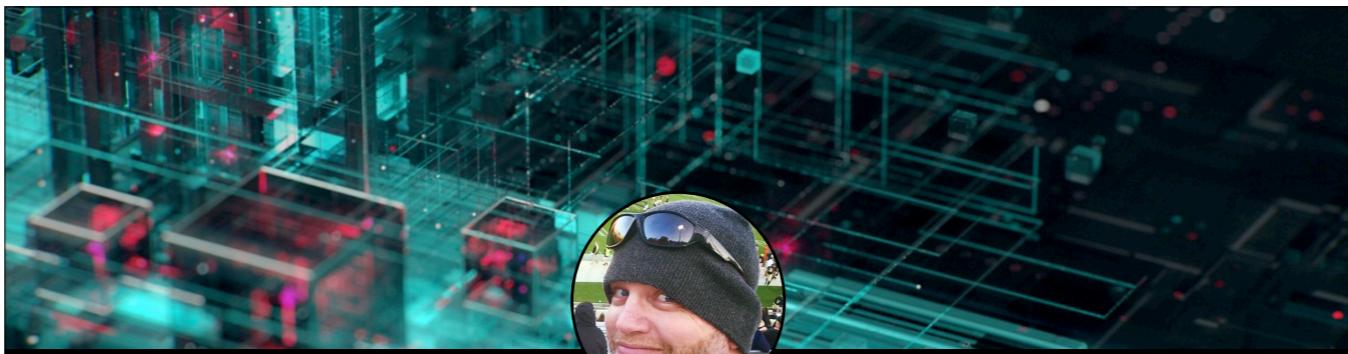




- Show of hands:
 - Have you heard of forge?
 - Have you used forge?
- The purpose of this talk is to just give you a quick intro to Forge
- Heads up that Forge is an Autodesk product
- I am not a salesman and I'm not trying to sell Forge
- This is just a session to make you aware of Forge and the APIs available to developers
- and to help you find get more information if you're interested



JOSH TOMLINSON

Senior Software Engineer, Autodesk

Josh worked at Rhythm & Hues Studios for 11 years where he helped develop pipeline and workflow solutions for feature film visual effects. He then spent a year and a half building an open source pipeline at Clemson University, his alma mater, before joining Autodesk in 2015. Josh is a Software Engineer as part of the Shotgun Toolkit team and enjoys building tools and APIs used by hundreds of production studios around the world.



What will you learn?

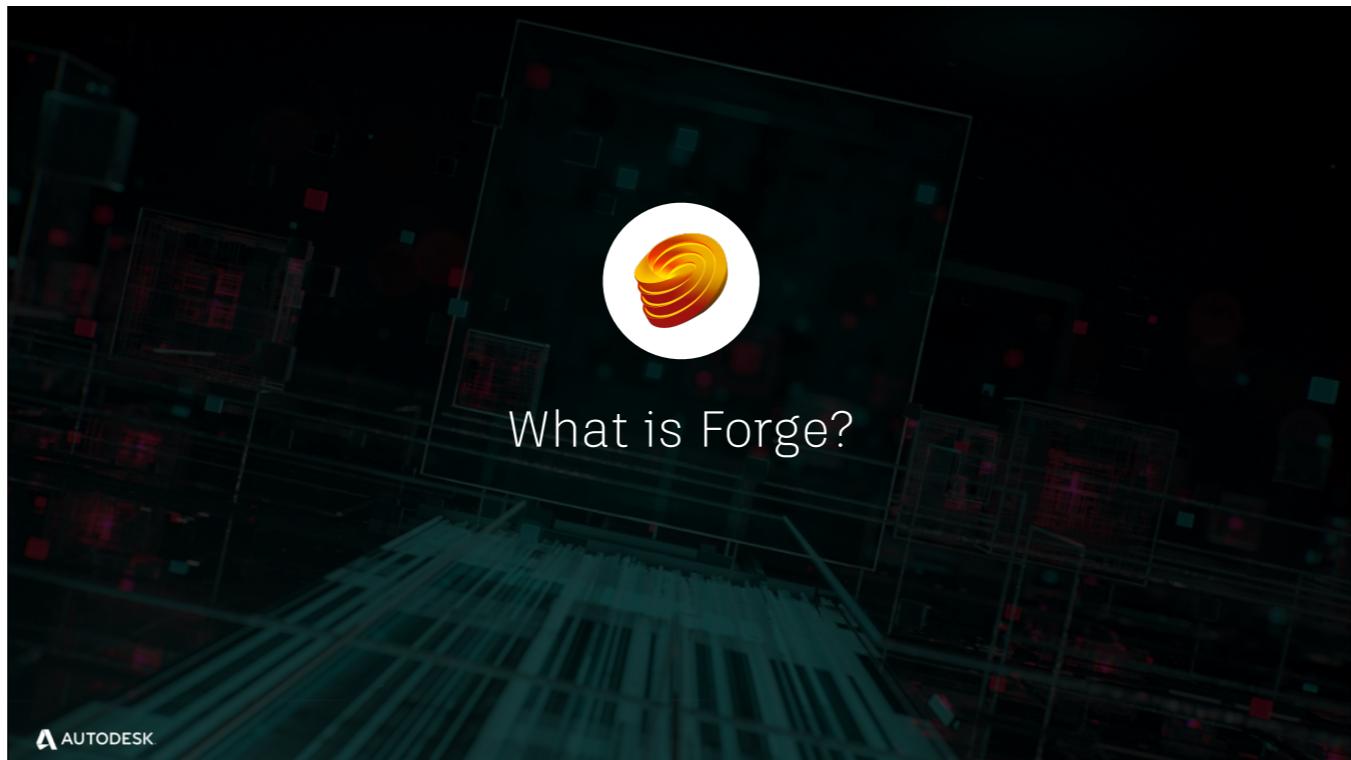
- What is Forge?
- What APIs are available?
- What a Forge + Shotgun integration might look like
- How to find more info about Forge

