

# SHOTGUN SIGGRAPH 2019



- 9:00 - 9:50AM** Introduction to Shotgun
- 10:00 - 10:50AM** Accelerating Creative Teams with Shotgun Create
- 11:00 - 11:50PM** The Future of Shotgun Platform
- 1:00 - 2:20PM** Developing for Shotgun - An Introduction
- 2:30 - 3:50PM** Developing for Shotgun - Advanced Topics
- 4:00 - 4:50PM** Shotgun Technical Roundtable





# Working with Shotgun Events

Brandon Ashworth





## Brandon Ashworth

Principle Software Engineer, Autodesk

I came to Shotgun from the Visual Effect industry where I worked for 8 years in positions ranging from FX Artist to Lead Software Engineer. I have been with Autodesk for the last 2 years working on projects such as the REST API and Webhooks.

## Agenda

- Events in Shotgun
- What we are going to build
- Event Daemon
  - What is it?
  - Getting started
  - Writing a plugin
- Webhooks
  - What are Webhooks?
  - Porting our plugin to Webhooks
  - Other ways to use Webhooks
- Q&A



# Where to get these materials

---



<https://github.com/shotgunsoftware/sg-siggraph-2019>

# Getting ready to chat

---



<https://app.sli.do/event/10wljptx>

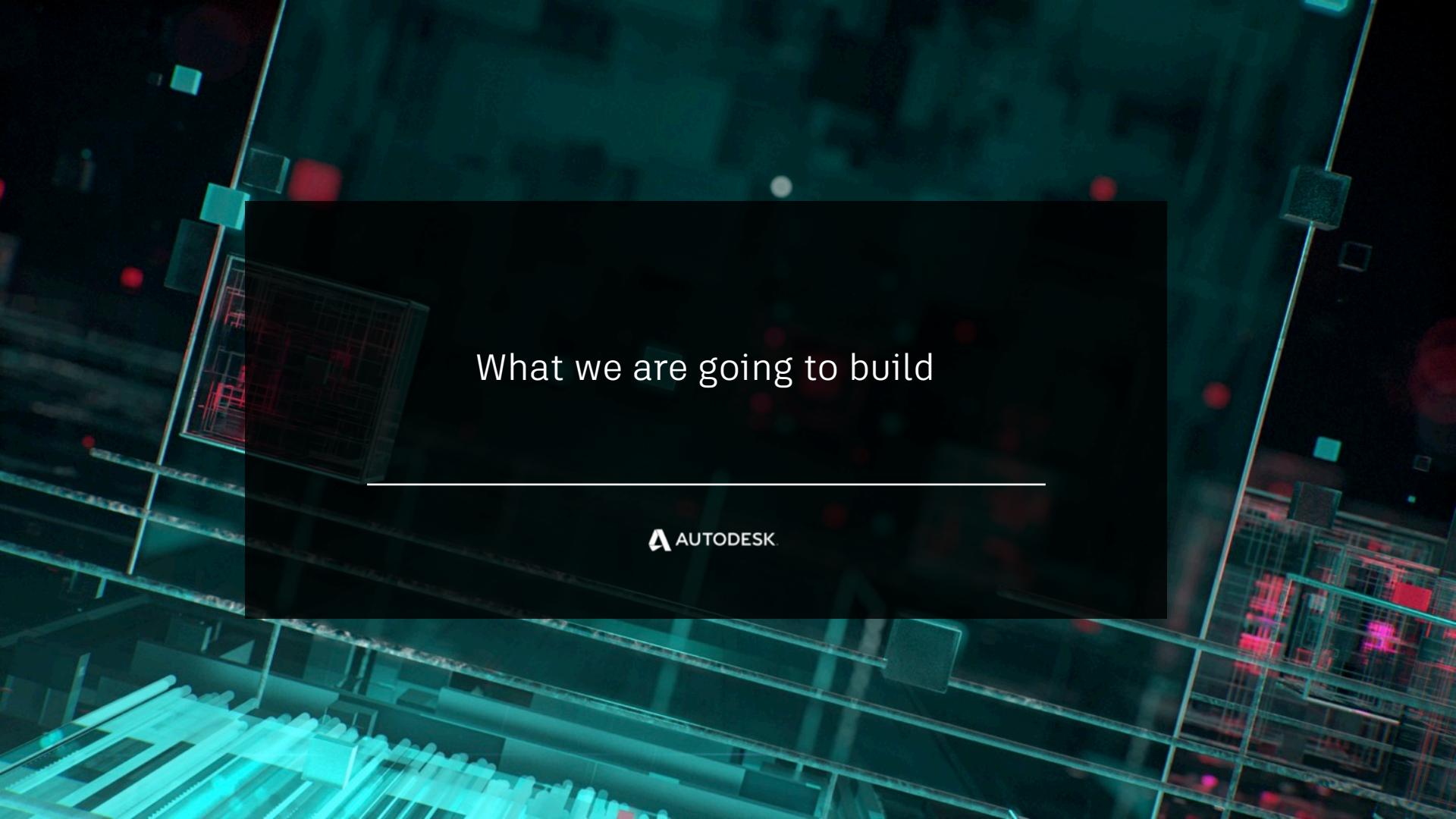
# Shotgun Events

---



# Shotgun Event

```
{  
  "id": 553254,  
  "meta": {  
    "type":  
      "attribute_change",  
    "attribute_name": "code",  
    "field_data_type": "text"  
    "entity_type": "Asset",  
    "entity_id": 2234,  
    "new_value": "Bunny",  
    "old_value":  
      "Big Buck Bunny"  
  },  
  "user": {  
    "id": 91,  
    "type": "HumanUser"  
  },  
  "entity": {  
    "id": 2234,  
    "type": "Asset"  
  },  
  "project": {  
    "id": 331,  
    "type": "Project"  
  },  
  "operation": "update",  
  "created_at":  
    "2019-07-19 21:18:03.108277",  
  "event_type": "Shotgun_Asset_Change",  
  "session_uuid": null,  
  "attribute_name": "code"  
}
```



