

# CTS API Introduction and Demo

Hubert Hickman

## CTS API

The Cancer Clinical Trials Search API <https://www.cancer.gov/syndication/api> is a NCI supported API that provides a wide range of features including trial information and search capabilities. Much of the API content uses the NCI Thesaurus.

- CTS API Helpful links:
  - Swagger page - <https://clinicaltrialsapi.cancer.gov/doc>
  - You will need an API key - signup for a key here : <https://clinicaltrialsapi.cancer.gov/signin>
  - Rate limits for standard 1 request per second
  - Maximum number of trials returned per request is 50.
- Searching is supported via POST or GET. I typically use POST for convenience.
- Searching by code, strings, geolocation, and much much more.
- Has both unstructured and structured eligibility criteria.

## EVS API

NCI's Enterprise Vocabulary Services provides several tools and downloads of the National Institute Thesaurus.

- NCI Thesaurus Helpful links:
  - EVS Explore (defaults to NCIt) <https://evsexplore.semantics.cancer.gov/evsexplore/welcome>
  - EVS API : <https://evsexplore.semantics.cancer.gov/evsexplore/evsapi>
    - \* No API key required
    - \* Will timeout if too much activity (need sleep/backoff timing code)
  - NCIt releases are available as for download as well as TSV and OWL files.
    - \* <https://evs.nci.nih.gov/evs-download/thesaurus-downloads>

- I have used the TSV extensively in my work (computing the transitive closure of NCIt digraph and enumerating all the paths in a database table - it speeds up inheritance computations)
- TSV file is a subset of the OWL representation.

## Simple Example

This example shows how to query the CTS API to get a count of active treatment trials. It retrieves one trial.

```
import requests
import sys
import json
import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import time

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
                  "Content-Type": "application/json"}
includes = ['nct_id',
            'diseases',
            'biomarkers',
            'prior_therapy',
            'brief_title'
            ]

active_treatment_trials_that_are_recruiting = {'current_trial_status': 'Active',
        'sites.recruitment_status' : 'ACTIVE',
        'primary_purpose': 'TREATMENT',
        'size':1,
        'from':0,
        'include':includes
        }

r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
                  json = active_treatment_trials_that_are_recruiting, headers=cts_api_header)

j = r.json()
```

```
mr.JSON(j)
time.sleep(2)
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

## Diseases

Expanding upon the above example, let us look at the diseases returned from the trial.

```
import requests
import sys
import json
import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import itables
import time

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
                  "Content-Type": "application/json"}
includes = ['nct_id',
            'diseases',
            'biomarkers',
            'prior_therapy',
            'brief_title'
            ]

active_treatment_trials_that_are_recruiting = {'current_trial_status': 'Active',
        'sites.recruitment_status' : 'ACTIVE',
        'primary_purpose': 'TREATMENT',
        'size':1,
        'from':0,
        'include':includes
        }

r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
```

```

        json = active_treatment_trials_that_are_recruiting, headers=cts_api_header

j = r.json()
diseases_df = pd.DataFrame(j['data'][0]['diseases'])
itables.show(diseases_df, column_filters="header")
time.sleep(2)

```

TRIAL level diseases are those coded to the trial by clinical trial abstractors at NCI. TREE level diseases go ‘up’ the NCIt digraph.

Lead disease is/are the most focused trial level disease for the trial. Other trial level diseases are generally more broad or alternative matches.

## Biomarkers

Biomarkers are abstracted as discrete data using NCIt codes. Biomarkers have been coded on new trials for a couple of years now – older trials may not have them even if the trial calls has biomarkers as inclusion/exclusion criteria.

As with diseases, the TREE terms go ‘up’ the NCIt digraph. Note that NCIt is a multiaxial hierarchy, and hence you may 1 parent node.

```

import requests
import sys
import json
import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import itables
import time

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
                  "Content-Type": "application/json"}
includes = ['nct_id',
            'diseases',
            'biomarkers',
            'prior_therapy',
            'brief_title'
            ]

```

```

active_treatment_trials_that_are_recruiting = {'current_trial_status': 'Active',
        'sites.recruitment_status' : 'ACTIVE',
        'primary_purpose': 'TREATMENT',
        'size':1,
        'from':0,
        'include':includes
    }
r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
        json = active_treatment_trials_that_are_recruiting, headers=cts_api_header)

j = r.json()
biomarkers_df = pd.DataFrame(j['data'][0]['biomarkers'])
itables.show(biomarkers_df, column_filters="header",
        buttons=["pageLength", "copyHtml5", "csvHtml5", "excelHtml5"])
time.sleep(2)

```

## Retrieving a trial by NCT\_ID that has prior therapy records

The trial NCT02914405 contains prior therapy terms. These are shown as a dataframe and as a rather busy digraph.

