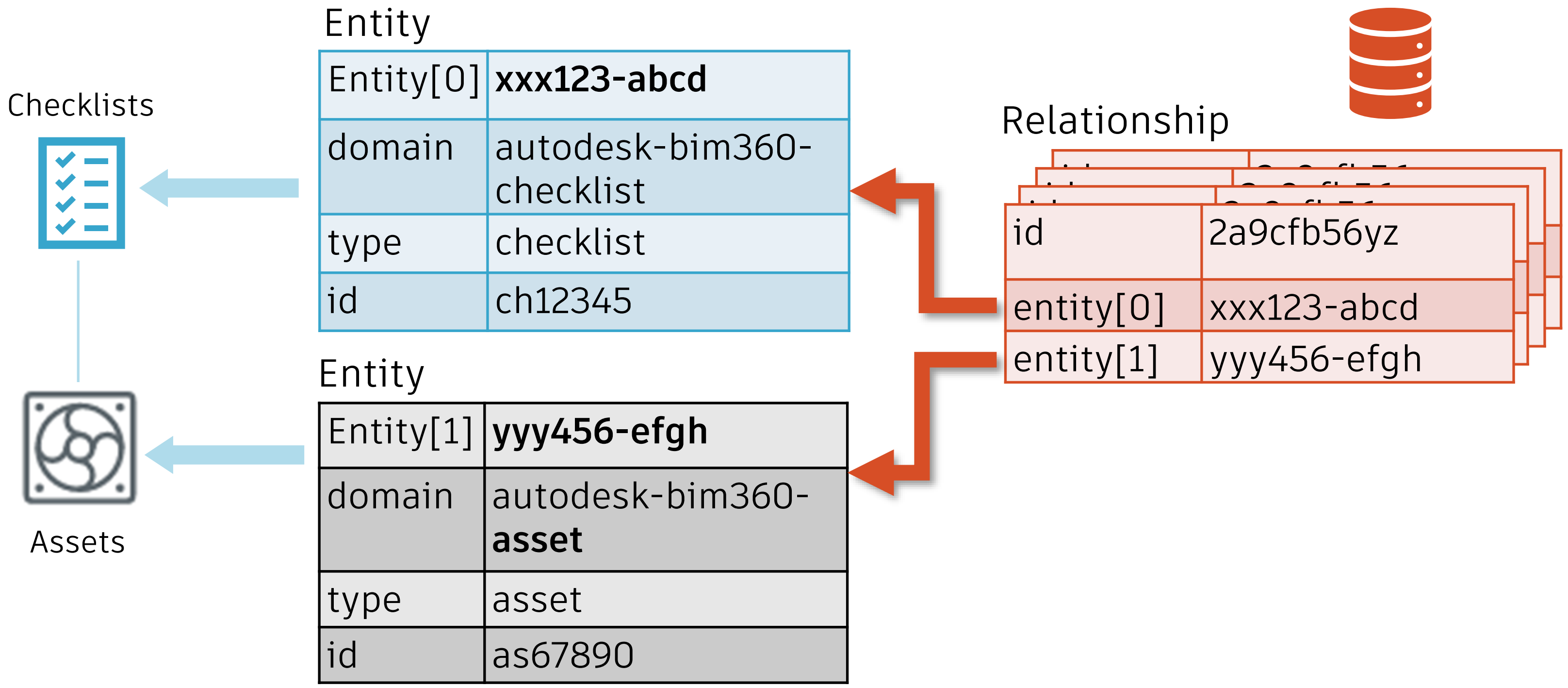
Entity

Entity[1]	**yyy456-efgh**
domain	autodesk-bim360-**asset**
type	asset
id	as67890

Relationship

id	2a9cfb56yz
entity[0]	xxx123-abcd
entity[1]	yyy456-efgh

Relationships API



Entity[0]	**xxx123-abcd**
domain	autodesk-bim360-checklist
type	checklist
id	ch12345

Entity[1]	**yyy456-efgh**
domain	autodesk-bim360-**asset**
type	asset
id	as67890

id	2a9cfb56yz
entity[0]	xxx123-abcd
entity[1]	yyy456-efgh

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

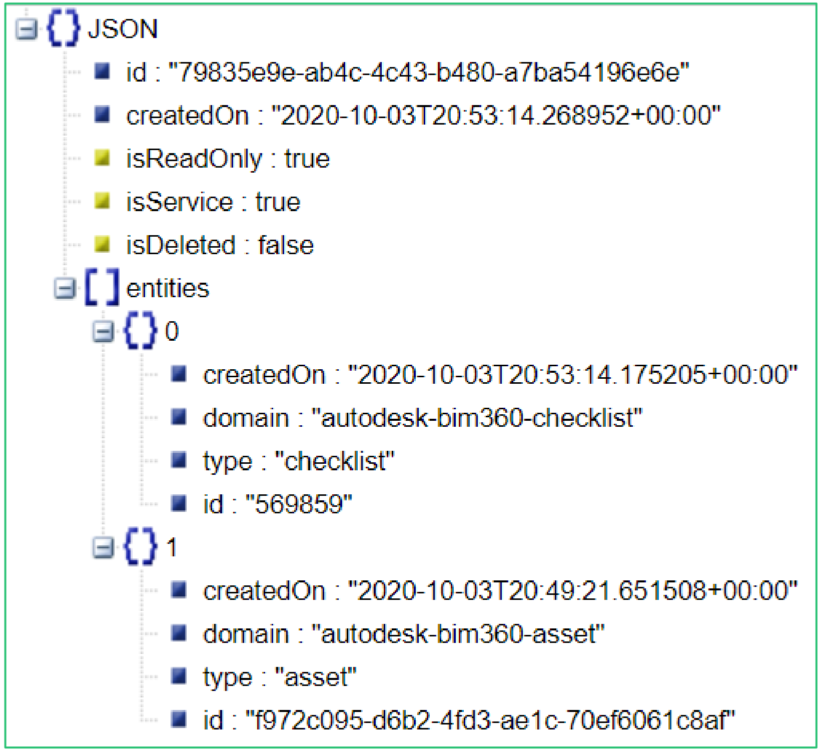
Save and search relationships independent from the implementation detail of each components

9 endpoints

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

Get relationship(s) from id(s)



```
{ } JSON
    id : "79835e9e-ab4c-4c43-b480-a7ba54196e6e"
    createdOn : "2020-10-03T20:53:14.268952+00:00"
    isReadOnly : true
    isService : true
    isDeleted : false
  [ ] entities
    { } 0
        createdOn : "2020-10-03T20:53:14.175205+00:00"
        domain : "autodesk-bim360-checklist"
        type : "checklist"
        id : "569859"
    { } 1
        createdOn : "2020-10-03T20:49:21.651508+00:00"
        domain : "autodesk-bim360-asset"
        type : "asset"
        id : "f972c095-d6b2-4fd3-ae1c-70ef6061c8af"
```

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

Search rel's that match the given criteria (domain, type, ids, date). e.g.,

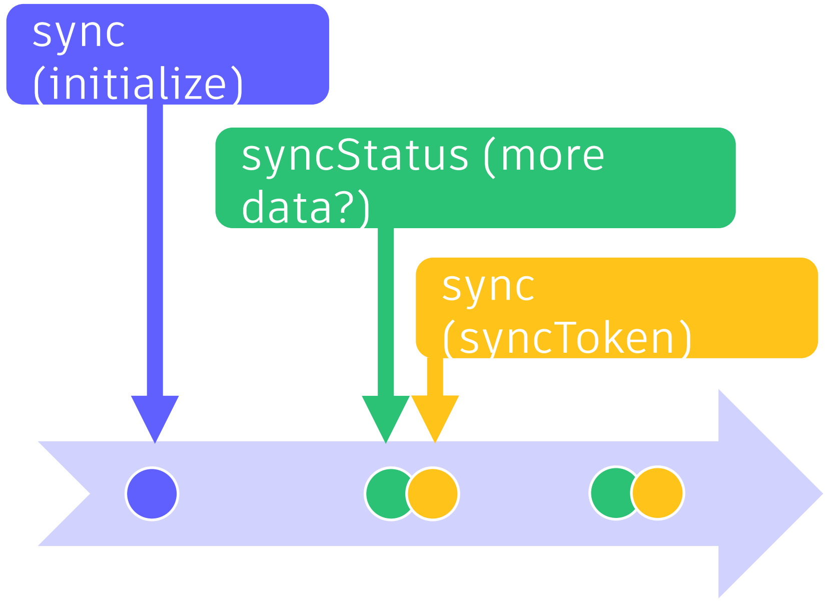
"Give me all the relationships between assets and checklists"

```
GET relationships:/search?
domain=autodesk-bim360-asset
&withDomain=autodesk-bim360-
checklist
```

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

Used to sync rel. data and external repository.



Relationships API - Additions

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

- Retrieves **compatible entity types** to create/delete relationships
- Supported domain/entity types:
  - Asset/asset category
  - Document/File lineage/version
  - Issue
  - Form/form field/form template
  - Photo
  - Etc.
- Unsupported types - may have workaround.
- Expected to be extended overtime

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

```
[
  {
    "domain": "autodesk-bim360-asset",
    "entityTypes": [
      {
        "entityType": "asset",
        "allow": [
          {
            "domain": "autodesk-bim360-issue",
            "entityTypes": [
              "issue"
            ]
          },
          {
            "domain": "autodesk-bim360-documentmanagement",
            "entityTypes": [
              "documentlineage"
            ]
          },
          {
            "domain": "autodesk-construction-photo",
            "entityTypes": [
              "photo"
            ]
          },

          […]
```

Relationships API

Category	Endpoints
Search	GET relationships/:**relationshipId**
	POST relationships:**batch**
	GET relationships:**search**
	POST relationships:**intersect**
Sync	POST relationships:**sync**
	POST relationships:**syncStatus**
Utilities	POST relationships:**writable**
Modify	PUT **relationships**
	POST relationships:**delete**

- **Create** a relationship(2) between two entities.
  - Two entities are writable
  - Entities must exist
  - must have write access to both entities
- **Delete** a given relationship
  - “Soft” delete
  - must have write access to both entities

Relationships API: Developer Resources

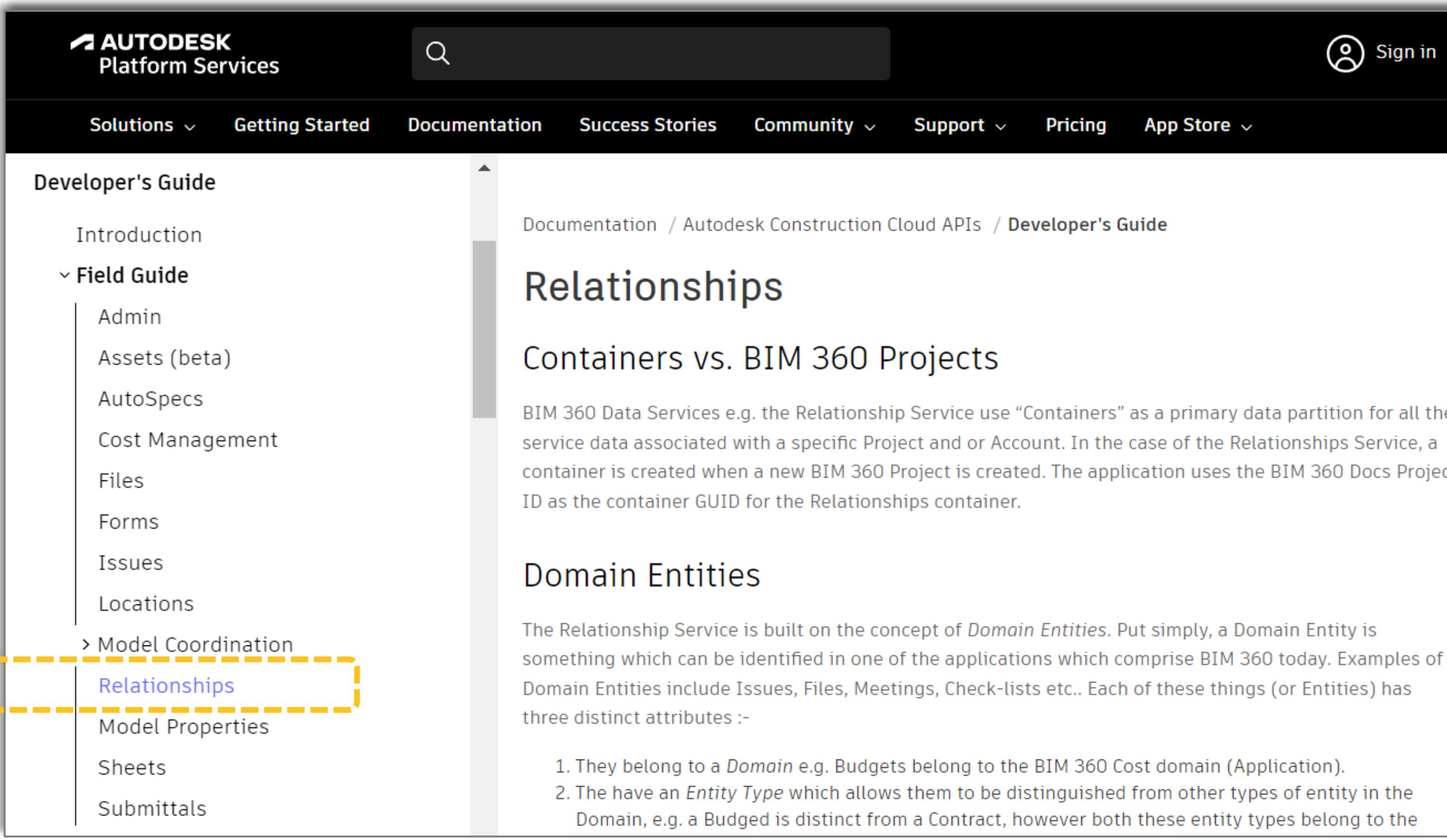
SDK

- .NET Core nuget packages
- JavaScript and Node.js

Relationships API: Developer Resources

Documentation

- Field Guide
- Step by Step Tutorials



Documentation / Autodesk Construction Cloud APIs / Developer's Guide

Relationships

Containers vs. BIM 360 Projects

BIM 360 Data Services e.g. the Relationship Service use "Containers" as a primary data partition for all the service data associated with a specific Project and or Account. In the case of the Relationships Service, a container is created when a new BIM 360 Project is created. The application uses the BIM 360 Docs Projec ID as the container GUID for the Relationships container.