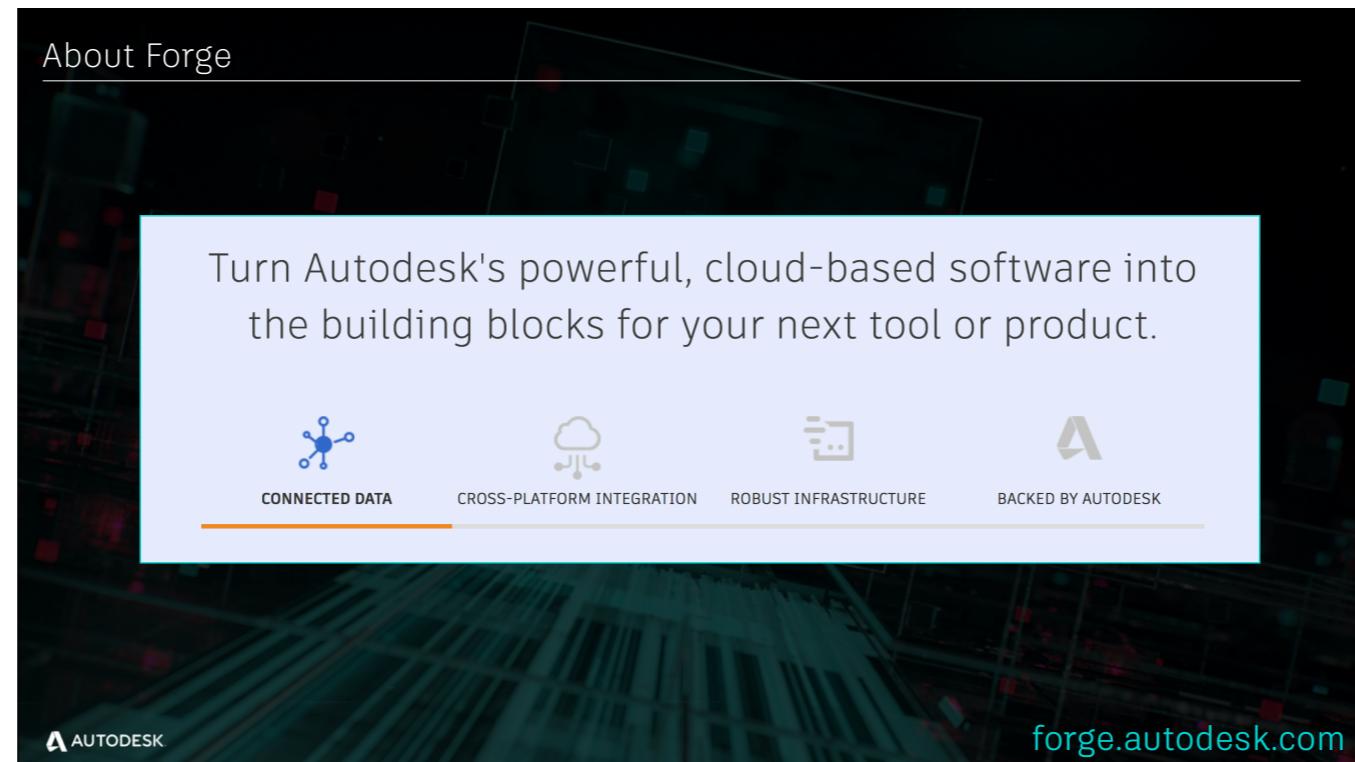


AUTODESK

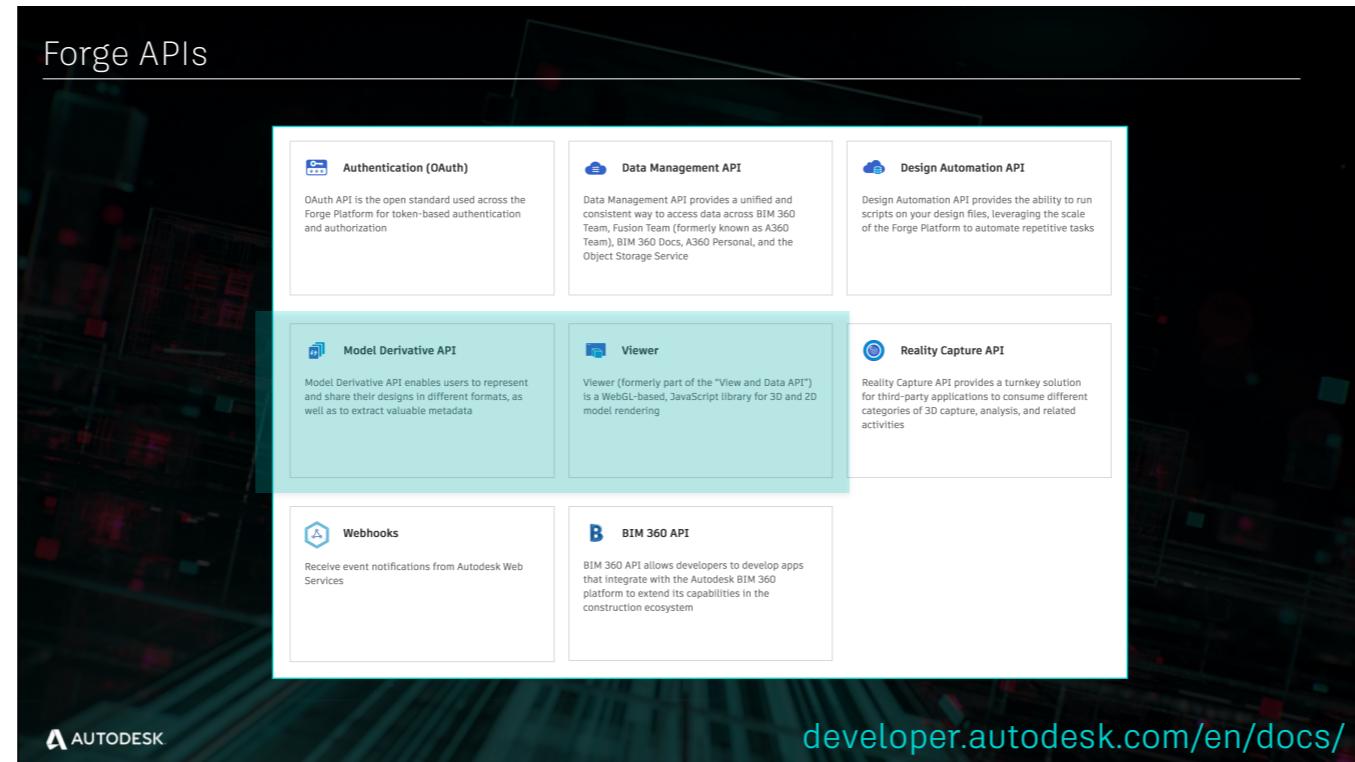
- What will you learn in this session?
- We're going to keep it pretty high level just to give a broad introduction to the platform
- We'll take a quick look at some of the APIs available via Forge
- I have a really basic example of how you might use Forge with SG
- And I'll point you to all the forge documentation so you can begin exploring if you're interested







- Forge is Autodesk's Developer Platform
- This is a screenshot from the forge website
- What it is - is a collection of cloud-based services and APIs
- The official description is this:
 - Forge enables companies to leverage design and engineering data to develop custom software applications and connected workflows for manufacturing, media/entertainment, architecture, engineering, and construction.
- These are the same services used internally at Autodesk
- There's also a Forge Investment Fund
- Autodesk is looking to invest in companies that are using the forge api to build new businesses or extend their existing business to the cloud
- I recommend reading more about that on the forge site as well