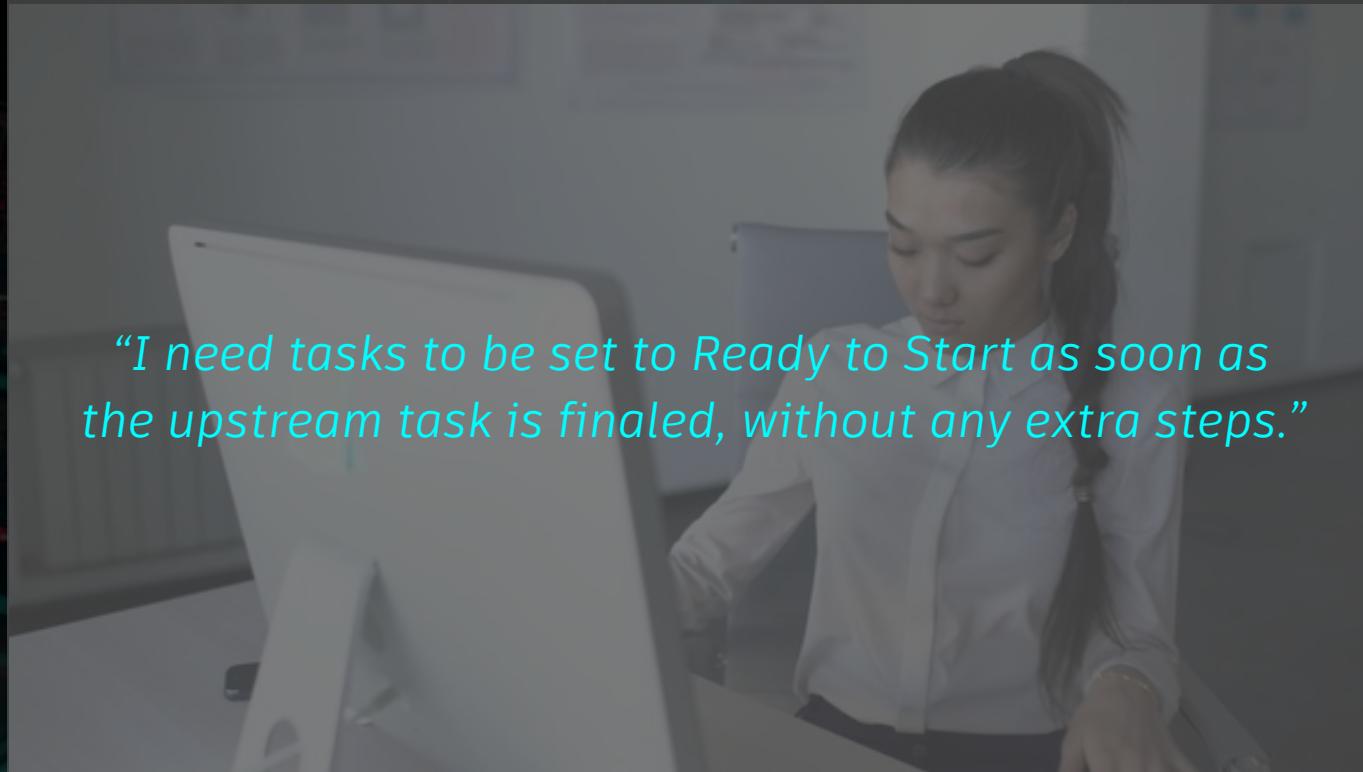
What we are going to build

---



Violet is a layout supervisor.

---



*“I need tasks to be set to Ready to Start as soon as the upstream task is finaled, without any extra steps.”*



## DEV DAY

Overview

Media

Assets

Shots

Tasks

Other ▾

Project Pages ▾

Project Actions ▾

## Tasks

Modified



+ Task

▲ Sort ▾

✖ Group ▾

Fields ▾

More ▾

▼ Search Tasks...



Filter ▾

Task Name	Pipeline Step ▾	Status	Link	Upstream Dependency
▼ TST0100 (5)				
<input checked="" type="checkbox"/> Layout	<span style="background-color: cyan;">■</span> Layout	<span style="color: green;">✓</span>	<span style="color: green;">✓</span> TST0100	<input checked="" type="checkbox"/> Layout
<input checked="" type="checkbox"/> Anim	<span style="background-color: purple;">■</span> Anim	-	<span style="color: green;">✓</span> TST0100	<input checked="" type="checkbox"/> Layout
<input checked="" type="checkbox"/> FX	<span style="background-color: red;">■</span> FX	-	<span style="color: green;">✓</span> TST0100	<input checked="" type="checkbox"/> Anim
<input checked="" type="checkbox"/> Light	<span style="background-color: orange;">■</span> Light	-	<span style="color: green;">✓</span> TST0100	<input checked="" type="checkbox"/> Anim
<input checked="" type="checkbox"/> Comp	<span style="background-color: lightgreen;">■</span> Comp	-	<span style="color: green;">✓</span> TST0100	