Domain Entities

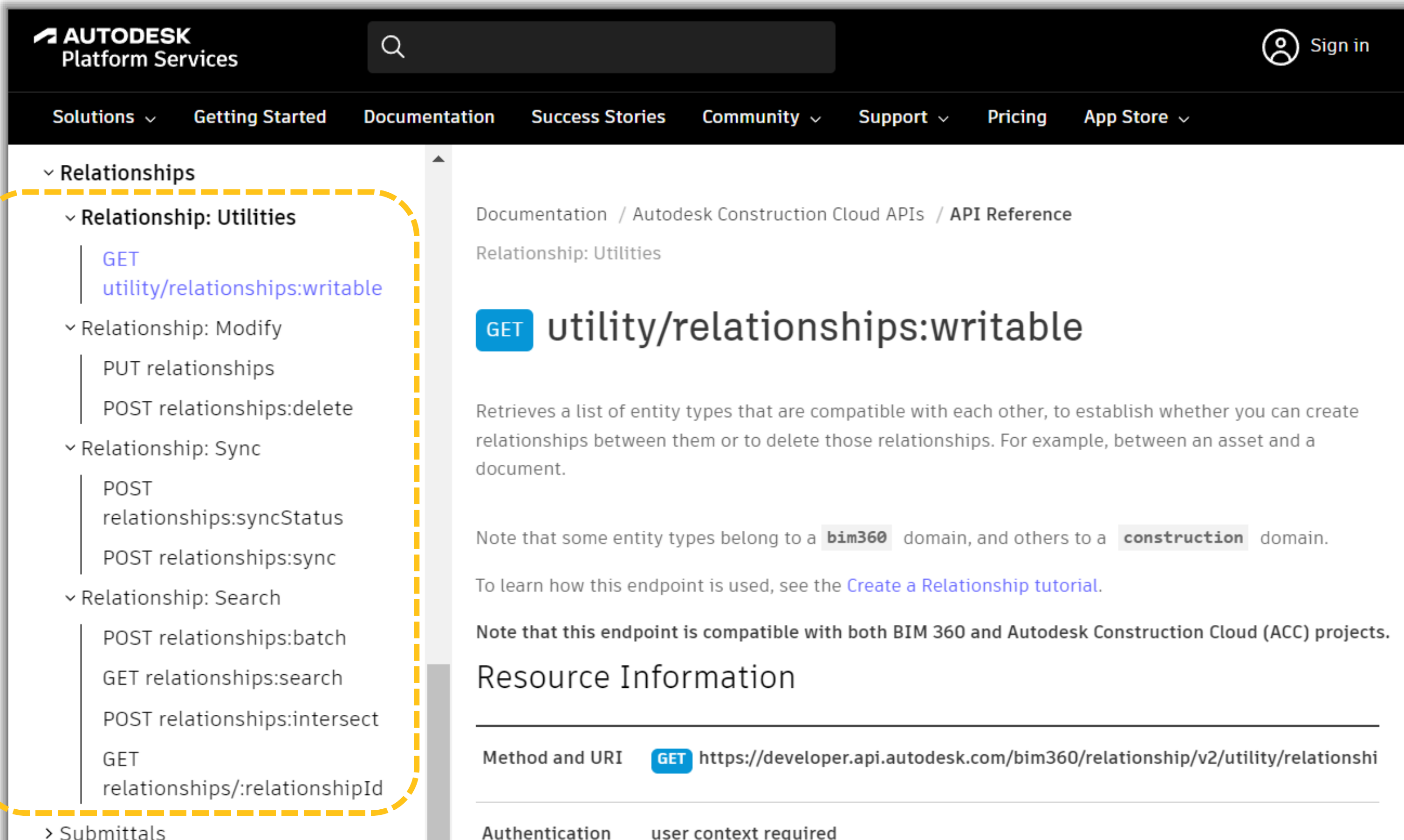
The Relationship Service is built on the concept of *Domain Entities*. Put simply, a Domain Entity is something which can be identified in one of the applications which comprise BIM 360 today. Examples of Domain Entities include Issues, Files, Meetings, Check-lists etc.. Each of these things (or Entities) has three distinct attributes :-

1. They belong to a *Domain* e.g. Budgets belong to the BIM 360 Cost domain (Application).
2. The have an *Entity Type* which allows them to be distinguished from other types of entity in the Domain, e.g. a Budged is distinct from a Contract, however both these entity types belong to the

Relationships API: Developer Resources

Documentation

- Reference Guide



Relationships

Relationship: Utilities

GET utility/relationships:writable

Relationship: Modify

PUT relationships

POST relationships:delete

Relationship: Sync

POST relationships:syncStatus

POST relationships:sync

Relationship: Search

POST relationships:batch

GET relationships:search

POST relationships:intersect

GET relationships/:relationshipId

Submittals

Documentation / Autodesk Construction Cloud APIs / API Reference

Relationship: Utilities

GET utility/relationships:writable

Retrieves a list of entity types that are compatible with each other, to establish whether you can create relationships between them or to delete those relationships. For example, between an asset and a document.

Note that some entity types belong to a `bim360` domain, and others to a `construction` domain.

To learn how this endpoint is used, see the Create a Relationship tutorial.

Note that this endpoint is compatible with both BIM 360 and Autodesk Construction Cloud (ACC) projects.

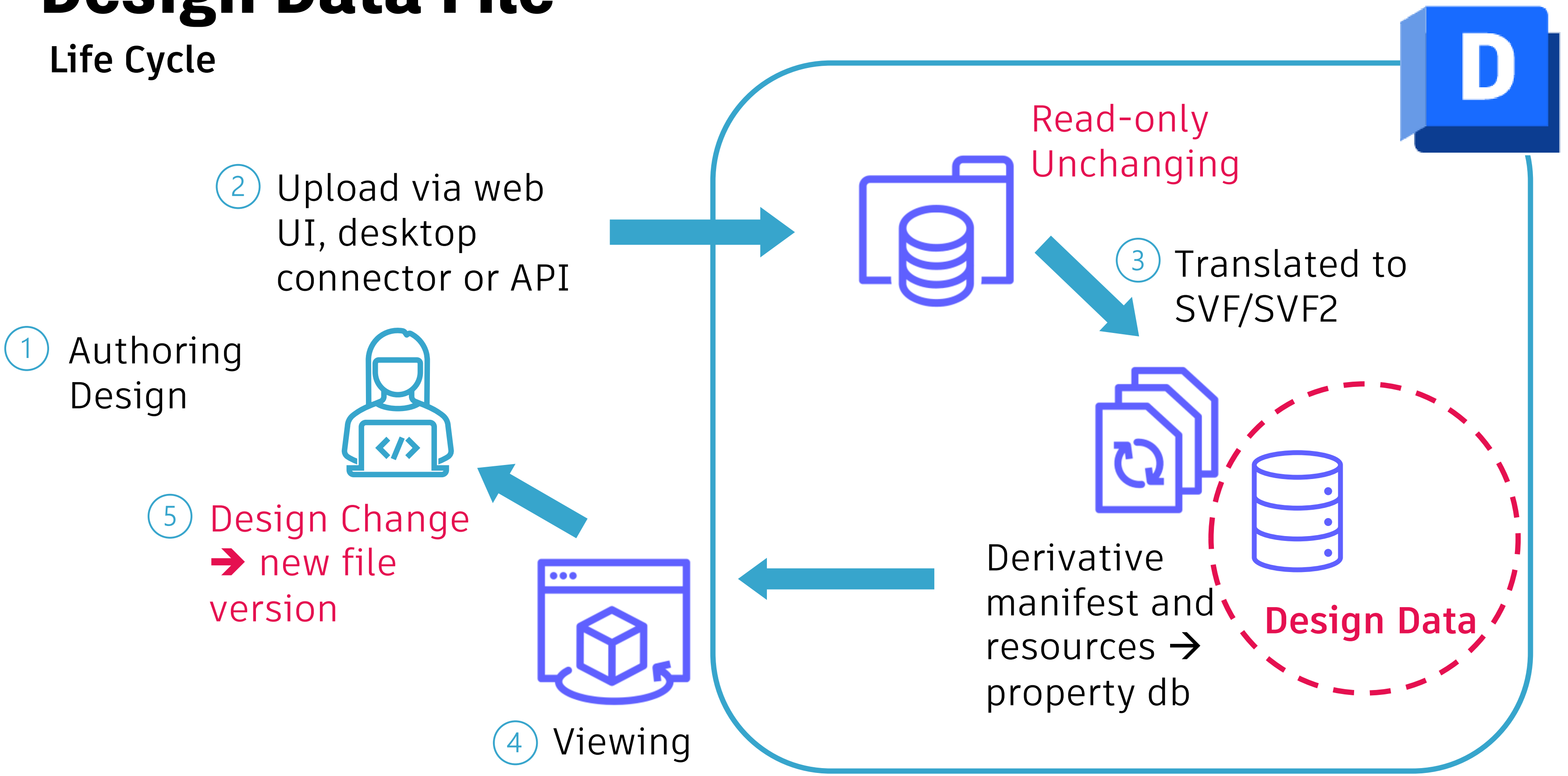
Resource Information

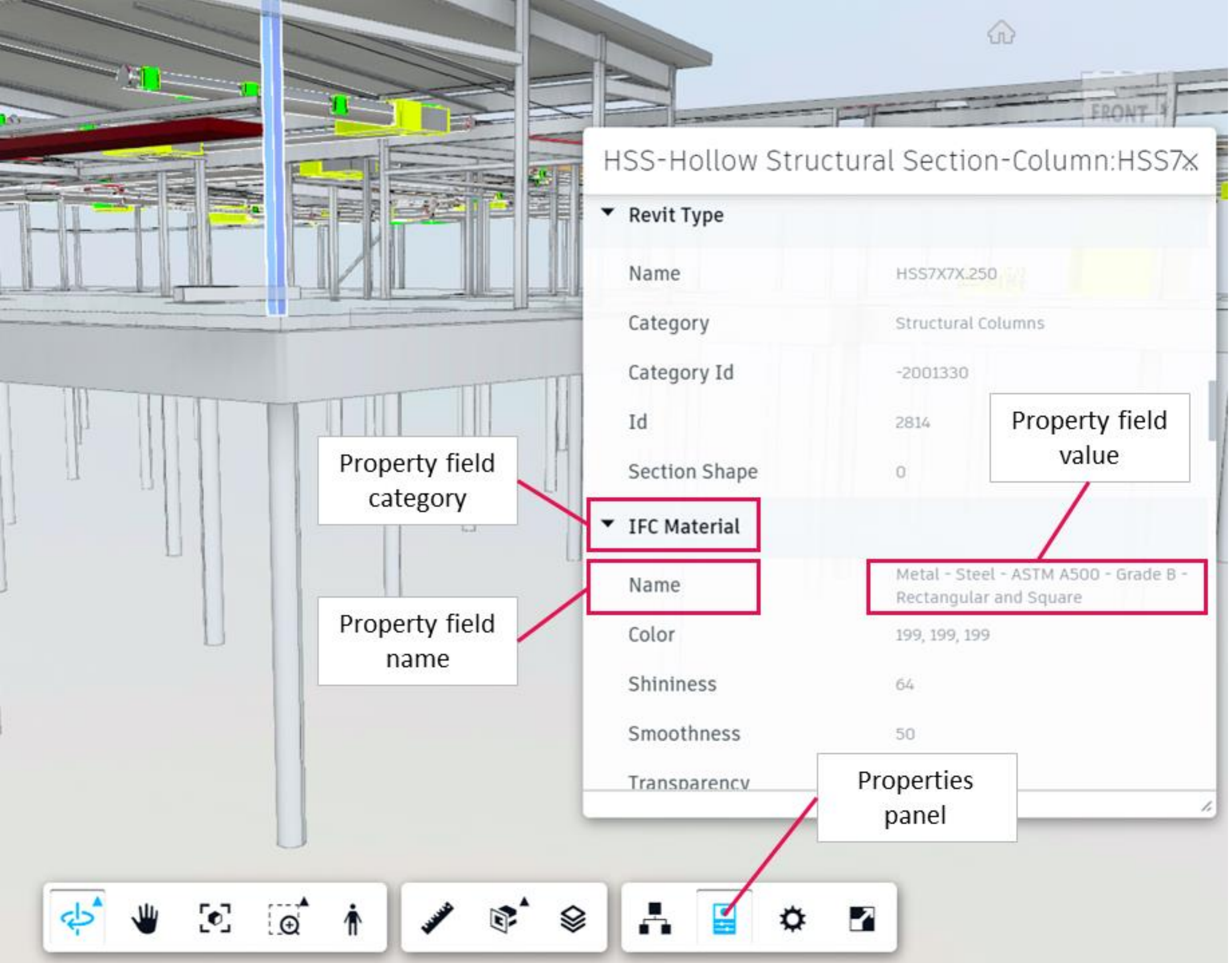
Method and URI	GET https://developer.api.autodesk.com/bim360/relationship/v2/utility/relationshi
Authentication	user context required