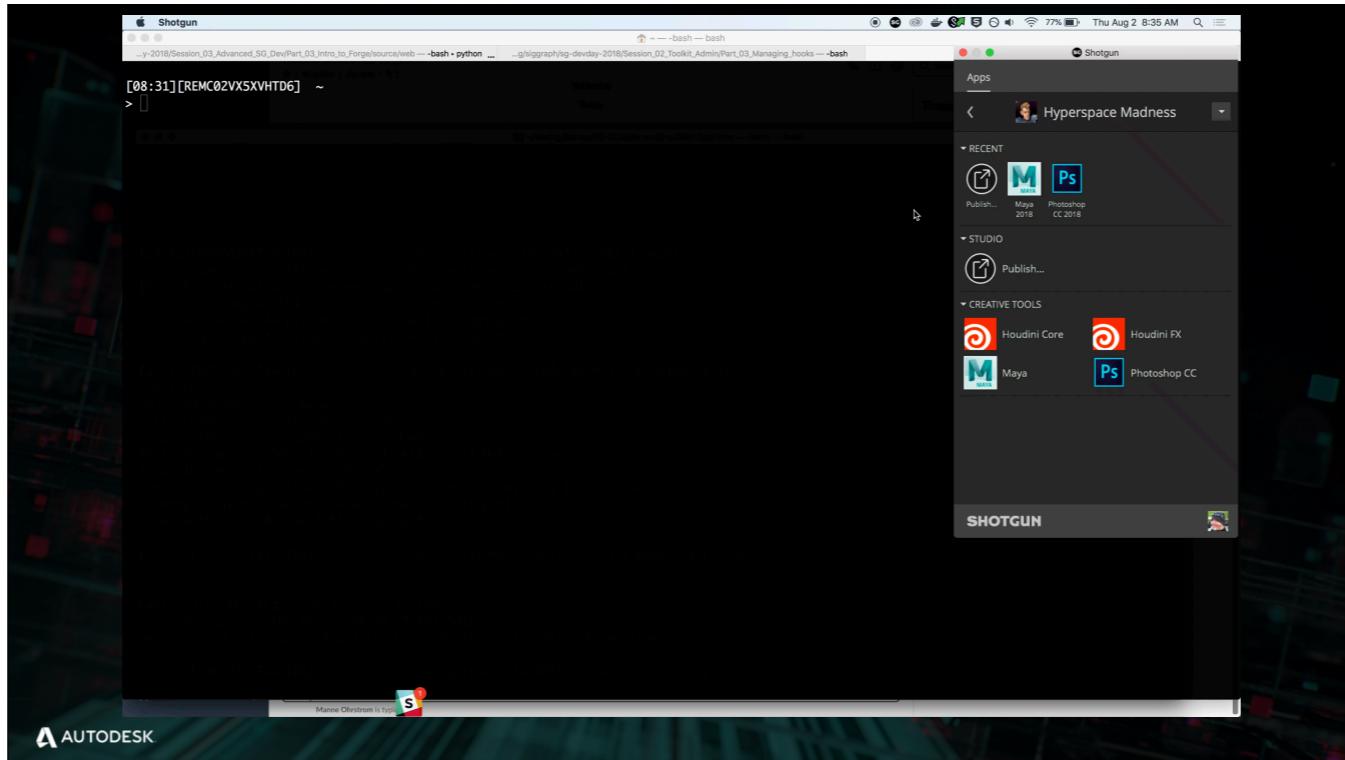
- There are a number of APIs available on the platform
 - OAuth2, token-based authentication for accessing forge services
 - There's a data management API for creating and handling project data, schemas, and object storage
 - A web hooks API for responding to events from the data management service
 - There's also a reality capture API for aerial photogrammetry and creating geometry from hand held camera photos
- The two APIs that I'd like to focus on for this session though are
 - The Model Derivative API - for storing, converting, and sharing geometry and metadata
 - And the Viewer API - which is a web-based 3D viewer



- Rather than go over more info you can read about on the forge website
- I thought it might be more interesting to see a really basic example of what's possible using some of the Forge APIs in conjunction with SG
- Everything in this example exists today, there's no new or unreleased tech involved
- It's just connecting Forge with SG
- All of the code to make this work is available in the dev day resources repo, so you can try it out if you like
- I also highly recommend the Forge step-by-step tutorials on the forge developer site



- So let's start with a little motivation for this demo
- A supervisor wants to review the latest published model for an asset directly in SG
- Hmmmm. Ok.
- Well, sups can review 2D stuff in Shotgun... this isn't a crazy request
- Maybe we can make this work with Forge...
- Forge has a 3D viewer, let's see if we can upload the published model to Forge and then display it in a Forge viewer that is embedded into Shotgun

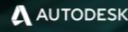


- To begin, I've created a new publish plugin
 - If you're trying to publish an obj file to an asset context
 - This plugin will simply upload the geometry to forge
 - and it will tell the service to convert it to a format that is viewable via the forge model viewer
 - From the user's perspective, it's just another plugin on the left

Forge: Model Derivative API

```
# upload the model
with open(model_path, 'rb') as f:
    result = requests.put(
        FORGE_OBJECT_UPLOAD.format(object_name=object_name),
        headers={
            "Authorization": "Bearer {access_token}".format(
                access_token=forge_token
            ),
            "Content-Type": "application/octet-stream",
        },
        data=f
    )
```

upload_to_forge.py (publish plugin)

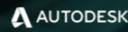


- How does this work under the hood?
- The publish plugin is fairly simple
- Once a forge access token has been acquired,
- The model is uploaded via the Model Derivative REST API

