

Asana Integration

Every 2 minutes, the appliance syncs to Asana:

- We filter for devices updated since the 'day we go live,' which is a different day for the Detroit and LA appliances
- A device must have an Asana asset tag entered into the user interface
- A device must have at least one wipe
- We have one mapping for the Asana fields
- We have another mapping for an HTML entry
- The HTML entry is different depending on whether it is Apple, a Laptop, a Desktop, or a Tablet
- We push the new Asana mappings and the HTML at the same time, per item
- If the asset tag does not exist in Asana, we log the error and blacklist the item

NOTE: Any changes to the below mappings require authorization from Human IT.

The mappings as of October 6, 2022 are as follows:

```
erp_template = [  
  {'Serial Number': '<machine.system.serial>'},  
  {'Model Number': '<machine.system.model>'},  
  {'Model Number Apple': '<machine.apple.a_model>'},  
  {'MPN Apple': '<machine.apple.order_number>'},  
  {'MPN': '<machine.system.part_number>'},  
  {'Date Wiped': '<job.wipes.date>', # MM/DD/YYYY},  
  {'Drive Condition?': '<job.wipes.result>', # 'Wiped (D2)' if successful, 'Not Removed (D1)' if  
failed},  
  {'Have Caddy?': 'yes_or_no_any(<machine.storage.disks.serial>)', # if hard drive exists,  
then "Yes"},  
  {'Hard Drive Capacity': 'summarize_by_count(<machine.storage.disks.size>  
<machine.storage.disks.smart_log.kind> <machine.storage.disks.smart_log.form_factor>)'},  
  {'Form Factor': '<machine.system.chassis>'},  
  {'Networking Speed': '<machine.network_cards.network_cards.speed>'},  
  {'Screen Size': 'inch_to_double_quote(<machine.displays.displays.diagonal>)'},  
  {'OS and Edition': 'Not Installed'},  
  {'CPU Type': '<machine.cpus.cpus.vendor> <machine.cpus.cpus.series>'},  
  {'CPU Model': 'merge_core_series(<machine.cpus.cpus.series>  
<machine.cpus.cpus.model>)'},  
  {'CPU Speed': '<machine.cpus.cpus.speed>'},  
  {'Number of CPU': 'number_of(<machine.cpus.cpus.model>)'},  
  {'Memory Capacity': '<machine.rams.ram>'},  
  {'Primary Drive': 'first(<machine.storage.disks.size> <machine.storage.disks.smart_log.kind>  
<machine.storage.disks.smart_log.form_factor>)'},
```

```
{'Graphics Processor': '<machine.graphics_card.graphics_card.model>'},
{'Optical Drive': 'yes_or_no_any(<machine.storage.optical.model>)'},
{'Ports': 'Please view our pictures for details'},
{'Resolution': '<machine.displays.displays.resolution>'},
{'Wi-Fi': 'yes_or_no_any(<machine.network_cards.network_cards>)', # i have no idea
what this is supposed to be
{'Type': '<machine.apple.wipeos_id>',
{'EMC': '<machine.apple.emc>',
{'Model Year': '<machine.apple.release_era>',
{'# of HDDs': 'number_of(<machine.storage.disks.serial>)'},
{'HDD Serial Number': 'first(<machine.storage.disks.serial>)'},
]
```

```
html_template = [
{'vendor': '<machine.system.vendor>',
{'model': '<machine.system.model>',
{'chassis': '<machine.system.chassis>',
{'screen_size': 'first(<machine.displays.displays.diagonal>)'},
{'screen_size_backup': '<machine.apple.family>',
{'form_factor': '<machine.user_fields.form_factor>',
{'resolution': '<machine.displays.displays.resolution>',
{'wipe_status': '<job.wipes.result>', # We need this for goofiness that a template parser
can't do
{'cpu_vendor_series': '<machine.cpus.cpus.company> <machine.cpus.cpus.series>',
{'cpu_series_model': 'merge_core_series(<machine.cpus.cpus.series>
<machine.cpus.cpus.model>)'},
{'number_of_cpu': 'number_of(<machine.cpus.cpus.company>)'},
{'cpu_speed': '<machine.cpus.cpus.speed> GHz'},
{'memory_capacity': '<machine.rams.ram>',
{'memory_type': '<machine.rams.ram_sticks.type>',
{'gpu': '<machine.graphic_cards.graphic_cards.model>',
{'optical_drive': 'yes_or_no_any(<machine.storage.optical_drives.capability>)'},
{'wifi': 'yes_or_no_any(<machine.extra.webcam>)'},
{'model_year': '<machine.apple.year>',
{'emc': '<machine.apple.emc>',
{'nice_name': '<machine.apple.nice_name>',
{'a_number': '<machine.apple.a_model>',
{'wipeos_id': '<machine.apple.wipeos_id>',
{'battery': '<machine.batteries.batteries.model>',
{'battery_cycle_count': '<machine.batteries.batteries.cycle_count>',
{'drive_type': 'first(<machine.storage.disks.size> <machine.storage.disks.smart_log.kind>)'
]
```