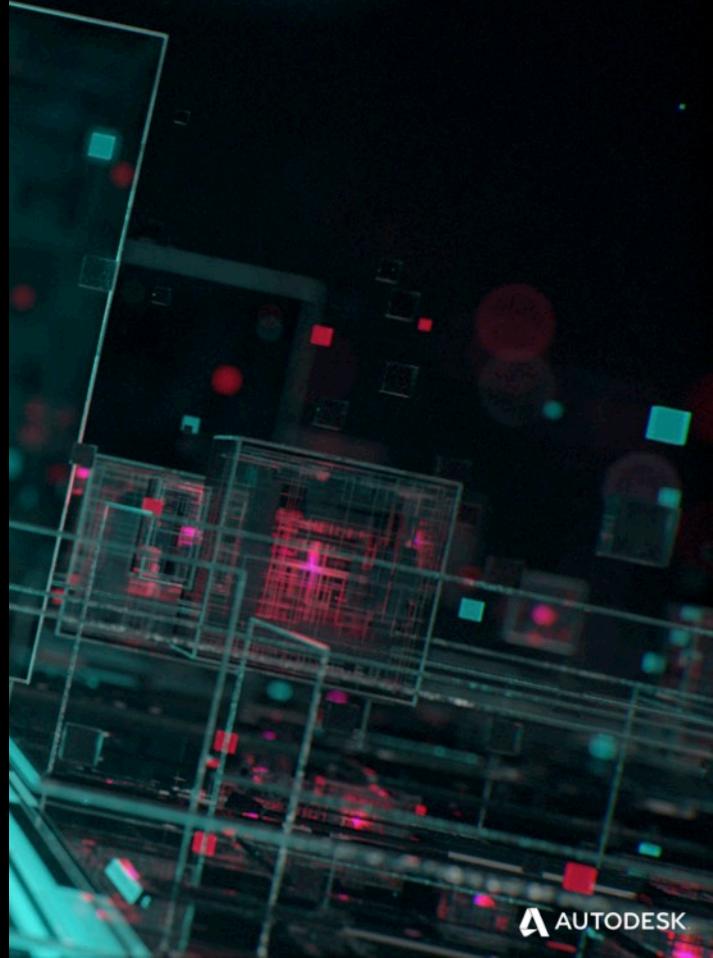
# Shotgun Event Daemon

---



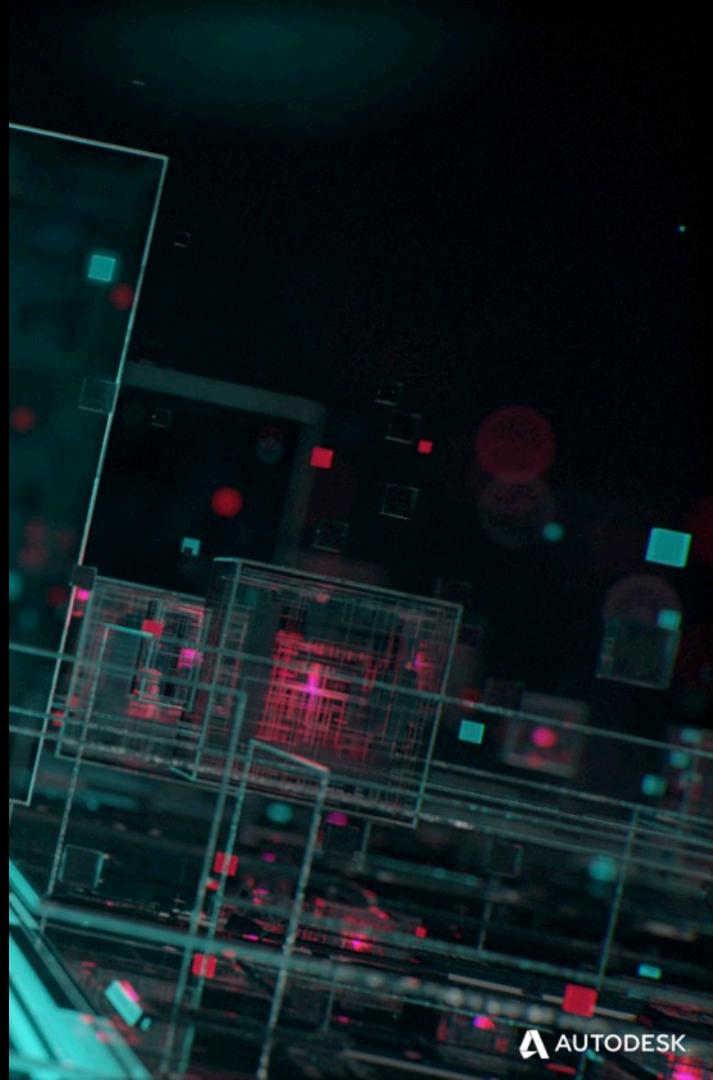
## What is the Event Daemon?

- Open Source
- Python service
- Plugin based
- Polling
- Uses the EventLogEntries entity to find event



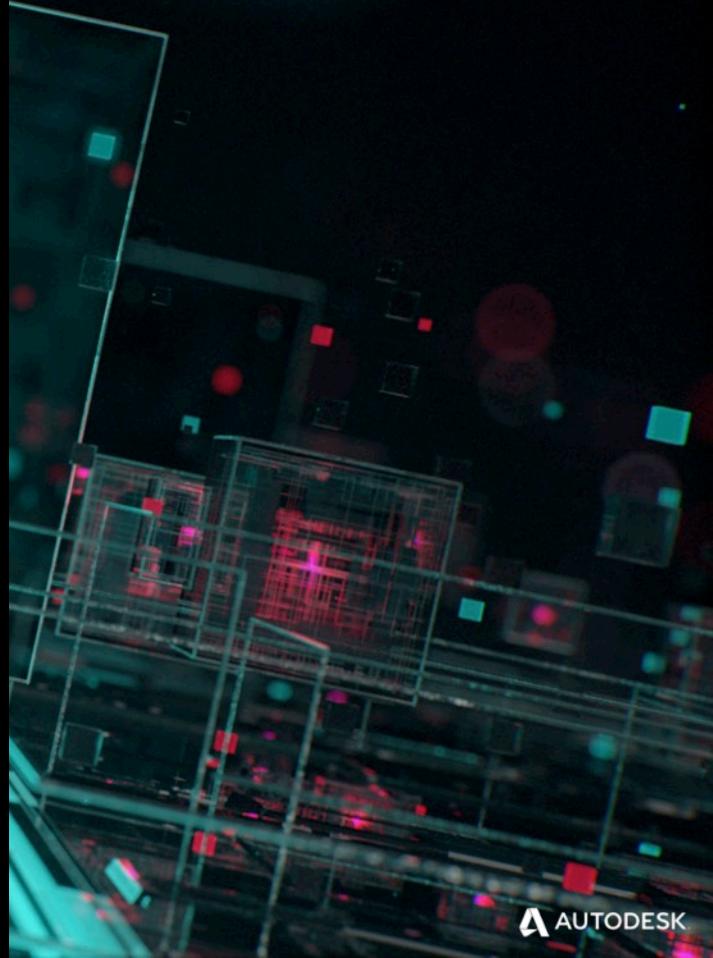
## Getting started

- Clone the repo:
  - <https://github.com/shotgunsoftware/shotgunEvents>
- Configure the daemon
- Develop the plugin



## Project Layout

- deploy/shotgunEventDaemon.conf
  - Event Daemon config file
- deploy/plugins/taskStatusFlipper.py
  - Source code of the status flipper plugin
- src/
  - ShotgunEvents source code.
- requirement.txt
  - Python dependencies
- Dockerfile
  - Dockerfile to describe our container image
- docker-compose.yml
  - Settings for running our container