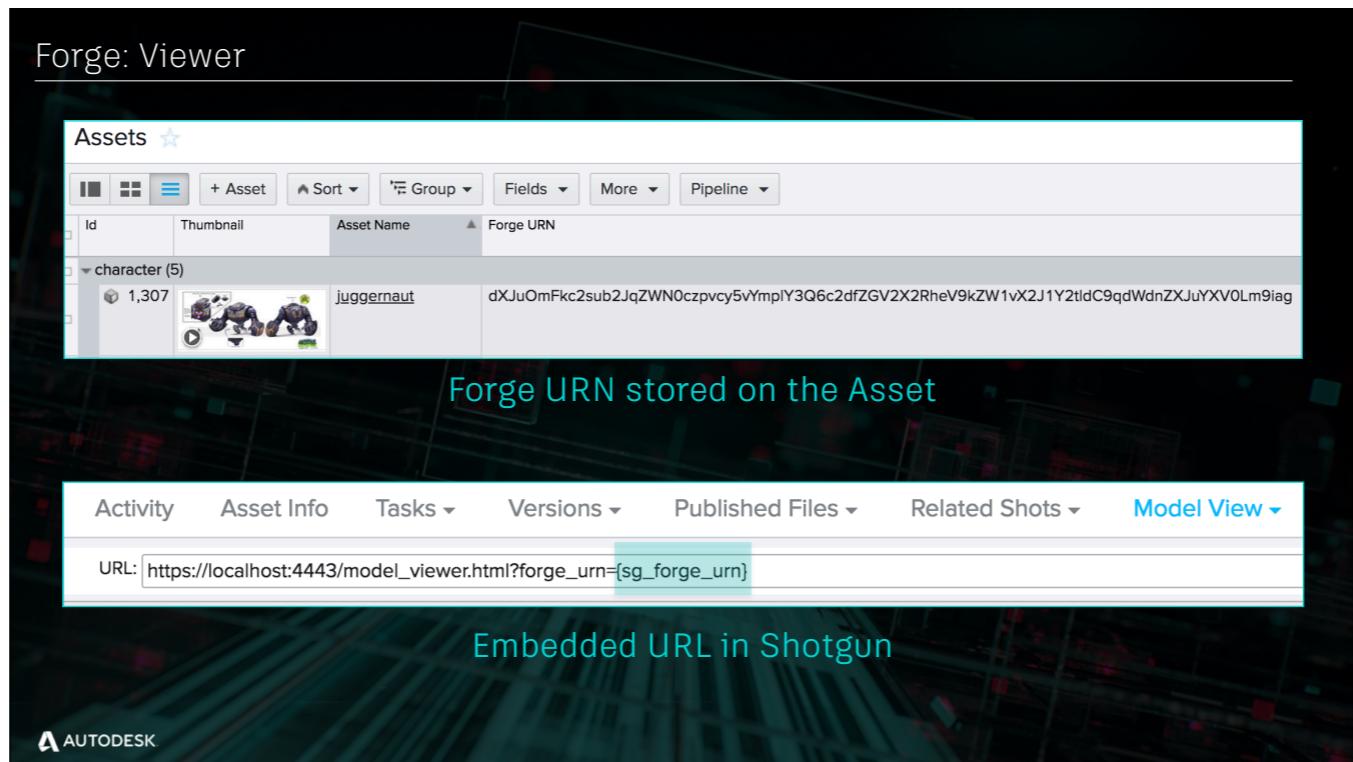
Forge: Model Derivative API

```
# convert to SVF format on the server. this is required for viewing the
# model using the forge viewer API
result = requests.post(
    FORGE_DESIGNDATA_JOB,
    headers={
        "Authorization": "Bearer {access_token}".format(
            access_token=forge_token
        ),
    },
    json={
        "input": {
            "urn": model_urn_base64
        },
        "output": {
            "formats": [
                {
                    "type": "svf",
                    "views": ["2d", "3d"]
                }
            ]
        }
    }
)
```

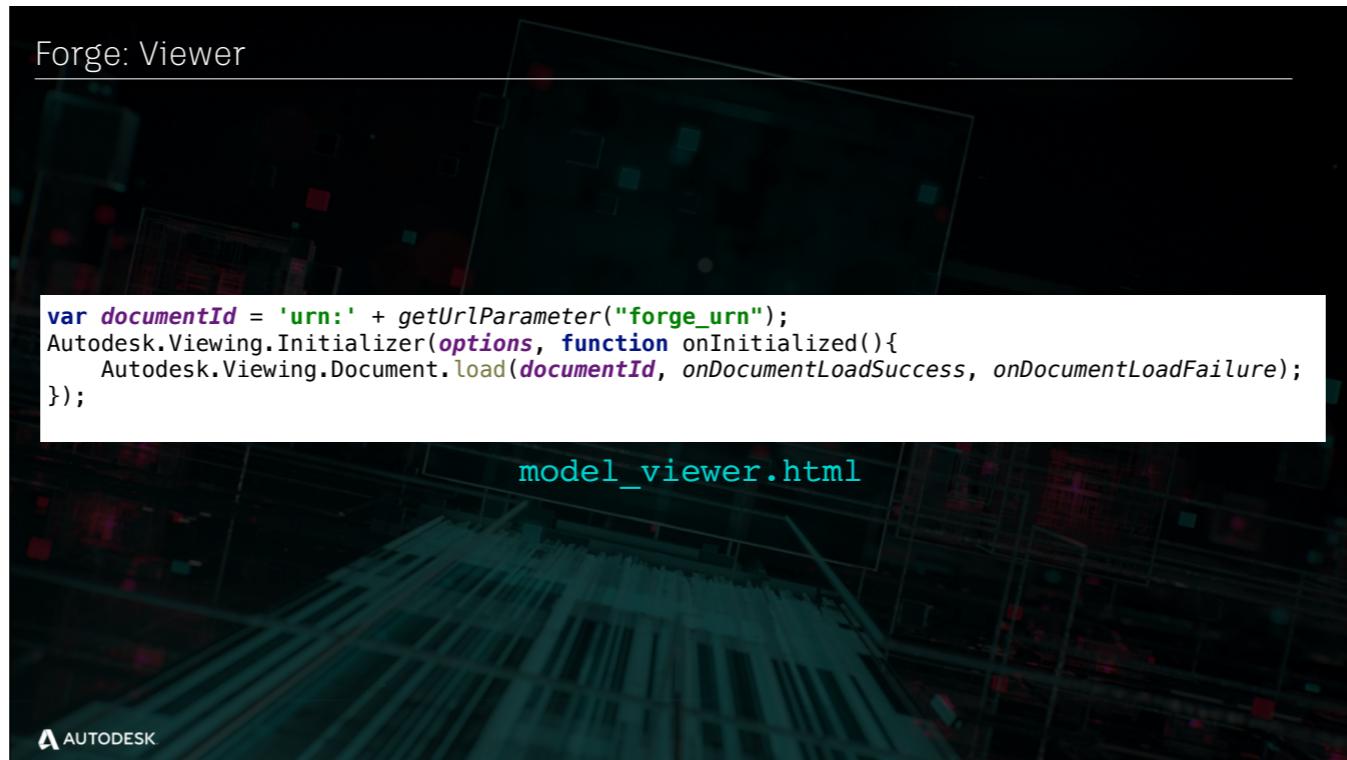
`upload_to_forge.py` (publish plugin)