Model Properties API

Design Data File

Life Cycle


1 Authoring Design
2 Upload via web UI, desktop connector or API
Read-only Unchanging
3 Translated to SVF/SVF2
Derivative manifest and resources → property db
Design Data
4 Viewing
5 Design Change ➔ new file version



HSS-Hollow Structural Section-Column:HSS7x

▾ Revit Type

Name	HSS7X7X.250
Category	Structural Columns
Category Id	-2001330
Id	2814
Section Shape	0

▾ IFC Material

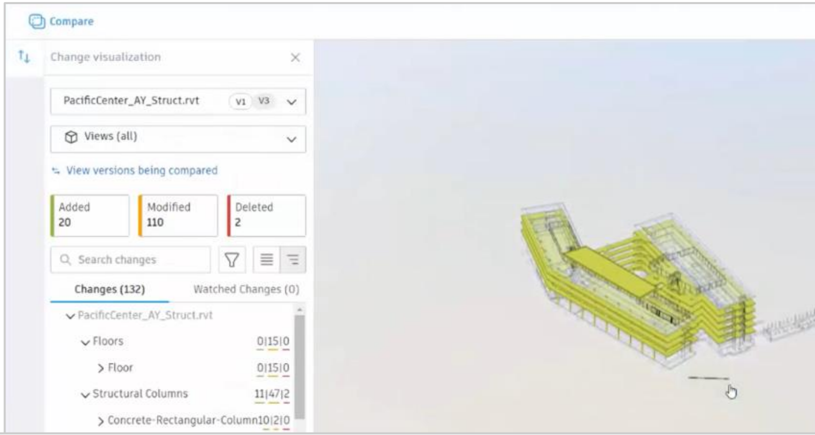
Name	Metal - Steel - ASTM A500 - Grade B - Rectangular and Square
Color	199, 199, 199
Shininess	64
Smoothness	50
Transparency	

Classifying (BIM) SVF2 model properties

- Category
- Name
- Type (e.g., string, double)
- (Optional) Unit of Measure

Model Properties API

- Released Feb 2022
- Autodesk/BIM 360 Docs based products
- Built on PropertyDb from Derivative Services (svf translation)
  - **Index** (Base) – query, filter properties of svf2 objects + extra (viewable bbox)
  - **Diff** – Index + compare two versions
- Used in product
  - Design Collaboration - Change analysis
  - Model Coordination
    - Model property breakdown / advanced filtering
    - Publishing views to Docs
    - Object exclusion
  - Assets – integrating with models


Compare

Change visualization

PacificCenter_AY_Struct.rvt V1 V3

Views (all)

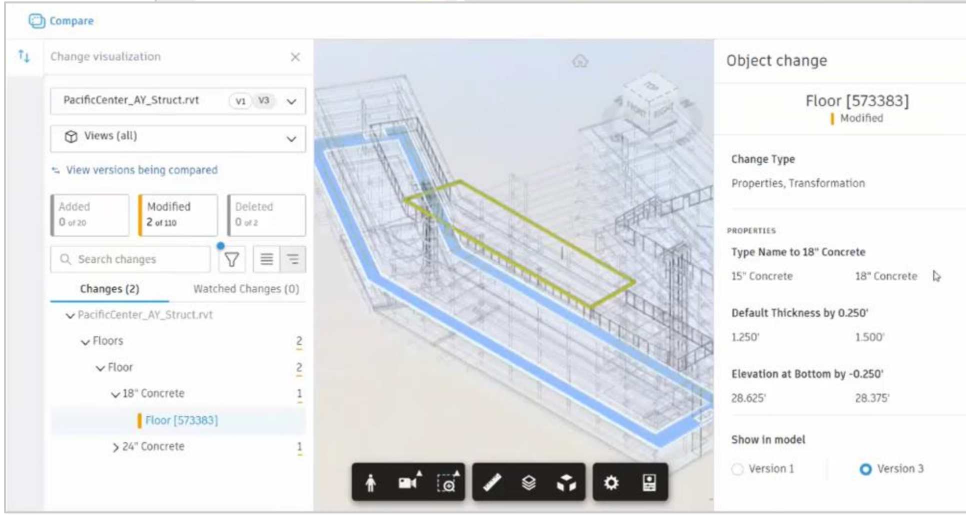
View versions being compared

Added	Modified	Deleted
20	110	2

Search changes

Changes (132) Watched Changes (0)

PacificCenter_AY_Struct.rvt	
Floors	0|15|0
Floor	0|15|0
Structural Columns	11|47|2
Concrete-Rectangular-Column	10|2|0


Compare

Change visualization

PacificCenter_AY_Struct.rvt V1 V3

Views (all)

View versions being compared

Added	Modified	Deleted
0 of 20	2 of 110	0 of 2

Search changes

Changes (2) Watched Changes (0)

PacificCenter_AY_Struct.rvt	
Floors	2
Floor	2
18" Concrete	1
Floor [573383]	
24" Concrete	1

Object change

Floor [573383]
Modified

Change Type
Properties, Transformation

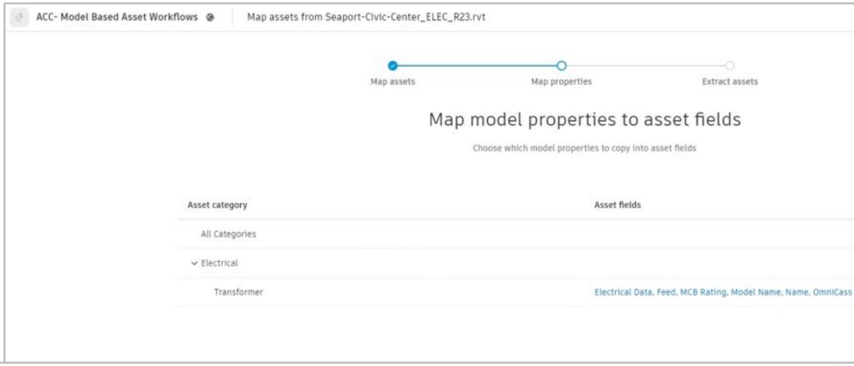
PROPERTIES

Type Name to 18" Concrete	
15" Concrete	18" Concrete
Default Thickness by 0.250'	
1.250'	1.500'
Elevation at Bottom by -0.250'	
28.625'	28.375'

Show in model
Version 1 [✓ Version 3]

Model Properties API

- Released Feb 2022
- Autodesk/BIM 360 Docs based products. US and EMEA
- Built on PropertyDb from Derivative Services (svf translation)
  - **Index** (Base) – query, filter properties of svf2 objects + extra (viewable bbox)
  - **Diff** – Index + compare two versions
- Used in product
  - Design Collaboration - Change analysis
  - Model Coordination
    - Model property breakdown / advanced filtering
    - Publishing views to Docs
    - Object exclusion
  - Assets – integrating with models

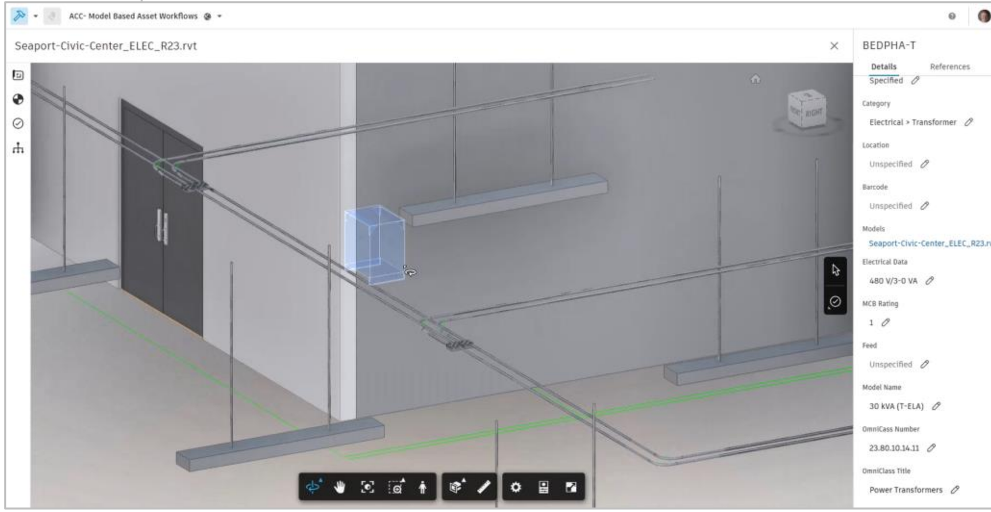

ACC- Model Based Asset Workflows

Map assets from Seaport-Civic-Center_ELEC_R23.rvt

Map model properties to asset fields

Choose which model properties to copy into asset fields

Asset category	Asset fields
All Categories	
Electrical	
Transformer	Electrical Data, Feed, MCB Rating, Model Name, Name, OmniCass


ACC- Model Based Asset Workflows

Seaport-Civic-Center_ELEC_R23.rvt

BEDPHA-T

Specified

Category: Electrical > Transformer

Location: Unspecified

Barcode: Unspecified

Models: Seaport-Civic-Center_ELEC_R23.r

Electrical Data: 480 V/3-0 VA

MCB Rating: 1

Feed: Unspecified

Model Name: 30 kVA (T-ELA)

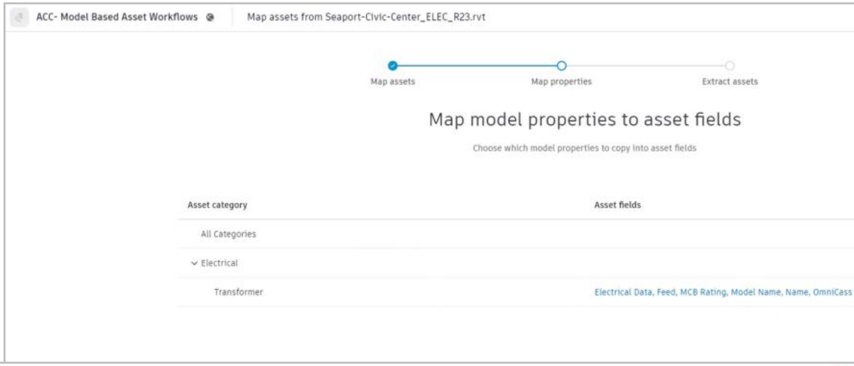
OmniCass Number: 23.80.10.14.11

OmniClass Title: Power Transformers

Model Properties API

Supported Files

- **Index** – any files supported Derivative Service /SVF2
- **Diff** - element id needs to be stable (uniquely identifiable)
  - **RVT**
  - **DWG**
  - **NWC** exported from:
    - Revit and
    - AutoCAD verticals
  - **IFC** exported from:
    - AutoCAD Architecture, MEP, Civil 3D 2018+
    - ARCHCAD
    - Revit
    - MigiCAD for Revit
    - Tekla Structures