# deploy/shotgunEventDaemon.conf

## [daemon]

pidFile: /var/run/shotgunEventDaemon.pid

eventIdFile: /shotgun\_events/logs/shotgunEventDaemon.id

logPath: /shotgun\_events/logs

## [shotgun]

server: http://<your site>.shotgunstudio.com

name: event\_daemon

key: p!glctxpqlofalsntcPh5vnuy

## [plugins]

paths: /shotgun\_events/plugins



So you want to write a plugin

---



## deploy/plugins/taskStatusFlipper.py - Part 1

```
import os
import logging

def registerCallbacks(reg):
    eventFilter = {'Shotgun_Task_Change': ['sg_status_list']}

    reg.registerCallback(
        os.environ["SGDAEMON_TASKSTATUS_NAME"],
        os.environ["SGDAEMON_TASKSTATUS_KEY"],
        taskStatusFlipper,
        eventFilter,
        None,
    )

    reg.logger.setLevel(logging.DEBUG)
```

## deploy/plugins/taskStatusFlipper.py - Part 2

```
def taskStatusFlipper(sg, logger, event, args):
    if event['meta']['new_value'] == 'fin':
        downstream_tasks = sg.find(
            'Task',
            [['upstream_tasks', 'is', event['entity']]],
            ['sg_status_list']
        )

        batch_updates = [
            {
                "request_type": "update",
                "entity_type": "Task",
                "entity_id": task['id'],
                "data": { "sg_status_list": 'rdy' }
            } for task in downstream_tasks
            if task['sg_status_list'] == 'wtg'
        ]
        sg.batch(batch_updates)
```

## Tasks ☆

Modified



■ ■ ■ + Task Sort Group Fields More Search Tasks... Filter

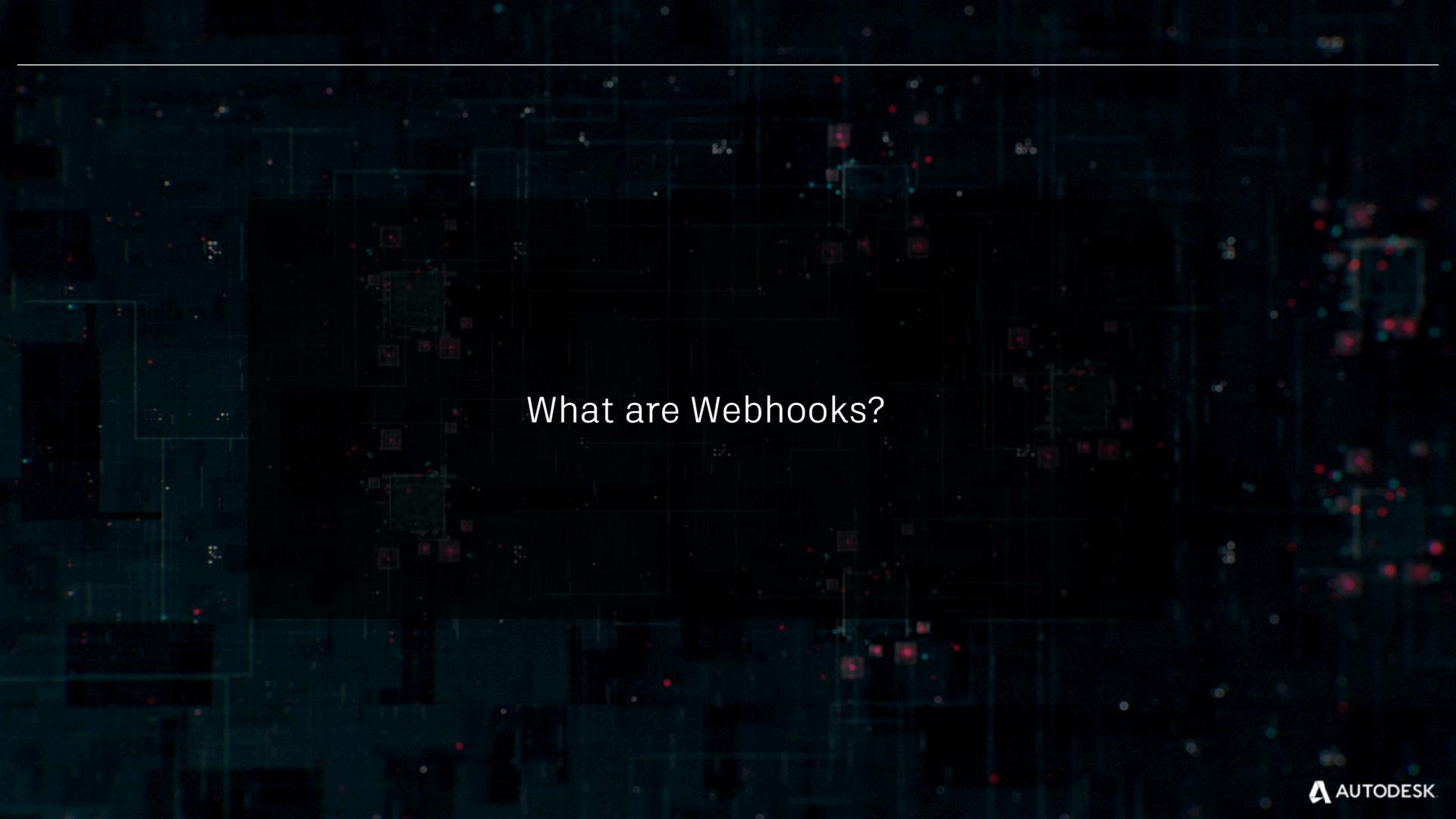
Task Name	Pipeline Step ^	Status	Link	Upstream Dependency	+
▼ TST0100 (5)					
✓ Layout	Layout		TST0100	✓ Layout	
✓ Anim	Anim	-	TST0100	✓ Anim	
✓ FX	FX	-	TST0100	✓ Anim	
✓ Light	Light	-	TST0100	✓ Anim	
✓ Comp	Comp	-	TST0100		