- Once the model is uploaded, we use the REST API to tell the service to convert the model to a format that can be displayed in the viewer



- On the SG side, all I did was add a new field to the Asset entity to store a Forge URN
- This is the identifier of the model within Forge
- Then, I just created a new tab on assets which is an embedded web page
- The url is to a local server that loads up the forge viewer and pulls the model id for the asset via a field token
- CLICK



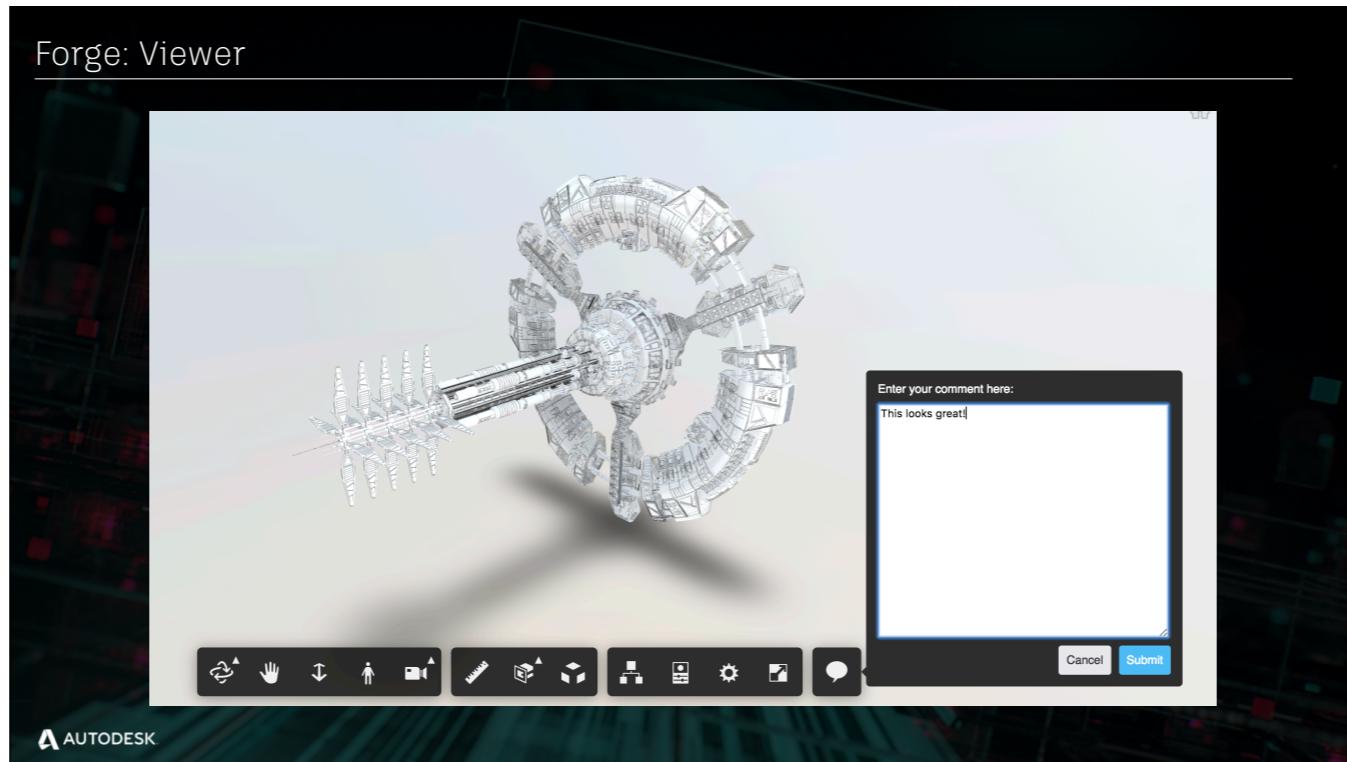
- In terms of the embedded webpage, this is just a little snippet that shows the viewer initialization via the Viewer API
- I'll note that the Viewer tutorials on the forge website are pretty good
- This code comes straight from that tutorial with only the urn parameter parsing being specific to this demo