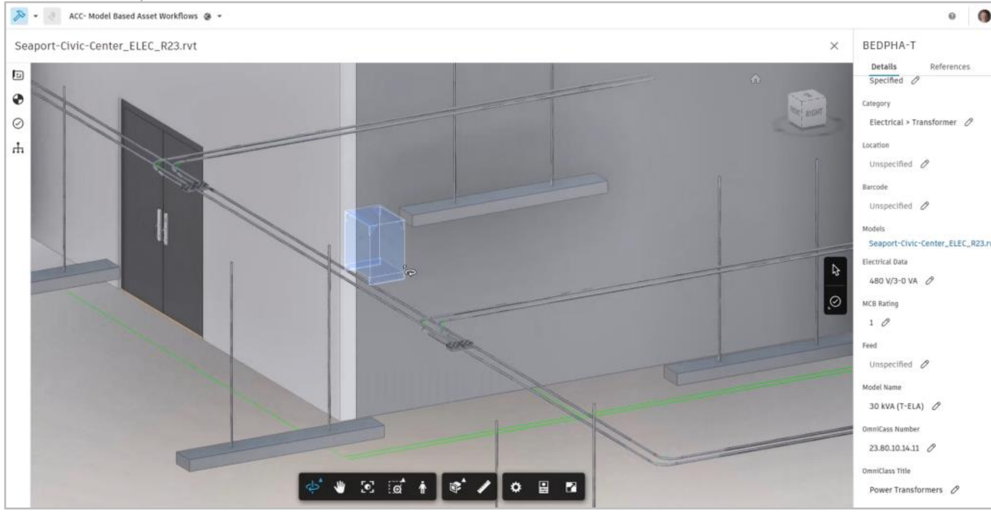


Map assets from Seaport-Civic-Center_ELEC_R23.rvt

Map model properties to asset fields

Choose which model properties to copy into asset fields

Asset category	Asset fields
All Categories	
Electrical	
Transformer	Electrical Data, Feed, MCB Rating, Model Name, Name, OmniCass



Seaport-Civic-Center_ELEC_R23.rvt

BEDPHA-T

Specified

Category: Electrical > Transformer

Location: Unspecified

Barcode: Unspecified

Models: Seaport-Civic-Center_ELEC_R23.r

Electrical Data: 480 V/3-0 VA

MCB Rating: 1

Feed: Unspecified

Model Name: 30 kVA (T-ELA)

OmniCass Number: 23.80.10.14.11

OmniClass Title: Power Transformers

Sample Applications

Model Properties API

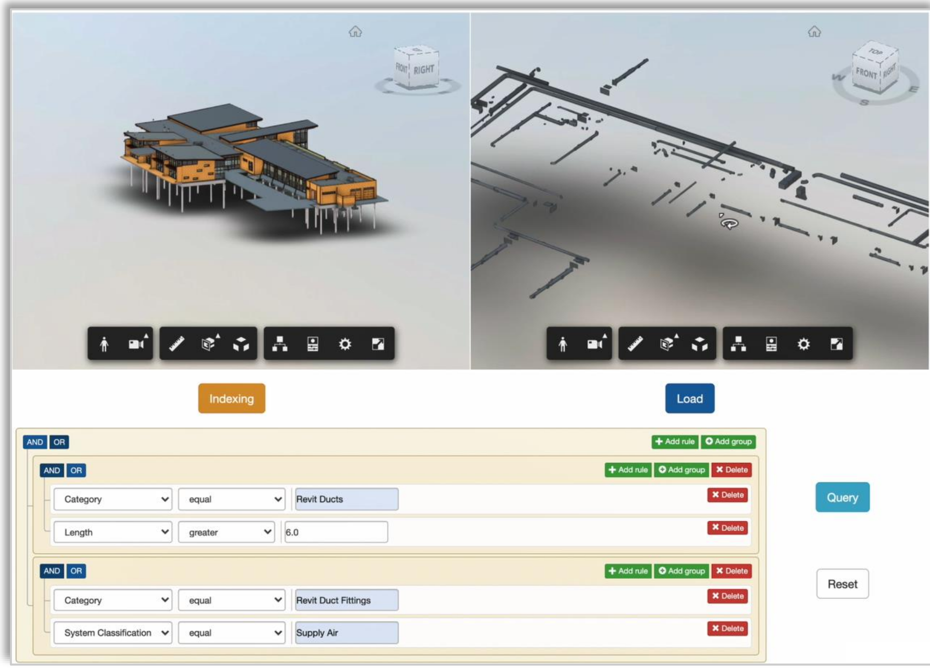
Filter Elements & Partial Model Load

Index

What it does: Filters elements by properties, which can be geometric properties such as length and height. A filter condition is defined as a binary expression form and can be combined by AND/OR. The results are visualized in the Forge viewer.

Code:
https://github.com/autodesk-platform-services/aps-model.properties-elements.filtering

Author: Xiaodong Liang, Autodesk



Category equal Revit Ducts
Length greater 6.0

Category equal Revit Duct Fittings
System Classification equal Supply Air

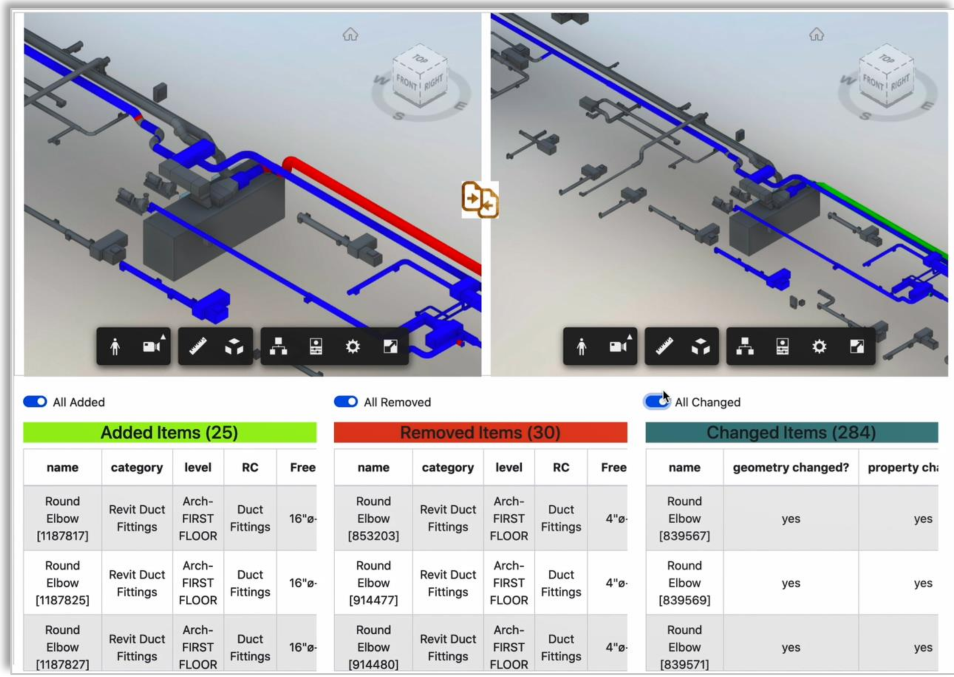
Compare Two Versions

Diff

What it does: Compares two versions of a model and visualizes the differences found in the Forge Viewer. Differences can be in geometries and properties. Elements may be added, modified and removed.

Code:
https://github.com/autodesk-platform-services/aps-model.properties-versions.difference

Author: Xiaodong Liang, Autodesk



Added Items (25)

name	category	level	RC	Free
Round Elbow [1187817]	Revit Duct Fittings	Arch-FIRST FLOOR	Duct Fittings	16"ø-
Round Elbow [1187825]	Revit Duct Fittings	Arch-FIRST FLOOR	Duct Fittings	16"ø-
Round Elbow [1187827]	Revit Duct Fittings	Arch-FIRST FLOOR	Duct Fittings	16"ø-

Removed Items (30)

name	category	level	RC	Free
Round Elbow [853203]	Revit Duct Fittings	Arch-FIRST FLOOR	Duct Fittings	4"ø-
Round Elbow [914477]	Revit Duct Fittings	Arch-FIRST FLOOR	Duct Fittings	4"ø-
Round Elbow [914480]	Revit Duct Fittings	Arch-FIRST FLOOR	Duct Fittings	4"ø-

Changed Items (284)

name	geometry changed?	property ch
Round Elbow [839567]	yes	yes
Round Elbow [839569]	yes	yes
Round Elbow [839571]	yes	yes

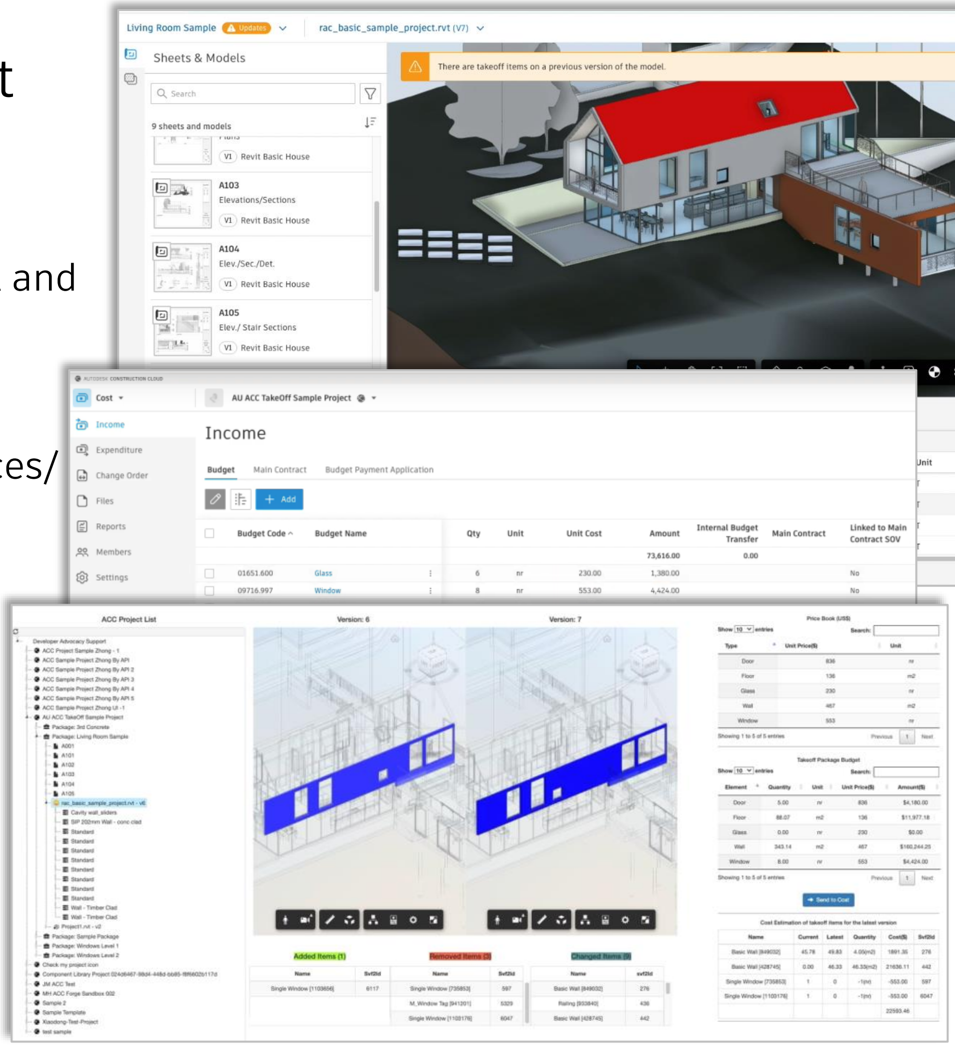
Change Analysis in Takeoff and Cost

Application of **Diff** in Estimate

What it does: Identifies changes in takeoff items, compares current and previous versions of a model and visualizes the differences in the Viewer. Update the budgets data in Cost module.

Code: https://github.com/autodesk-platform-services/aps-acc-takeoff-versions-compare-cost

Author: Zhong Wu, Autodesk



There are takeoff items on a previous version of the model.

Income

Budget Code	Budget Name	Qty	Unit	Unit Cost	Amount	Internal Budget Transfer	Main Contract	Linked to Main Contract SOV
					73,616.00	0.00		
01651.600	Glass	6	nr	230.00	1,380.00			No
09716.997	Window	8	nr	553.00	4,424.00			No

Price Book (US$)

Type	Unit Price($)	Unit
Door	836	nr
Floor	136	m2
Glass	230	nr
Wall	467	m2
Window	553	nr

Showing 1 to 5 of 5 entries

Takeoff Package Budget

Element	Quantity	Unit	Unit Price($)	Amount($)
Door	5.00	nr	836	$4,180.00
Floor	88.07	m2	136	$11,977.18
Glass	0.00	nr	230	$0.00
Wall	343.14	m2	467	$160,244.25
Window	8.00	nr	553	$4,424.00

Showing 1 to 5 of 5 entries

Cost Estimation of takeoff items for the latest version

Name	Current	Latest	Quantity	Cost($)	Svf2Id
Basic Wall [849032]	45.78	49.83	4.05(m2)	1891.35	276
Basic Wall [428745]	0.00	46.33	46.33(m2)	21636.11	442
Single Window [735853]	1	0	-1(nr)	-553.00	597
Single Window [1103176]	1	0	-1(nr)	-553.00	6047
				22593.46	

Added Items (1)

Name	Svf2Id
Single Window [1103656]	6117

Removed Items (3)

Name	Svf2Id
Single Window [735853]	597
M_Window Tag [941201]	5329
Single Window [1103176]	6047

Changed Items (9)

Name	svf2Id
Basic Wall [849032]	276
Railing [933840]	436
Basic Wall [428745]	442

How does it work?