working\_with\_events + ➔ \$  
dc up event\_daemon

# Webhooks



AUTODESK

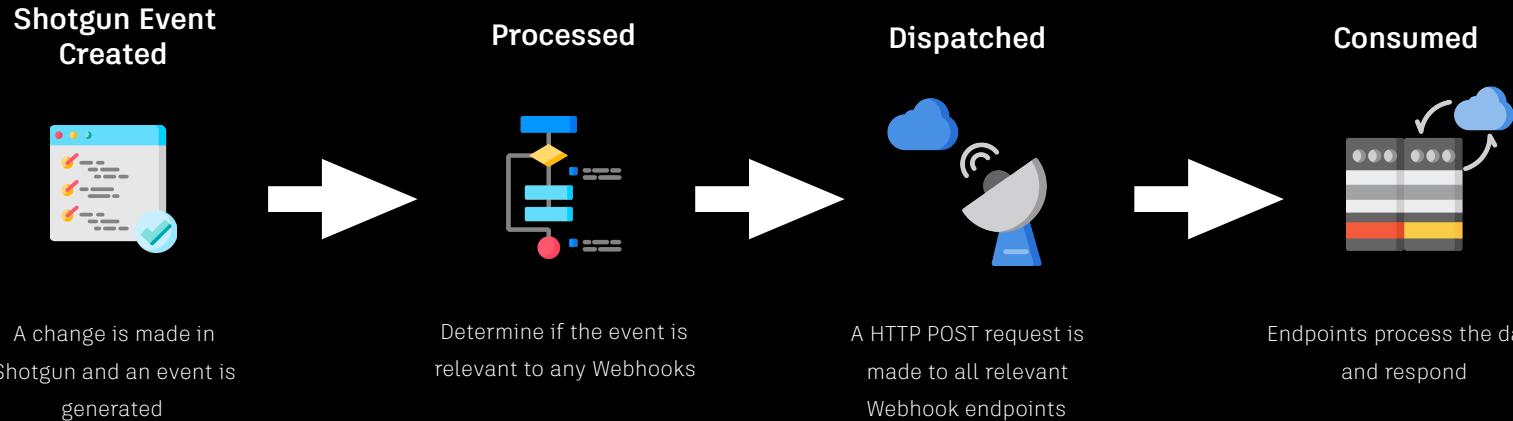


# What are Webhooks?

“... an architectural pattern for web developers to provide user-defined callbacks over HTTP in their applications.”

- Jeff Lindsay

## Webhooks Flow



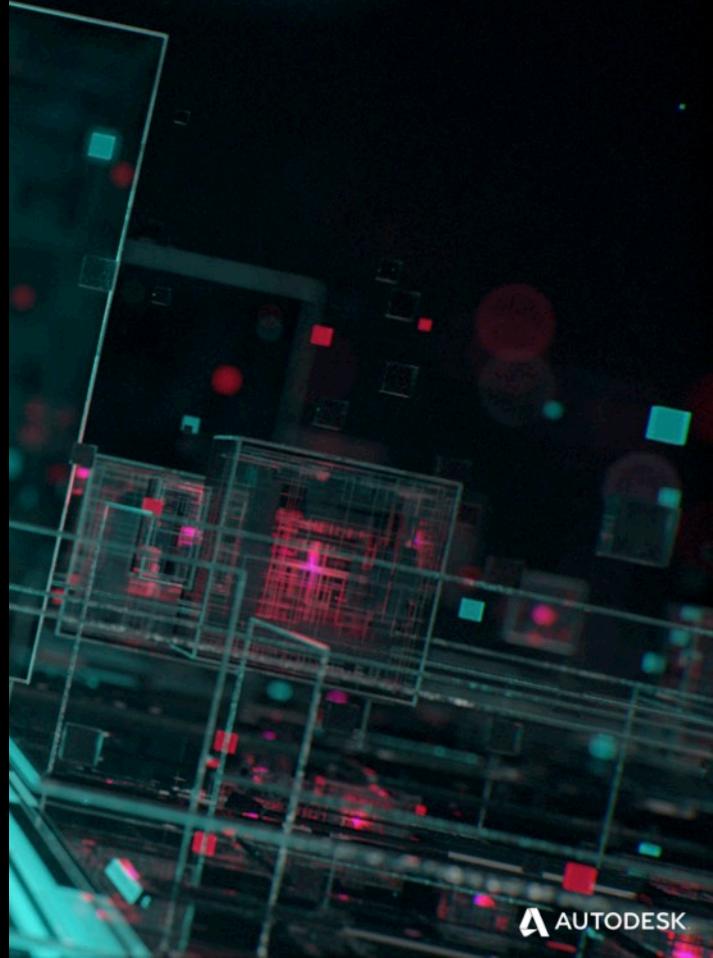
## Shotgun Webhooks Terminology

- Event - The information stored about an action in Shotgun
- Delivery - The event data sent to the Webhook