The screenshot shows the Autodesk Shotgun software interface for the 'HYPERSPACE MADNESS' project. The 'Assets' tab is selected. The main view is a grid of assets with the following columns:

- ID**: Includes asset IDs and a '+' icon for creating new assets.
- Thumbnail**: Preview images for each asset.
- Asset Name**: Names of the assets, such as 'Juggernaut', 'killamari', 'sentinel', 'sven', and 'trilobod'.
- Status**: Current status of the assets.
- Sequences**: Associated sequences for each asset.
- Type**: Asset type, mostly 'character'.
- Art**: Status icons for Art tasks.
- Model**: Status icons for Model tasks.
- Texture**: Status icons for Texture tasks.
- LookDev**: Status icons for LookDev tasks.
- Chk**: Status icons for Chk tasks.
- Ensemble**: Status icons for Ensemble tasks.
- Description**: Detailed descriptions of each asset, including notes from team members like Chris, Belma, Nancy, and Andrew.
- Open Notes**: Number of open notes for each asset.
- Open Notes Count**: Total number of open notes across all assets.
- Task T**: Task tracking section.

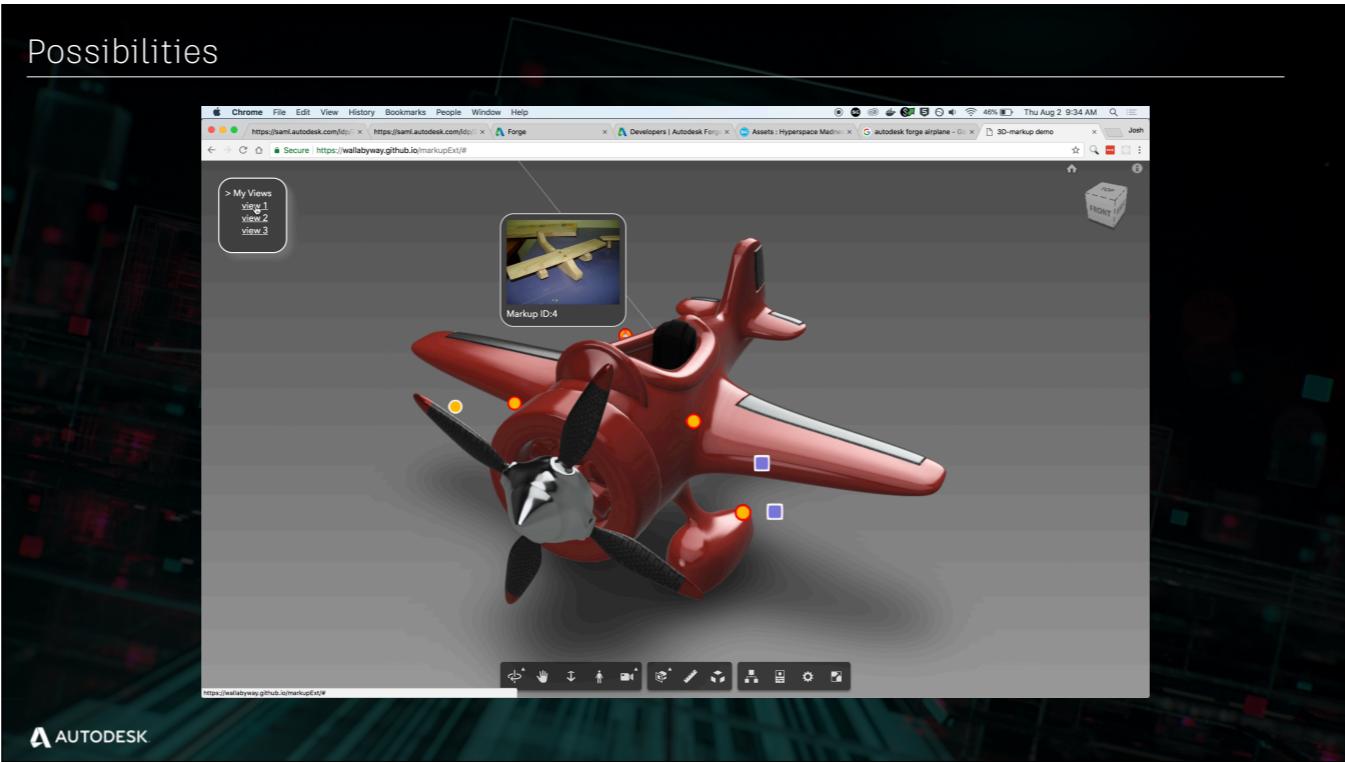
At the bottom left, there's a sidebar with categories: character (9), element (16), environment (5), matte (9), and prop (18). At the bottom right, it says '1 - 53 of 53 Assets' and '100 per page +'. The Autodesk logo is at the bottom center.