Model Properties API

How does it work?

Basic Workflow

Model Derivative

Design Data

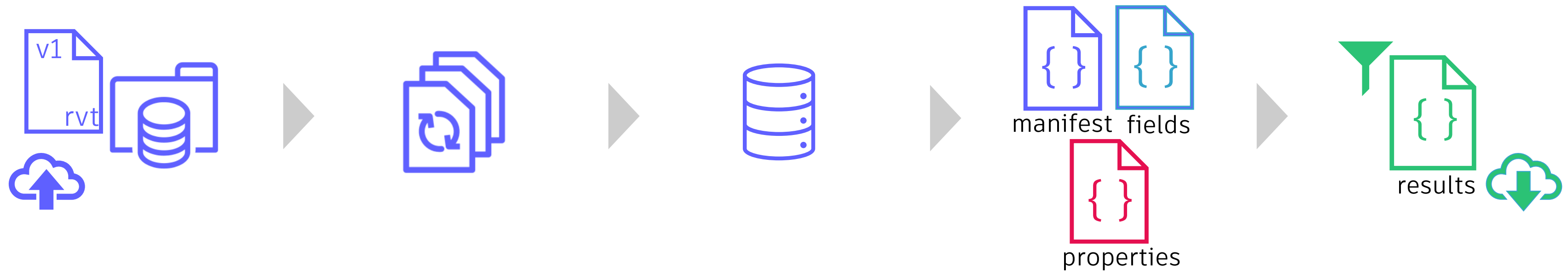
DM/Docs	Model Derivative	PropertyDb
Files uploaded to Docs via UI, Desktop Connector and API	Files are translated to svf/svf2. In Docs, translation is triggered automatically.	Property database, manifest, resources.

How does it work?

Basic Workflow

Model Derivative

Model Properties



Design Data

DM/Docs

Files uploaded to Docs via UI, Desktop Connector and API

Model Derivative

Files are translated to svf/svf2. In Docs, translation is triggered automatically.

PropertyDb

Property database, manifest, resources.

Index

Basic index generates three downloadable JSON resources:

- manifest
- fields
- properties

Query

results: JSON files matching the query filter.

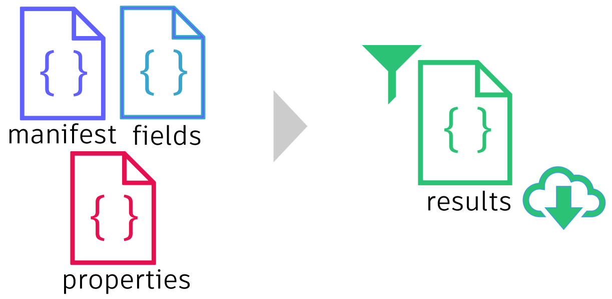
How does it work?

Resources Generated

Resource	Type	Description
manifest	JSON	Manifest for index or query detailing the seed files, svf2 propertyDb used to generate index rows
fields	NDJSON	The set of unique fields (property types) extracted for an index or query
properties	NDJSON	The raw objects property values for the index.
results	NDJSON	The object property results from a query executed.

NDJSON = new-line delimited JSON

Model Properties



Index

Basic index generates three downloadable JSON resources:

- manifest
- fields
- properties

Query

results: JSON files matching the query filter.

How does it work?

Index Endpoints

	Endpoints		
Index	POST	**indexes:batch-status**	
	GET	indexes/**:indexId**	
	GET	indexes/:indexId/**manifest**	⬇
	GET	indexes/:indexId/**fields**	⬇
	GET	indexes/:indexId/**properties**	⬇
Query	POST	indexes/:indexId/**queries**	
	GET	indexes/:indexId/queries/**:queryId**	
	GET	indexes/:indexId/queries/:queryId/**properties**	⬇

- 8 endpoints

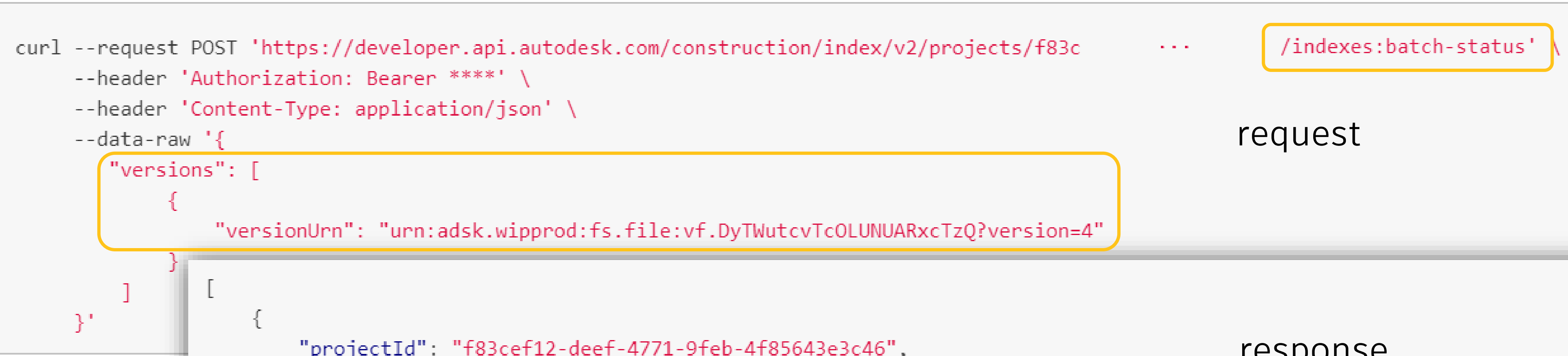
How does it work?

Index Endpoints for Creation

	Endpoints		
Index	POST	**indexes:batch-status**	
	GET	indexes/**:indexId**	
	GET	indexes/:indexId/**manifest**	⬇
	GET	indexes/:indexId/**fields**	⬇
	GET	indexes/:indexId/**properties**	⬇
Query	POST	indexes/:indexId/**queries**	
	GET	indexes/:indexId/queries/**:queryId**	
	GET	indexes/:indexId/queries/:queryId/**properties**	⬇

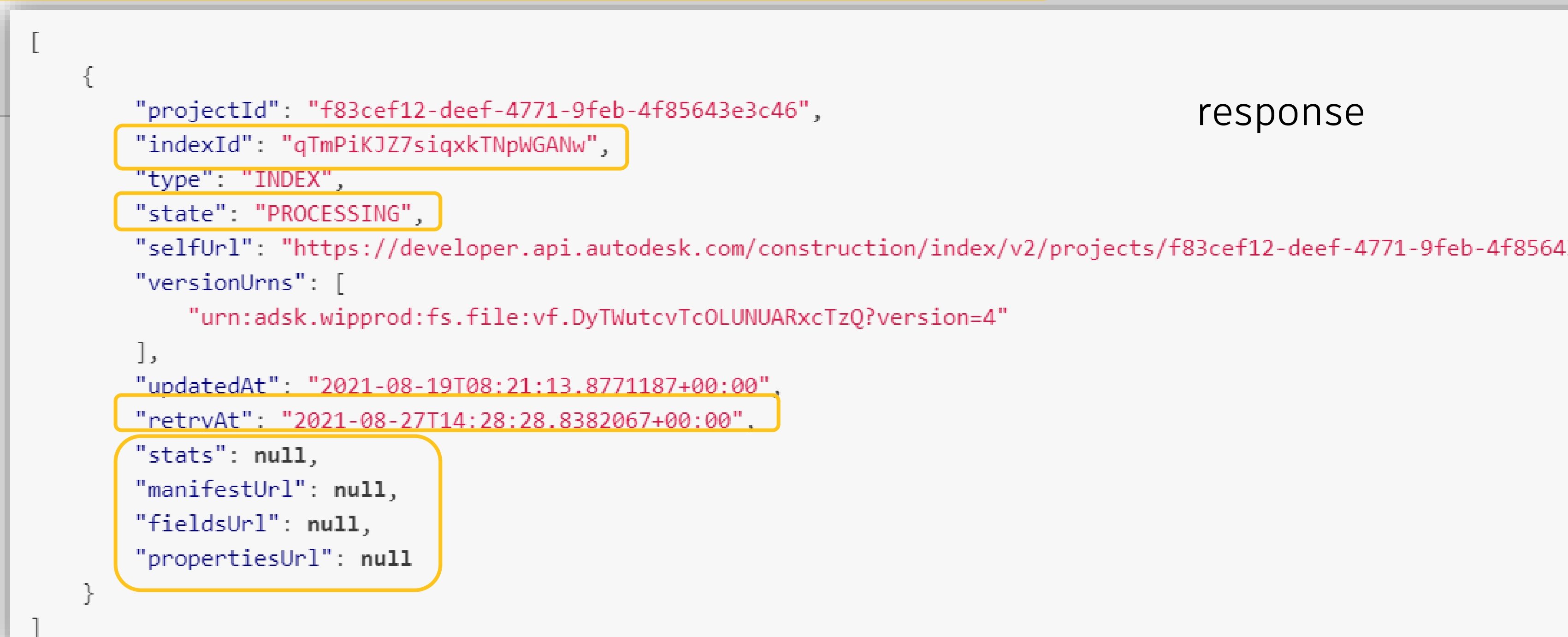
- Create basic index – "lazy"
  - First time – start the indexing job and cache the results
  - Once executed – use the cache
  - Cached 30 days since the last used
- Poll for progress
  - state: PROCESSING, FINISHED, FAILED
- Response JSON is identical
- stats: objects (# of object)
- Create 3 downloadable json.gz resources
  - Manifest, fields, properties

Ex. Create basic index – POST indexes:batch-status



```
curl --request POST 'https://developer.api.autodesk.com/construction/index/v2/projects/f83c   ···   /indexes:batch-status' \
     --header 'Authorization: Bearer ****' \
     --header 'Content-Type: application/json' \
     --data-raw '{
       "versions": [
            {
                "versionUrn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4"
            }
         ]
     }'
```

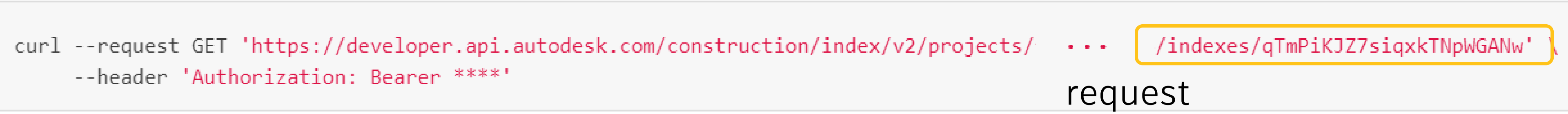
request



```
[
    {
        "projectId": "f83cef12-deef-4771-9feb-4f85643e3c46",
        "indexId": "qTmPiKJZ7siqxkTNpWGANw",
        "type": "INDEX",
        "state": "PROCESSING",
        "selfUrl": "https://developer.api.autodesk.com/construction/index/v2/projects/f83cef12-deef-4771-9feb-4f8564
        "versionUrns": [
            "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4"
        ],
        "updatedAt": "2021-08-19T08:21:13.8771187+00:00",
        "retryAt": "2021-08-27T14:28:28.8382067+00:00",
        "stats": null,
        "manifestUrl": null,
        "fieldsUrl": null,
        "propertiesUrl": null
    }
]
```