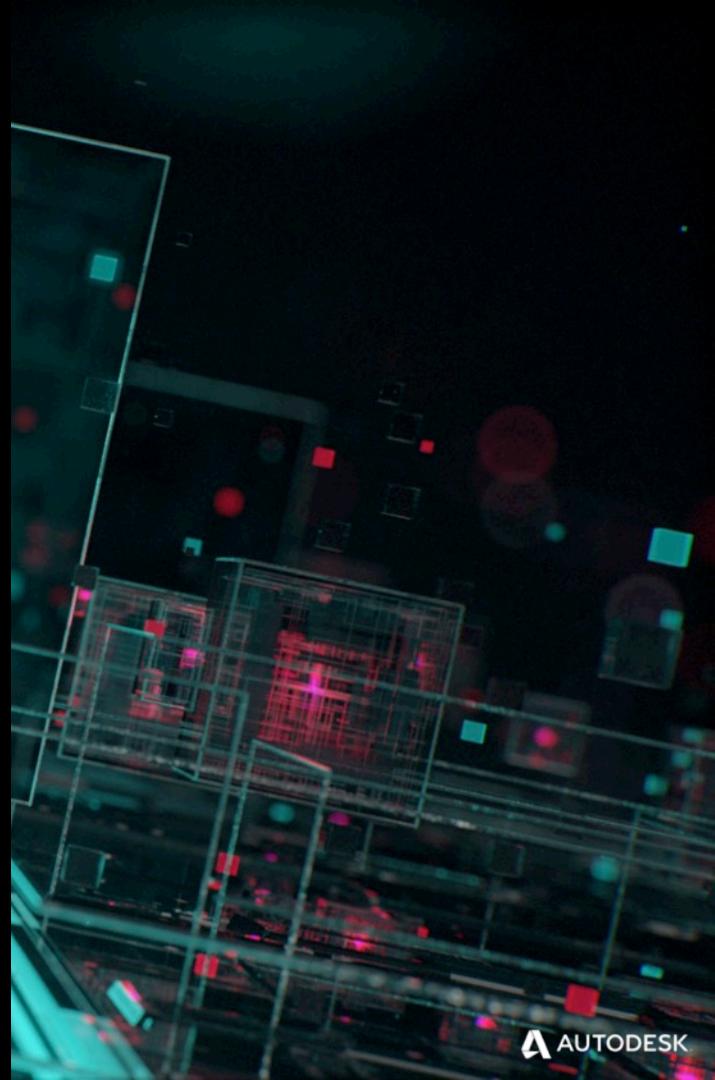
## Webhooks Features

- Subscriptions to:
  - Creates
  - Deletes
  - Updates
  - Can be per Project
- Redeliveries
- Acknowledgments
- Signed requests
- SSL Validation
- REST API



## Things to think about when writing a Webhook

- Errors and Webhook deactivation
  - If a Webhook fails 10 or more time in 24 hour it will be deactivated.
    - Failures: Timeouts, Network failures, Status codes  
 $\geq 400$
- Response time
  - All requests have **6 seconds** to respond before the delivery will be marked as failed.
- Acknowledgments
  - For webhooks that take longer than the timeout.
  - Respond that the requests was successfully received
  - Update the Delivery's 'acknowledgment' field with the final status.



## Webhooks delivery data structure

```
{  
  "data": {  
    "id": "313.0",  
    "event_log_entry_id": 496158,  
    "meta": {...},  
    "user": {...},  
    "entity": {...},  
    "project": {...},  
    "operation": "update",  
    "created_at": "2019-07-25 05:39:16.813053",  
    "event_type": "Shotgun_Shot_Change",  
    "session_uuid": "86c815c2-ae9e-11e9-a964-0242ac140007",  
    "attribute_name": "sg_status_list"  
  },  
  "timestamp": "2019-07-25 05:40:16.813053"  
}
```

## Porting our plugin to Webhooks

- Python
  - Flask
  - Shotgun API



## Project Layout

- app.py
  - Python code for the Flask app
- requirement.txt
  - Python dependencies
- Dockerfile
  - Dockerfile to describe our container image
- docker-compose.yml
  - Settings for running our container



## app.py - Part 1

```
import os
import logging

from flask import Flask, request, make_response
from shotgun_api3 import Shotgun

app = Flask(__name__)
sg = Shotgun(
    os.environ.get('SG_HOST'),
    os.environ.get('SG_SCRIPT_NAME'),
    os.environ.get('SG_SCRIPT_KEY')
)
```

## app.py - Part 2

```
@app.route('/task_status', methods=['POST'])
def task_status():
    try:
        data = request.get_json()
        event = data['data']
        sg.set_session_uuid(event['session_uuid'])
        process_task_status_event(event)
    except Exception, e:
        return make_response(e.message, 500, {})
    return "Ok" # status 200
```

## app.py - Part 3

```
def process_task_status_event(event):
    if event['meta']['new_value'] == 'fin':
        downstream_tasks = sg.find(
            'Task',
            [['upstream_tasks', 'is', event['entity']]],
            ['sg_status_list']
        )
        batch_updates = [
            {
                "request_type": "update",
                "entity_type": "Task",
                "entity_id": task['id'],
                "data": {
                    "sg_status_list": 'rdy'
                }
            } for task in downstream_tasks
            if task['sg_status_list'] == 'wtg'
        ]
        sg.batch(batch_updates)
```