- So now if we browse to that asset in SG
- And with the model uploaded to Forge, we can use the model view directly in SG
- You can see the viewer comes with a bunch of tools by default
- Hopefully this gives you an idea of what is possible with the Forge API and Shotgun



- Here's a screenshot from an earlier iteration of this demo
- The viewer has the notion of extensions and here I was able to write an extension that
 - Adds a new button down at the bottom
 - Pops up a text entry box
 - And when you click submit it uses the SG REST API to add a new comment to the asset
- This part isn't included in the dev day resources code, but I wanted to show you that there is a lot of possibility here

Possibilities



- Here's another demo I wanted to show
- someone else wrote this and it's available online (link is included in the dev day resources)
- It shows 3D markup in the Forge Viewer which is pretty cool
- Again, hopefully this is helping you get an idea of some of the basics and what is possible

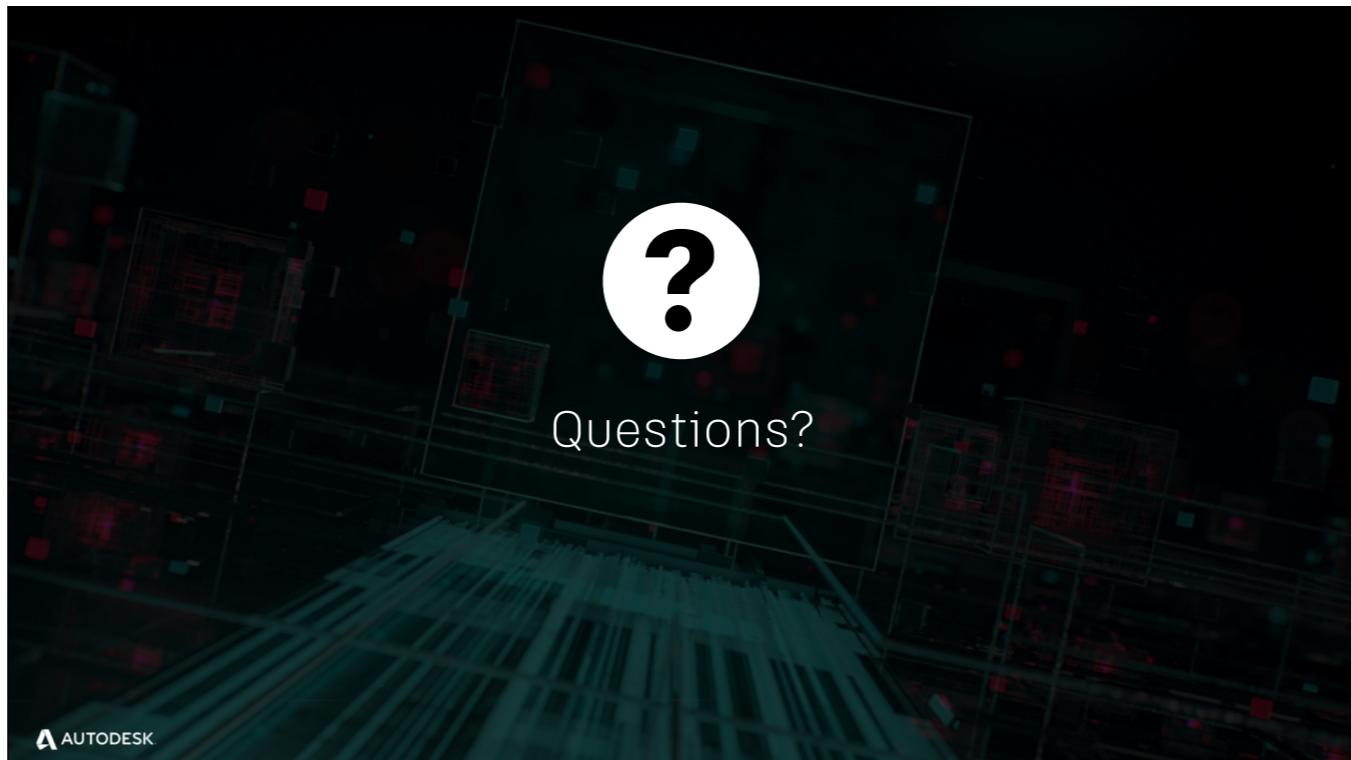
Further Reading...

- Forge Homepage - forge.autodesk.com
- Forge APIs - developer.autodesk.com/en/docs/
- Publisher Dev Docs - developer.shotgunsoftware.com/tk-multi-publish2
- 3D Markup Example - wallabyway.github.io/markupExt



AUTODESK

- Forge Homepage - forge.autodesk.com
- Forge APIs - developer.autodesk.com/en/docs/
- Publisher Developer Docs - developer.shotgunsoftware.com/tk-multi-publish2
- 3D Markup Example - wallabyway.github.io/markupExt



- I'd like to introduce Kevin Vandecar who is a Forge Developer Advocate at Autodesk
- And hopefully we can answer any questions you all might have

SHOTGUN ECOSYSTEM USER GROUP

3:30 - 4:30PM

PRESENTED BY



Rob



OVERVIEW

Join us to hear about our recent developments and our upcoming plans. We also want to hear what your top priorities are, what we should fix, and what features you'd like to see to make Shotgun a more useful platform to build on.