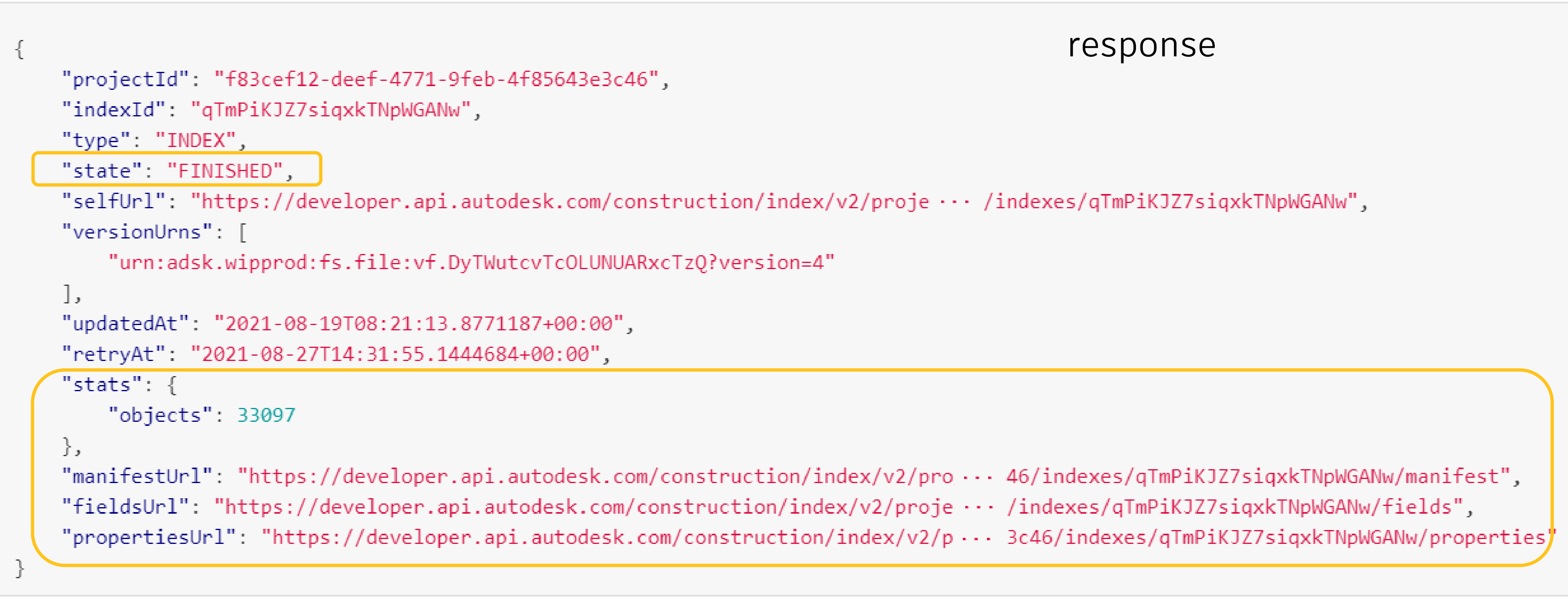
response

Ex. Polling for Progress - GET indexes/:indexId



```
curl --request GET 'https://developer.api.autodesk.com/construction/index/v2/projects/  ···  /indexes/qTmPiKJZ7siqxkTNpWGANw' \
     --header 'Authorization: Bearer ****'
```

request



response

```
{
    "projectId": "f83cef12-deef-4771-9feb-4f85643e3c46",
    "indexId": "qTmPiKJZ7siqxkTNpWGANw",
    "type": "INDEX",
    "state": "FINISHED",
    "selfUrl": "https://developer.api.autodesk.com/construction/index/v2/proje ··· /indexes/qTmPiKJZ7siqxkTNpWGANw",
    "versionUrns": [
        "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4"
    ],
    "updatedAt": "2021-08-19T08:21:13.8771187+00:00",
    "retryAt": "2021-08-27T14:31:55.1444684+00:00",
    "stats": {
        "objects": 33097
    },
    "manifestUrl": "https://developer.api.autodesk.com/construction/index/v2/pro ··· 46/indexes/qTmPiKJZ7siqxkTNpWGANw/manifest",
    "fieldsUrl": "https://developer.api.autodesk.com/construction/index/v2/proje ··· /indexes/qTmPiKJZ7siqxkTNpWGANw/fields",
    "propertiesUrl": "https://developer.api.autodesk.com/construction/index/v2/p ··· 3c46/indexes/qTmPiKJZ7siqxkTNpWGANw/properties"
}
```

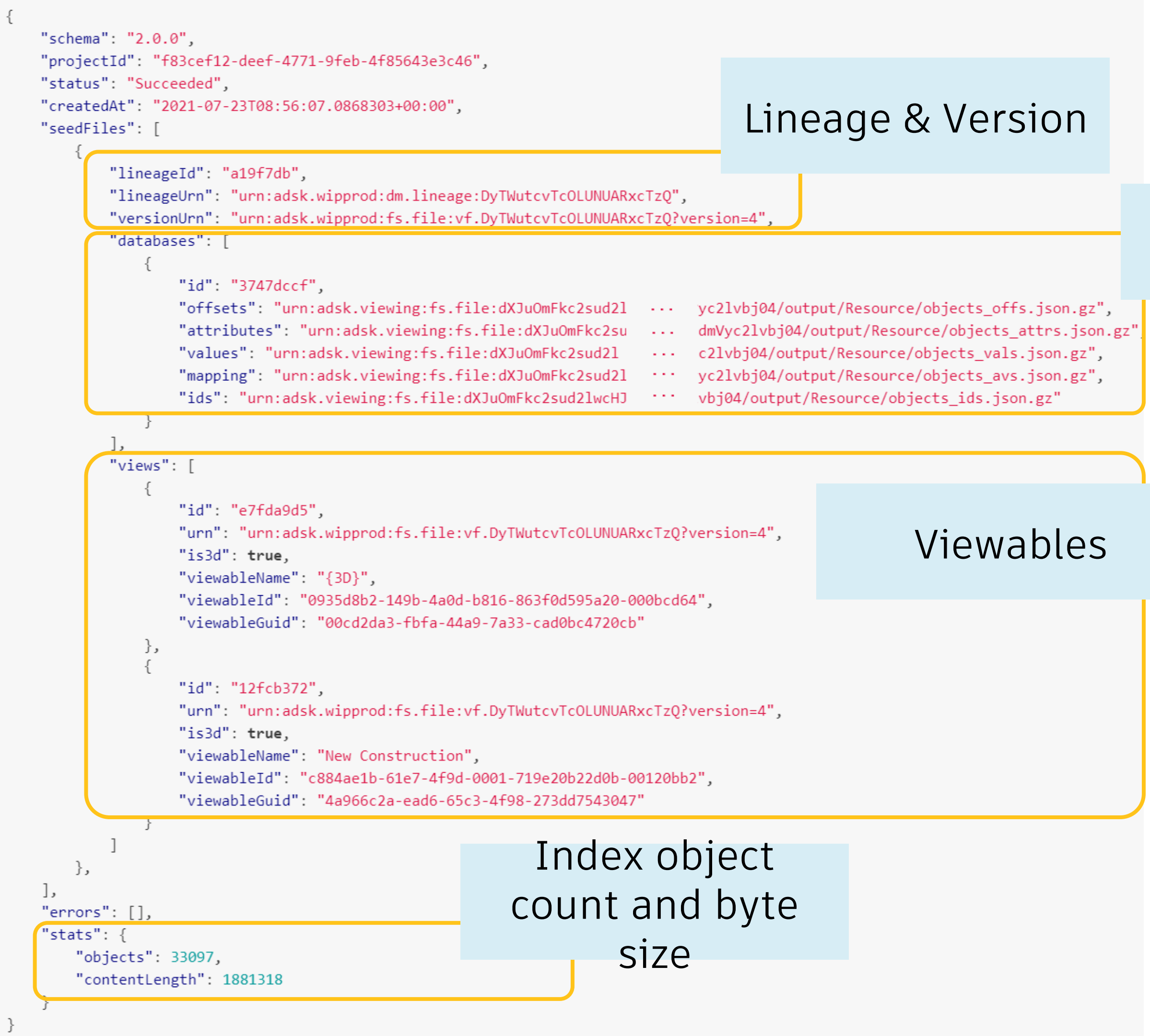
How does it work?

Index Endpoints for Download

	Endpoints		
Index	POST	**indexes:batch-status**	
	GET	indexes/**:indexId**	
	GET	indexes/:indexId/**manifest**	
	GET	indexes/:indexId/**fields**	
	GET	indexes/:indexId/**properties**	
Query	POST	indexes/:indexId/**queries**	
	GET	indexes/:indexId/queries/**:queryId**	
	GET	indexes/:indexId/queries/:queryId/**properties**	

- (Optional) download
  - manifest
  - fields
  - properties

Manifest (.json)



```
{
    "schema": "2.0.0",
    "projectId": "f83cef12-deef-4771-9feb-4f85643e3c46",
    "status": "Succeeded",
    "createdAt": "2021-07-23T08:56:07.0868303+00:00",
    "seedFiles": [
        {
            "lineageId": "a19f7db",
            "lineageUrn": "urn:adsk.wipprod:dm.lineage:DyTWutcvTcOLUNUARxcTzQ",
            "versionUrn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4",
            "databases": [
                {
                    "id": "3747dccf",
                    "offsets": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2l   ···   yc2lvbj04/output/Resource/objects_offs.json.gz",
                    "attributes": "urn:adsk.viewing:fs.file:dXJuOmFkc2su   ···   dmVyc2lvbj04/output/Resource/objects_attrs.json.gz",
                    "values": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2l    ···   c2lvbj04/output/Resource/objects_vals.json.gz",
                    "mapping": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2l   ···   yc2lvbj04/output/Resource/objects_avs.json.gz",
                    "ids": "urn:adsk.viewing:fs.file:dXJuOmFkc2sud2lwcHJ   ···   vbj04/output/Resource/objects_ids.json.gz"
                }
            ],
            "views": [
                {
                    "id": "e7fda9d5",
                    "urn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4",
                    "is3d": true,
                    "viewableName": "{3D}",
                    "viewableId": "0935d8b2-149b-4a0d-b816-863f0d595a20-000bcd64",
                    "viewableGuid": "00cd2da3-fbfa-44a9-7a33-cad0bc4720cb"
                },
                {
                    "id": "12fcb372",
                    "urn": "urn:adsk.wipprod:fs.file:vf.DyTWutcvTcOLUNUARxcTzQ?version=4",
                    "is3d": true,
                    "viewableName": "New Construction",
                    "viewableId": "c884ae1b-61e7-4f9d-0001-719e20b22d0b-00120bb2",
                    "viewableGuid": "4a966c2a-ead6-65c3-4f98-273dd7543047"
                }
            ]
        },
    ],
    "errors": [],
    "stats": {
        "objects": 33097,
        "contentLength": 1881318
    }
}
```