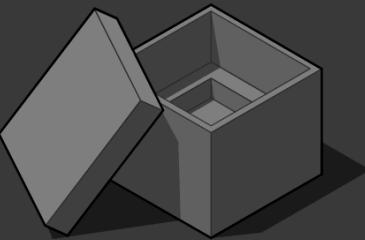
# Creating a Webhook

---



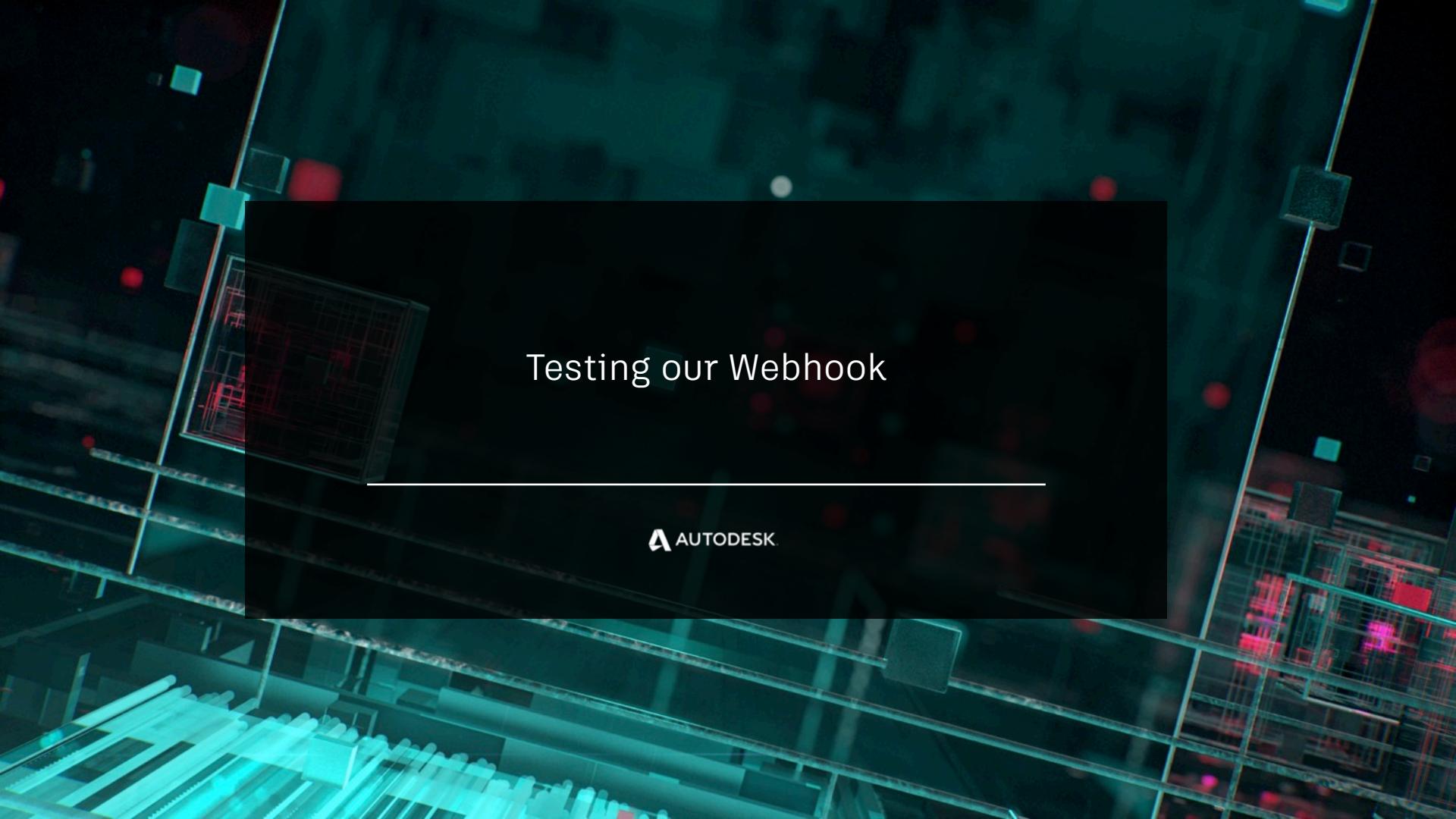
[+ Create Webhook](#)

## Welcome to Webhooks!



**There is no webhook selected.**

Select or create a webhook to get started!



# Testing our Webhook

---



Tasks ★

Modified

+ Task ▲ Sort ▾ Group ▾ Fields ▾ More ▾  Filter ▾

Task Name	Pipeline Step	Status	Link	Upstream Dependency
▼ TST0100 (5)				
✓ Layout	Layout	-	TST0100 -	
✓ Anim	Anim	-	TST0100 -	✓ Layout
✓ FX	FX	-	TST0100 -	✓ Anim
✓ Light	Light	-	TST0100 -	✓ Anim
✓ Comp	Comp	-	TST0100 -	

```
working_with_events ➔ $ ➔  
dc up webhook
```

# Viewing the Webhook request

+ Create Webhook 

Task Status Flipper  5

Webhook URL: [http://e265224a.ngrok.io/task\\_status](http://e265224a.ngrok.io/task_status)

Projects: [Dev Day](#)

Triggered By: Task update

Validate SSL Certificate: true

Secret Token: \*\*\*\*\*

**Deliveries**

Status... | Older than... | Entity Name... | Clear Filters 

> <input type="checkbox"/> sg_status_list changed on Task FX by user task_flipper 1.0	7/26/2019, 2:40:24 PM
> <input type="checkbox"/> sg_status_list changed on Task Light by user task_flipper 1.0	7/26/2019, 2:40:24 PM
> <input type="checkbox"/> sg_status_list changed on Task Anim by user Brandon Ashworth	7/26/2019, 2:40:21 PM
> <input type="checkbox"/> sg_status_list changed on Task Anim by user task_flipper 1.0	7/26/2019, 2:40:17 PM
> <input type="checkbox"/> sg_status_list changed on Task Layout by user Brandon Ashworth	7/26/2019, 2:40:15 PM

Note: Delivery logs are automatically removed from the system after five days.

# Viewing the Webhook request

[+ Create Webhook](#) [Edit](#) [Disable](#) [Delete Webhook](#)

**Task Status Flipper** **ACTIVE**

Webhook URL: [http://e265224.ngrok.io/task\\_status](http://e265224.ngrok.io/task_status)

Projects: **Dev Day**

Triggered By: Task update

Validate SSL Certificate: true

Secret Token: \*\*\*\*\*

**Deliveries**

Status... | Older than... | Entity Name... | Clear Filters [Redeliver Selected](#)

sg\_status\_list changed on Task Anim by user task\_flipper 1.0 // 20/2019, 2:40:17 PM

sg\_status\_list changed on Task Layout by user Brandon Ashworth 7/26/2019, 2:40:15 PM

[Request](#) [Response](#)

**Headers**

```
{ "accept": "application/json", "content-type": "application/json; charset=utf-8", "x-sg-signature": "sha1=8d9715e038c086791754c8e785d1df0cd36ca2", "x-sg-webhook-id": "bcc69ec00-c4c1-48f4-95ae-3819b3ddd9ca", "x-sg-event-batch-id": "49594228971536536138491051139085982641178277584371187810", "x-sg-event-batch-size": "1", "x-sg-webhook-site-url": "https://brandon-webhooks-dev.shotgunstudio.com/", "x-sg-event-batch-index": "0" }
```

**Payload**

```
{ "data": { "id": "317.417.0", "meta": { "type": "attribute_change", "entity_id": 2124, "new_value": "fin", "old_value": "ip", "entity_type": "Task", "attribute_name": "sg_status_list", "field_data_type": "status_list" }, "user": { } } }
```

# Viewing the Webhook request

Request |  Response

## Headers

```
{  
  "accept": "application/json",  
  "content-type": "application/json; charset=utf-8",  
  "x-sg-signature": "sha1=8d9715e038c086791754c8e785d1dfd0cdd36ca2",  
  "x-sg-webhook-id": "bc69ec00-c4c1-48f4-95ae-3819bb3dd9ca",  
  "x-sg-event-batch-id": "49594228971536536138491051139085982641178277584371187810",  
  "x-sg-event-batch-size": "1",  
  "x-sg-webhook-site-url": "https://brandon-webhooks-dev.shotgunstudio.com/",  
  "x-sg-event-batch-index": "0"  
}
```

