
nuix
Simple. Powerful. Precise.


NUIX 2ND ANNUAL
DEVELOPER DAY
SMACK DOWN
CHALLENGE
nuix

- Have the Nuix API documentation available
  - Installed with Nuix:
    `C:\Program Files\Nuix\Nuix 7.0\doc`
  - Or Online (requires download site login):
    `https://download.nuix.com/releases/desktop/stable/docs`

Challenge 1

- **Create a script which tags items based on an input CSV**
  - First column of CSV is the tag name to be applied
  - Second column of CSV is the query to run
  - Skip the first row (headers)
  - See Challenge 1 folder from GitHub for input CSV
- **Hints**
  - You will need to read from a CSV, see “Code Snippets” on the wiki
  - You will need to search the case for the items responsive to each query
  - You will need to tag (annotate) the collection of items returned for each search

Challenge 1 Solution


```
require "csv"

# Obtain BulkAnnotater
annotater = $utilities.getBulkAnnotater

# Iterate each record, first row will be skipped
CSV.foreach("C:\\Path\\Input.csv",{:headers => :first_row}) do |row|
    # Get CSV row values
    tag = row[0]
    query = row[1]

    # Search for the hits to this query
    hits = $current_case.search(query)

    # Tag the hits with the tag
    annotater.addTag(tag,hits)
end
```

Challenge 2

- **Report to a CSV, counts based on each tag in the case**
  - First column should be the tag name
  - Second column should be the count of items with the tag
  - Third column should be the de-duplicated count of items with the tag
- **Hints**
  - You will need to get a listing of all tags in the case
  - You will need to search for each tag
  - You will need to de-duplicate the results of each search
  - You will need to write to a CSV, see “Code Snippets” on the wiki

Challenge 2 Solution


```
require "csv"

# Get ItemUtility
item_utility = $utilities.getItemUtility
# Get all tags in the current case
all_case_tags = $current_case.getAllTags

# Open our output CSV for writing
CSV.open("C:\\Path\\Output.csv","w:utf-8") do |csv|
	# Write headers
	csv << [
		"Tag",
		"Count",
		"Deduped Count",
	]

	# Write out record for each tag
	all_case_tags.each do |tag_name|
		# Determine count of items with this tag
		items = $current_case.search("tag:#{tag_name}")
		# Deduplicate
		deduplicated_items = item_utility.deduplicate(items)
		# Write tag name and count to CSV
		csv << [
			tag_name,
			items.size,
			deduplicated_items.size,
		]
	end
end
```

Challenge 3

- **Create a worker side script, for each item processed**
  - Tag an item with "**`WSSHit`**" if it matches the search "**`cat OR dog`**"
  - If the item has a "**`Communication`**", record the communication's date's year as a new metadata property named "**`CommunicationYear`**"
- Hints
  - See "Worker Side Scripting" on the wiki
  - You can get the year from a "**`DateTime`**" object by calling "**`getYear`**"

Challenge 3 Solution


```
def nuix_worker_item_callback(worker_item)
	# Get source item
	source_item = worker_item.getSourceItem

	# Search and apply tag if matches
	if source_item.matchesSearch("cat OR dog")
		worker_item.addTag("WSSHit")
	end

	# Get communication
	communication = source_item.getCommunication
	# Does this item have a communication (is it not nil/null)?
	if communication.nil? == false
		# Get the communication date
		communication_date = communication.getDateTime
		# Get the year
		year = communication_date.getYear
		# Get the current properties hash/map
		properties = source_item.getProperties
		# Add the new property
		properties["CommunicationYear"] = year
		# Store changes back
		worker_item.setItemProperties(properties)
	end
end
```

FIND OUT MORE:

- nuix.com
- nuix.com/blog
- twitter.com/nuix
- facebook.com/nuixsoftware
- linkedin.com/company/nuix
- youtube.com/nuixsoftware