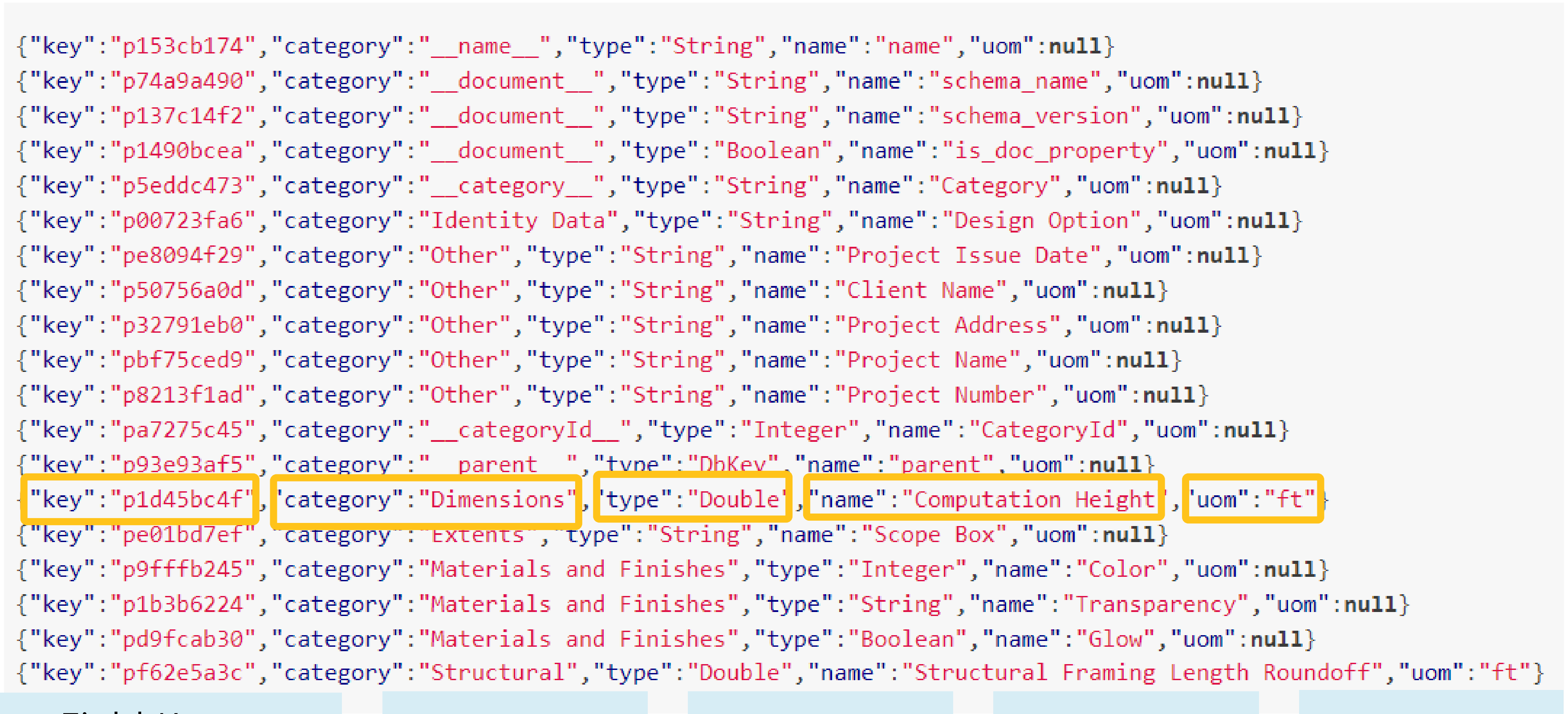
Lineage & Version

SVF2 Prop DB Resource URNs

Viewables

Index object count and byte size

Index Fields (json.gz)



```
{"key":"p153cb174","category":"__name__","type":"String","name":"name","uom":null}
{"key":"p74a9a490","category":"__document__","type":"String","name":"schema_name","uom":null}
{"key":"p137c14f2","category":"__document__","type":"String","name":"schema_version","uom":null}
{"key":"p1490bcea","category":"__document__","type":"Boolean","name":"is_doc_property","uom":null}
{"key":"p5eddc473","category":"__category__","type":"String","name":"Category","uom":null}
{"key":"p00723fa6","category":"Identity Data","type":"String","name":"Design Option","uom":null}
{"key":"pe8094f29","category":"Other","type":"String","name":"Project Issue Date","uom":null}
{"key":"p50756a0d","category":"Other","type":"String","name":"Client Name","uom":null}
{"key":"p32791eb0","category":"Other","type":"String","name":"Project Address","uom":null}
{"key":"pbf75ced9","category":"Other","type":"String","name":"Project Name","uom":null}
{"key":"p8213f1ad","category":"Other","type":"String","name":"Project Number","uom":null}
{"key":"pa7275c45","category":"__categoryId__","type":"Integer","name":"CategoryId","uom":null}
{"key":"p93e93af5","category":"__parent__","type":"DbKey","name":"parent","uom":null}
{"key":"p1d45bc4f","category":"Dimensions","type":"Double","name":"Computation Height","uom":"ft"}
{"key":"pe01bd7ef","category":"Extents","type":"String","name":"Scope Box","uom":null}
{"key":"p9fffb245","category":"Materials and Finishes","type":"Integer","name":"Color","uom":null}
{"key":"p1b3b6224","category":"Materials and Finishes","type":"String","name":"Transparency","uom":null}
{"key":"pd9fcab30","category":"Materials and Finishes","type":"Boolean","name":"Glow","uom":null}
{"key":"pf62e5a3c","category":"Structural","type":"Double","name":"Structural Framing Length Roundoff","uom":"ft"}
```

Field Key → SQL column name

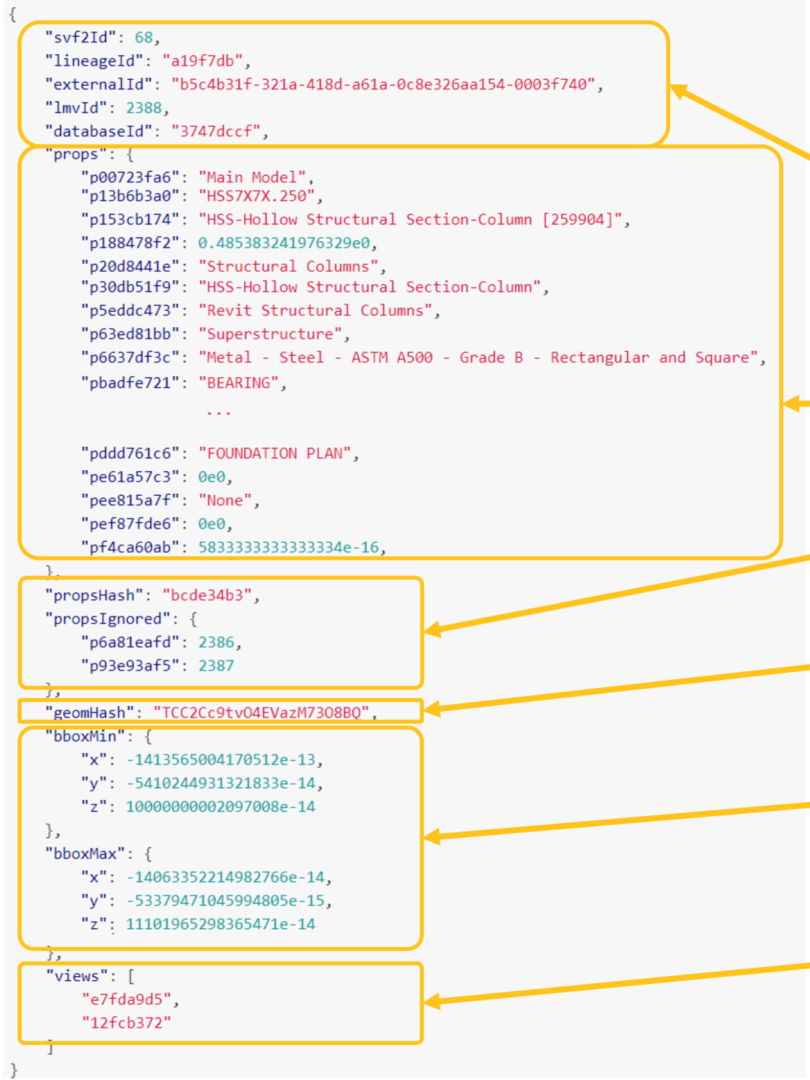
Category

Type

Name

UOM
(Unit of Measurement)

Basic Properties (json.gz)



```json
{
    "svf2Id": 68,
    "lineageId": "a19f7db",
    "externalId": "b5c4b31f-321a-418d-a61a-0c8e326aa154-0003f740",
    "lmvId": 2388,
    "databaseId": "3747dccf",
    "props": {
        "p00723fa6": "Main Model",
        "p13b6b3a0": "HSS7X7X.250",
        "p153cb174": "HSS-Hollow Structural Section-Column [259904]",
        "p188478f2": 0.485383241976329e0,
        "p20d8441e": "Structural Columns",
        "p30db51f9": "HSS-Hollow Structural Section-Column",
        "p5eddc473": "Revit Structural Columns",
        "p63ed81bb": "Superstructure",
        "p6637df3c": "Metal - Steel - ASTM A500 - Grade B - Rectangular and Square",
        "pbadfe721": "BEARING",
                    ...

        "pddd761c6": "FOUNDATION PLAN",
        "pe61a57c3": 0e0,
        "pee815a7f": "None",
        "pef87fde6": 0e0,
        "pf4ca60ab": 5833333333333334e-16,
    },
    "propsHash": "bcde34b3",
    "propsIgnored": {
        "p6a81eafd": 2386,
        "p93e93af5": 2387
    },
    "geomHash": "TCC2Cc9tvO4EVazM73O8BQ",
    "bboxMin": {
        "x": -1413565004170512e-13,
        "y": -5410244931321833e-14,
        "z": 10000000002097008e-14
    },
    "bboxMax": {
        "x": -14063352214982766e-14,
        "y": -53379471045994805e-15,
        "z": 11101965298365471e-14
    },
    "views": [
        "e7fda9d5",
        "12fcb372"
    ]
}
```

Object IDs, keys for index manifest JSON. (Type and ChangeType IF diff index)

Index field property values

Property hash + properties ignored when calculating the hash

SVF2 geometry hash

Bounding box min/max for viewable object

Viewables containing the object (manifest keys to viewable)

How does it work?

Index Endpoints for Query

	Endpoints		
Index	POST	**indexes:batch-status**	
	GET	indexes/**:indexId**	
	GET	indexes/:indexId/**manifest**	
	GET	indexes/:indexId/**fields**	
	GET	indexes/:indexId/**properties**	
Query	POST	indexes/:indexId/**queries**	
	GET	indexes/:indexId/queries/**:queryId**	
	GET	indexes/:indexId/queries/:queryId/**properties**	

- Build and run query.
  - Index queries are described using custom JSON schema, (which is converted to a filter expression. AWS S3 Select)
  - Columns can be restricted. Can use alias (have different header)
- Poll for progress.
  - state: PROCESSING, FINISHED, FAILED

Sample Query: Get Revit Classification with Column Transform

Index Fields

```
// Forge viewer element display name field
{"key":"p153cb174","category":"__name__","type":"String","name":"name","uom":null}
// Revit category name field
{"key":"p20d8441e","category":"__category__","type":"String","name":"_RC","uom":null}
// Revit family name field
{"key":"p30db51f9","category":"__category__","type":"String","name":"_RFN","uom":null}
// Revit type name field
{"key":"p13b6b3a0","category":"__category__","type":"String","name":"_RFT","uom":null}
```

Query

```
{
    "query": {
        "$and": [
            { "$notnull": "s.props.p20d8441e" },
            { "$notnull": "s.props.p30db51f9" },
            { "$notnull": "s.props.p13b6b3a0" },
            { "$gt": [{ "$count": "s.views" }, 0] }
        ]
    },
    "columns": {
        "s.svf2Id": true,
        "lmvName": "s.props.p153cb174",
        "revitCategory": "s.props.p20d8441e",
        "revitFamily": "s.props.p30db51f9",
        "revitType": "s.props.p13b6b3a0",
        "s.views": true
    }
}
```

Row has Revit classification

Views array has count more than 0

Columns define alias

=

Equivalent in S3 SQL

```
select
    s.svf2Id,
    s.props.p153cb174 as lmvName,
    s.props.p20d8441e as revitCategory,
    s.props.p30db51f9 as revitFamily,
    s.props.p13b6b3a0 as revitType,
    s.views
from S3Object[*] s
where
    s.props.p20d8441e is not null and
    s.props.p30db51f9 is not null and
    s.props.p13b6b3a0 is not null and
    count(s.views) > 0
```

How does it work?

Index Endpoint to Download Query Results

	Endpoints		
Index	POST	**indexes:batch-status**	
	GET	indexes/**:indexId**	
	GET	indexes/:indexId/**manifest**	⬇
	GET	indexes/:indexId/**fields**	⬇
	GET	indexes/:indexId/**properties**	⬇
Query	POST	indexes/:indexId/**queries**	
	GET	indexes/:indexId/queries/**:queryId**	
	GET	indexes/:indexId/queries/:queryId/**properties**	⬇

- Download the query results
  - use the queryResultsUrl in query call or query id to download the index rows which match the submitted query expression.
  - Result: line delimited JSON
  - a sub-set of the property index rows
  - Format is the exactly the same as properties we saw earlier.

How does it work?

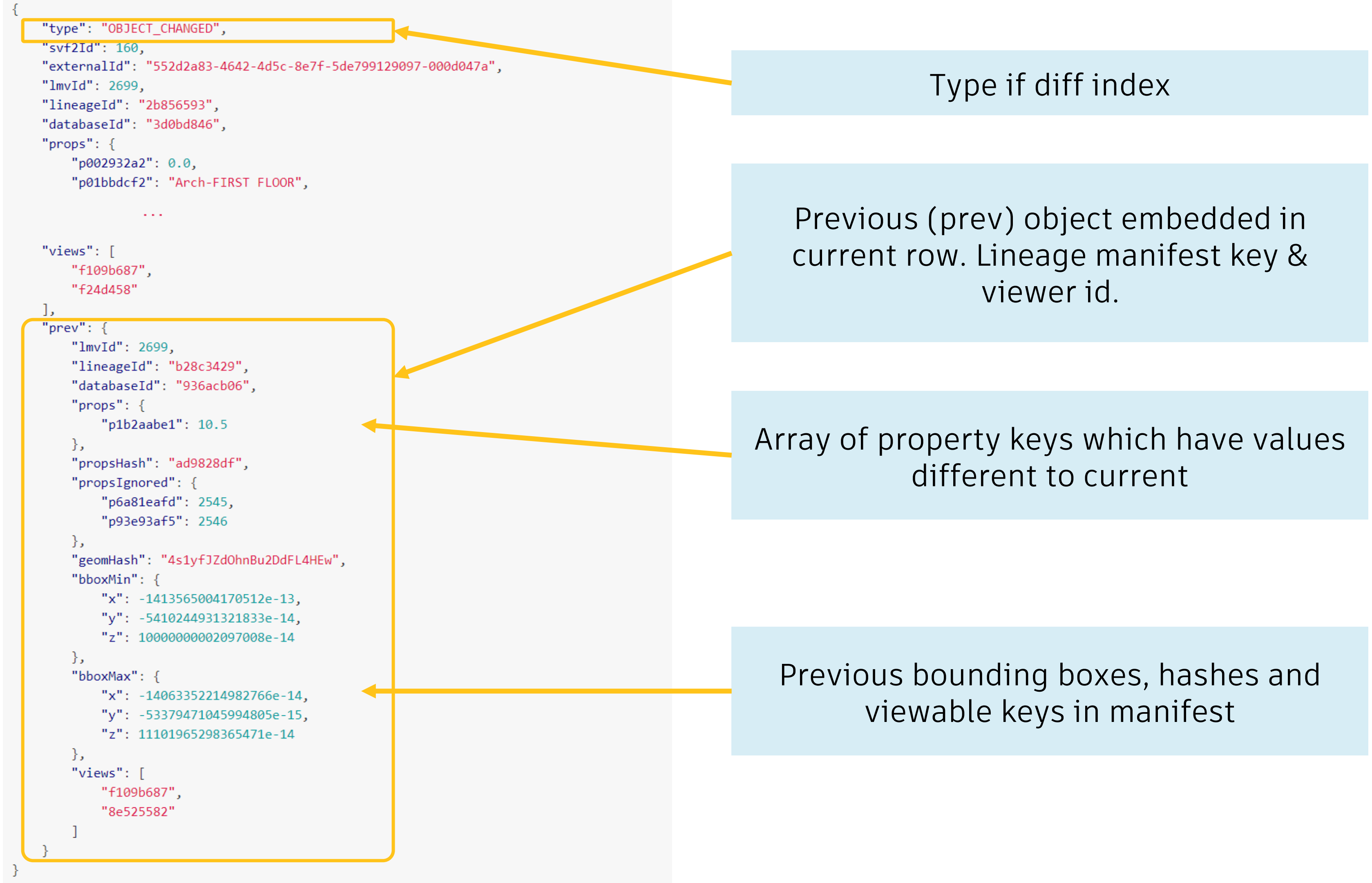
Diff Endpoints

	Endpoints		
Diff	POST	diffs:**batch-status**	
	GET	diffs/**:diffId**	
	GET	diffs/:diffId/**manifest**	⬇
	GET	diffs/:diffId/**fields**	⬇
	GET	diffs/:diffId/**properties**	⬇
Query	POST	diffs/:diffId/**queries**	
	GET	diffs/:diffId/queries/**:queryId**	
	GET	diffs/:diffId/queries/:queryId/**properties**	⬇