## Payload

```
{  
  "data": {  
    "id": "317.417.0",  
    "meta": {  
      "type": "attribute_change",  
      "entity_id": 2124,  
      "new_value": "fin",  
      "old_value": "ip",  
      "entity_type": "Task",  
      "attribute_name": "sg_status_list",  
      "status": "Success"  
    }  
  }  
}
```

# Viewing the Webhook request

## Payload

```
{  
  "data": {  
    "id": "317.417.0",  
    "meta": {  
      "type": "attribute_change",  
      "entity_id": 2124,  
      "new_value": "fin",  
      "old_value": "ip",  
      "entity_type": "Task",  
      "attribute_name": "sg_status_list",  
      "field_data_type": "status_list"  
    },  
    "user": {  
      "id": 155,  
      "type": "HumanUser"  
    },  
    "entity": {  
      "id": 2124,  
      "type": "Task"  
    },  
    "project": {  
      "id": 69,  
      "type": "Project"  
    },  
    "operation": "update",  
    "created_at": "2019-07-26 21:40:12.278876",  
    "event_type": "Shotgun_Task_Change",  
    "session_uuid": "cd7be24e-afea-11e9-a059-0242ac110006",  
    "attribute_name": "sg_status_list",  
    "event_log_entry_id": 249963  
  }  
}
```

# Viewing the Webhook request

▼  sg\_status\_list changed on Task Layout by user Brandon Ashworth

Request | **Response**

## Headers

```
{  
    "date": "Fri, 26 Jul 2019 21:40:15 GMT",  
    "server": "Werkzeug/0.15.5 Python/2.7.16",  
    "content-type": "text/html; charset=utf-8",  
    "content-length": "2"  
}
```

## Body

Ok

Note: Delivery logs are automatically removed from the system after five days.

# Viewing the Webhook request

## Task Status Flipper

UNSTABLE

Edit

Disable

Delete Webhook

Webhook URL: [http://e265224a.ngrok.io/task\\_status](http://e265224a.ngrok.io/task_status)

Projects: Dev Day

Triggered By: Task update

Validate SSL Certificate: true

Secret Token: \*\*\*\*\*

## Deliveries

Status...



Older than...



Entity Name...



Clear Filters

Redeliver Selected

▼  sg\_status\_list changed on Task Light by user Brandon Ashworth

7/26/2019, 2:46:11 PM

Request | Response

### Headers

```
{  
    "date": "Fri, 26 Jul 2019 21:46:11 GMT",  
    "server": "Werkzeug/0.15.5 Python/2.7.16",  
    "content-type": "text/html; charset=utf-8",  
    "content-length": "40"  
}
```

### Body

```
Can't authenticate script 'task_flipper'
```

## Other ways to use Webhooks

---



## Web Automation Tools

- IFTTT - If This Then That
- Zapier
  - Update a Google Spreadsheet every time a Shot's status is updated.



Name **your zap** Zap Settings

1. TRIGGER

**Set up this step**

**Choose App**

Choose Trigger  
Test This Step

+ Add a Step

## Choose a Trigger App

 Search hundreds of apps...**YOUR APPS**

Google Sheets

[show all](#)**BUILT-IN APPS**

Email



Zapier Manager



Code



Schedule



Email Parser



Push



RSS



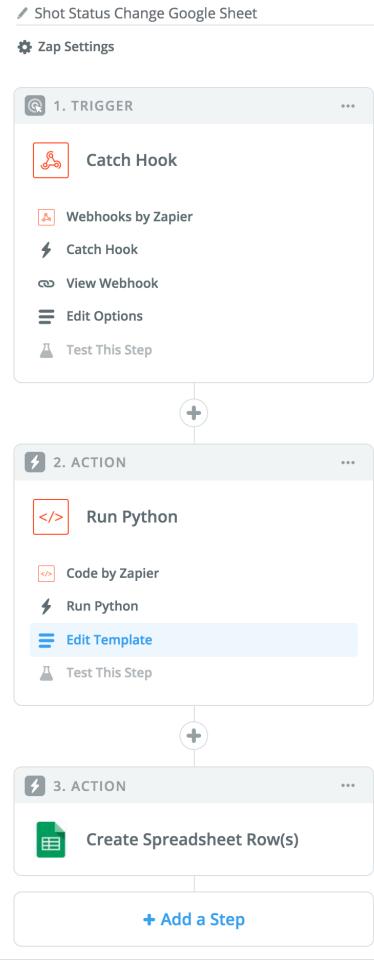
Webhooks



IMAP



Weather



## Set up Code by Zapier Run Python

Setup Preview [Learn more](#)

ON

## Input Data (optional)

What input data should we provide to your code (as strings) via a dictionary set to a variable named `input_data`?

shot\_id

Step 1 No data



-



## Code (required)

Warning! This is an advanced trigger! Please read the [Code documentation](#) for more information.

```
HOST = 'https://yoursite.shotgunstudio.com'

creds = {
    'grant_type': 'client_credentials',
    'client_id': 'zapier',
    'client_secret': 'api-key'
}

token_resp = requests.post(
    '{}/api/v1/auth/access_token'.format(HOST),
    data=creds,
    headers={'Content-Type': 'application/x-www-form-urlencoded'})
tokens = token_resp.json()

request_headers = {
    'Authorization': '{} {} {}'.format(**tokens)
}

shot_data = requests.get('{}/api/v1/entity/shots/{}'.format(HOST,
    input_data['shot_id']), headers=request_headers)
output = shot_data.json()
```

# Webhooks Beta - Coming Soon



webhooks-beta@shotgunsoftware.com



<https://app.sli.do/event/10wljptx>

# Thank You!

Before switching over to a more freeform discussion, we just wanted to say thanks for sharing your time with us!

Start a Shotgun eval



<https://www.shotgunsoftware.com/signup>

If you haven't started playing around with Shotgun, but like what you're seeing, take it for a test drive.

Take a look at our roadmap



<https://www.shotgunsoftware.com/roadmap>

If you want to know what's cooking or what we are thinking about tackling next, this should be your first stop.

Checkout our new forums



<https://community.shotgunsoftware.com>

We've just launch a new community forum where you can chat with us and other Shotgun users.