- Diff - The steps are the same as Index
- Specify two version urn's to compare:

```
{
  "diffs": [
    {
      "prevVersionUrn": "urn:adsk.w
      "curVersionUrn": "urn:adsk.wi
    }
  ]
 }
```

- stats: add, removed, modified



```
{
    "type": "OBJECT_CHANGED",
    "svf2Id": 160,
    "externalId": "552d2a83-4642-4d5c-8e7f-5de799129097-000d047a",
    "lmvId": 2699,
    "lineageId": "2b856593",
    "databaseId": "3d0bd846",
    "props": {
        "p002932a2": 0.0,
        "p01bbdcf2": "Arch-FIRST FLOOR",

                  ...

    "views": [
        "f109b687",
        "f24d458"
    ],
    "prev": {
        "lmvId": 2699,
        "lineageId": "b28c3429",
        "databaseId": "936acb06",
        "props": {
            "p1b2aabe1": 10.5
        },
        "propsHash": "ad9828df",
        "propsIgnored": {
            "p6a81eafd": 2545,
            "p93e93af5": 2546
        },
        "geomHash": "4s1yfJZdOhnBu2DdFL4HEw",
        "bboxMin": {
            "x": -1413565004170512e-13,
            "y": -5410244931321833e-14,
            "z": 10000000002097008e-14
        },
        "bboxMax": {
            "x": -14063352214982766e-14,
            "y": -5337947104599480 5e-15,
            "z": 11101965298365471e-14
        },
        "views": [
            "f109b687",
            "8e525582"
        ]
    }
}
```

Type if diff index

Previous (prev) object embedded in current row. Lineage manifest key & viewer id.

Array of property keys which have values different to current

Previous bounding boxes, hashes and viewable keys in manifest

Basic Index Row vs. Diff Index Row

Description	Current version	Previous version
IDs	s.svf2Id s.**externalId**	
Change type, previous vs. current		s.**type** s.**changeType**
lineage version info, SVF2 database URNs	s.lmvId s.lineageId s.databaseId	s.**prev**.lmvId s.prev.lineageId s.prev.databaseId
Property values	s.props.* s.propsHash s.propsIgnored.*	s.prev.props.* s.prev.propsHash s.prev.propsIgnored.*
Geometry hash and bounding box values IF viewable	s.geomHash s.bboxMin.x s.bboxMin.y s.bboxMin.z s.bboxMax.x s.bboxMax.y s.bboxMax.z	s.prev.geomHash s.prev.bboxMin.x s.prev.bboxMin.y s.prev.bboxMin.z s.prev.bboxMax.x s.prev.bboxMax.y s.prev.bboxMax.z
Viewable keys IF viewable	s.views s.views[*i*]	s.prev.views s.prev.views[*i*]

JSON Abstract Syntax Tree → S3 Select(AWS)

```
$not
$and
$or
$gt
$lt
$eq
$le
$ge
```

```
$like
$between
$in
$contains
$isnull
$notnull
$add
$sub
$mul
$div
```

```
$cat
$coalesce
$mod
$cast
$nullif
$date_add
$date_diff
$extract
$substring
$to_string
$to_timestamp
```

```
$char_length
$lower
$upper
$count
$sum
$avg
$min
$max
$trim
$utcnow
$case
```

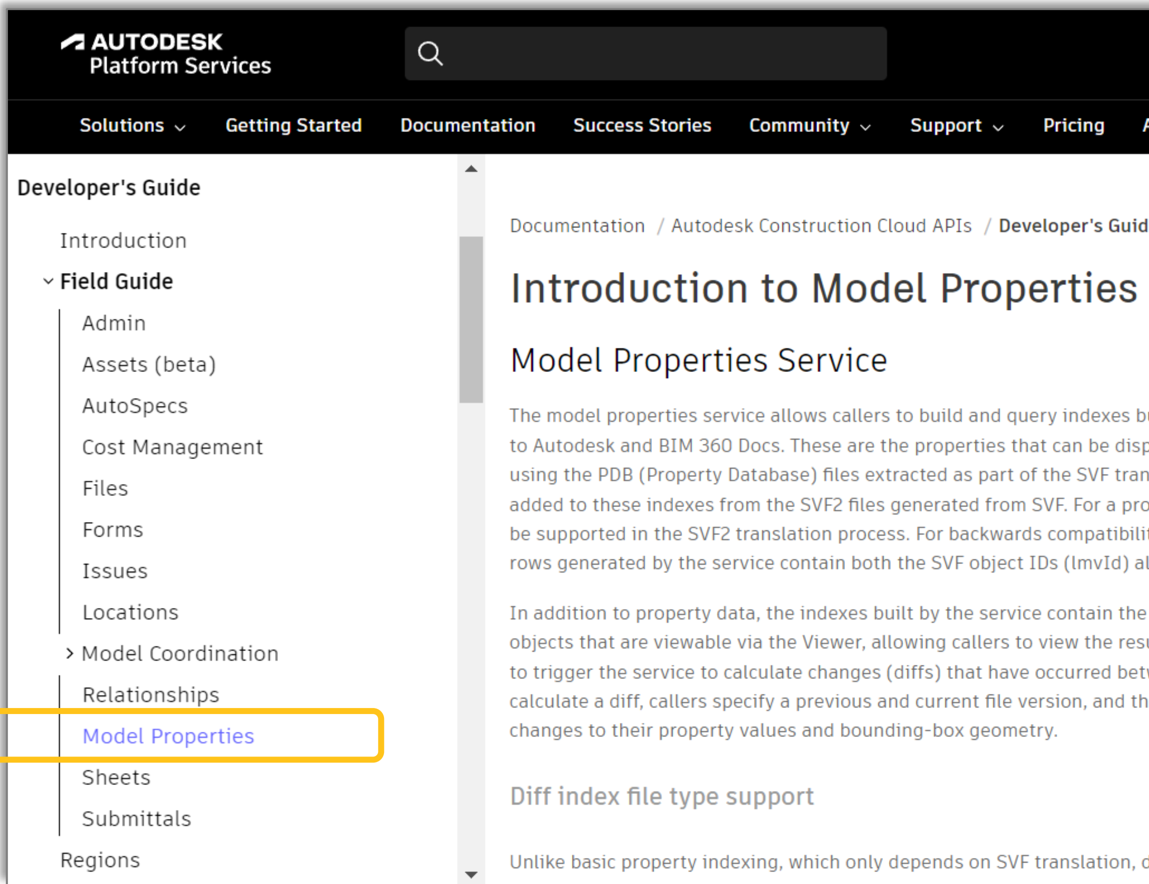
Developer Resources

Model Properties API

Developer Resources

Documentation

- Field Guide
  - Introduction to Model Properties
- Step-by-Step Tutorials
  - Index Querying
  - Tracking Changes
  - Query Language Reference
- Reference Guide
  - Index
  - Diff



Introduction to Model Properties

Model Properties Service

The model properties service allows callers to build and query indexes b
to Autodesk and BIM 360 Docs. These are the properties that can be disp
using the PDB (Property Database) files extracted as part of the SVF tran
added to these indexes from the SVF2 files generated from SVF. For a pro
be supported in the SVF2 translation process. For backwards compatibili
rows generated by the service contain both the SVF object IDs (lmvId) al

In addition to property data, the indexes built by the service contain the
objects that are viewable via the Viewer, allowing callers to view the resu
to trigger the service to calculate changes (diffs) that have occurred bet
calculate a diff, callers specify a previous and current file version, and th
changes to their property values and bounding-box geometry.

Diff index file type support

Unlike basic property indexing, which only depends on SVF translation, c

Developer Resources

- Code Samples on GitHub
  - Postman Collection (correspond to three Step-by-Step tutorials)
  - Model Properties API Walkthrough in PowerShell Core (scripting to explore query language)
  - Element Filtering and Partial Model Load (Integration with Viewer)
  - Compare Two Versions (Integration with Viewer)

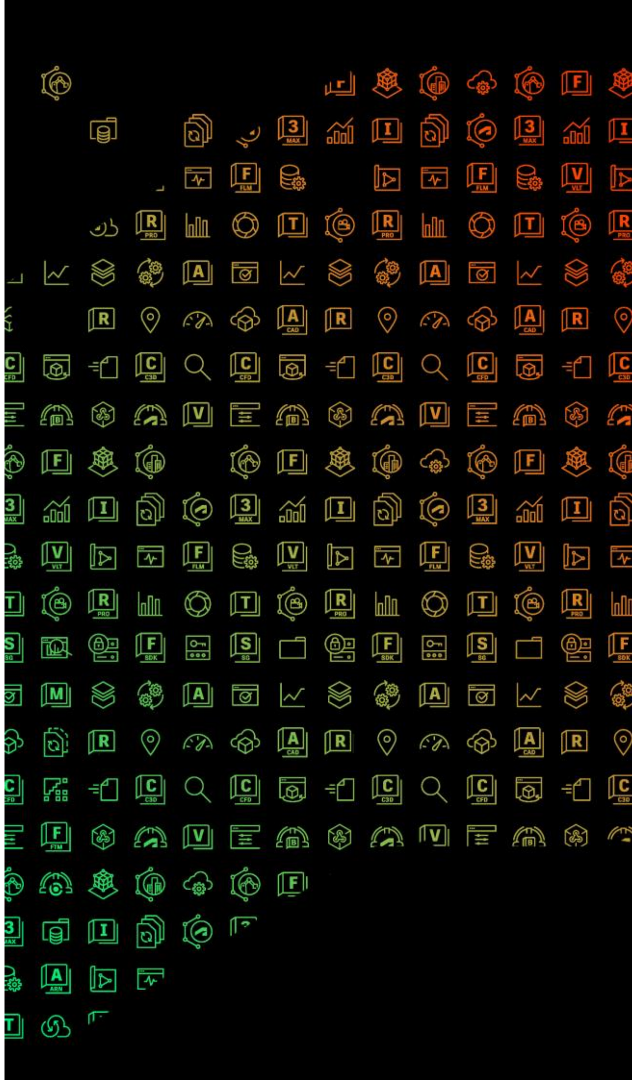
- Blog Post
  - "BIM 360/ACC Model Properties API"
    https://aps.autodesk.com/blog/bim-360acc-model-properties-api
    includes links to the resources
  - Search more with "Model Properties", e.g.,
    https://aps.autodesk.com/blog/model-properties-api-vs-model-derivative-api

Common Questions

Methods to Access Design Data

Model Derivative	Model Properties	Design Automation	Data Exchange	AEC Data Model
Platform	ACC/BIM 360	Platform	Autodesk Docs	Autodesk Docs[*2]
File based	File based	File based/RCM	Cloud hosted	Cloud hosted
Entire model	Entire model	Entire model Revit	Partial model Revit	Entire model Revit
Light weight query	Full query/filter	Control same as desktop add-in	Flexible query (GraphQL)	Flexible query (GraphQL)
Read	Read	Read/write	Read & evolving limited write	Read & longer term evolving limited write
Today	Today	Today	Public beta/Future	Public beta/Future

Summary



AUTODESK

Make Anything

https://aps.autodesk.com/get-help
aps.help@autodesk.com