```

import requests
import sys
import json
import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import itables
import time
import networkx as nx
import graphviz
import matplotlib.pyplot as plt

plt.clf()

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
        "Content-Type": "application/json"}

```

```

# No 'includes' so get everything

trial_ids = {
    'nct_id': ['NCT02914405']
}

r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
                  json = trial_ids, headers=cts_api_header)

j = r.json()
mr.JSON(j)
time.sleep(2)
prior_therapy_df = pd.DataFrame(j['data'][0]['prior_therapy'])
itables.show(prior_therapy_df, column_filters="header",
              buttons=["pageLength", "copyHtml5", "csvHtml5", "excelHtml5"])

# Now set up the graph for display
# set node
node_label_dict = {}
node_color_dict = {}
node_size_dict = {}

G = nx.DiGraph()
prior_therapy_df['node_label'] = prior_therapy_df['nci_thesaurus_concept_id'] + '\n' + prior_therapy_df['prior_therapy']

trial_pt_df = prior_therapy_df[prior_therapy_df['inclusion_indicator'] == 'TRIAL']

for index, pt in prior_therapy_df.iterrows():
    node_label_dict[str(pt['nci_thesaurus_concept_id'])] = str(pt['node_label'])

    if str(pt['inclusion_indicator']) == 'TRIAL':
        node_color_dict[str(pt['nci_thesaurus_concept_id'])] = 'green'
        node_size_dict[str(pt['nci_thesaurus_concept_id'])] = 1000
    else:
        node_color_dict[str(pt['nci_thesaurus_concept_id'])] = 'yellow'
        node_size_dict[str(pt['nci_thesaurus_concept_id'])] = 500

G.add_node(str(pt['nci_thesaurus_concept_id']))

```

```

    for p in pt['parents']:
        #print('adding edge ',str(pt['nci_thesaurus_concept_id']), str(p) )
        G.add_edge(str(pt['nci_thesaurus_concept_id']), str(p))

color_list = []
node_size_list = []
for node in G:
    color_list.append(node_color_dict[node])
    node_size_list.append(node_size_dict[node])

pos = nx.nx_pydot.graphviz_layout(G, prog="dot")
#pos = nx.spring_layout(G, k=20.0)
plt.clf()
fig = plt.gcf()
fig.set_size_inches(12,12)
nx.draw(G, with_labels=True,
        labels = node_label_dict,
        node_color = color_list,
        node_size = node_size_list)
plt.show()
plt.savefig('prior_therapy_example.pdf',dpi=300, format = 'pdf' )

```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>





```

import requests
import sys
import json
import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import itables
import time

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
                  "Content-Type": "application/json"}

# No 'includes' so get everything

trial_ids = {
    'nct_id': ['NCT05183035', 'NCT05188170', 'NCT02914405']
}

r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
                  json = trial_ids, headers=cts_api_header)

j = r.json()
mr.JSON(j)
time.sleep(2)
#diseases_df = pd.DataFrame(j['data'][0]['diseases'])
#itables.show(diseases_df, column_filters="header",
# buttons=["pageLength", "copyHtml5", "csvHtml5", "excelHtml5"])

```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

## Search for AML trials by NCIt code

Now search for AML trials by NCIt code.

```

import requests
import sys
import json

```

```

import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import itables
import time

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
                  "Content-Type": "application/json"}

aml_trials = {'current_trial_status': 'Active',
              'sites.recruitment_status' : 'ACTIVE',
              'primary_purpose': 'TREATMENT',
              'size':10,
              'from':0,
              'diseases.nci_thesaurus_concept_id': ['C3171']
              # 'include':includes
              }

r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
                  json = aml_trials, headers=cts_api_header)

j = r.json()

mr.JSON(j)

aml_df = pd.DataFrame(j['data'])
itables.show(aml_df, column_filters="header",
              buttons=["pageLength", "copyHtml5", "csvHtml5", "excelHtml5"])
time.sleep(2)

```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

## AML Trials within 100 miles of my location

```

import requests
import sys
import json
import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import itables
import time

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
                  "Content-Type": "application/json"}

aml_trials = {'current_trial_status': 'Active',
              'sites.recruitment_status' : 'ACTIVE',
              'primary_purpose': 'TREATMENT',
              'size':10,
              'from':0,
              'diseases.nci_thesaurus_concept_id': ['C3171'],
              'sites.org_coordinates_lat': 41.2749,
              'sites.org_coordinates_lon': -96.0212,
              'sites.org_coordinates_dist': '100 mi'
              # 'include':includes
              }

r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
                  json = aml_trials, headers=cts_api_header)

j = r.json()

mr.JSON(j)

aml_df = pd.DataFrame(j['data'])
itables.show(aml_df, column_filters="header",
              buttons=["pageLength", "copyHtml5", "csvHtml5", "excelHtml5"])
time.sleep(2)

```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

## Trials with EGFR positive as an inclusion or exclusion criteria

```
import requests
import sys
import json
import os
import pandas as pd
from dotenv import dotenv_values
import mercury as mr
import itables
import time

config = dotenv_values('.env')
CTS_API_KEY=config['CTS_API_KEY']
cts_api_header = {"x-api-key": CTS_API_KEY,
                  "Content-Type": "application/json"}

egfr_pos_trials = {'current_trial_status': 'Active',
                   'sites.recruitment_status' : 'ACTIVE',
                   'primary_purpose': 'TREATMENT',
                   'size':10,
                   'from':0,
                   'biomarkers.nci_thesaurus_concept_id': ['C134501'],
                   'biomarkers.eligibility_criterion':'inclusion'
                   # 'include':includes
                   }

r = requests.post('https://clinicaltrialsapi.cancer.gov/api/v2/trials',
                  json = egfr_pos_trials, headers=cts_api_header)

j = r.json()

mr.JSON(j)

egfr_df = pd.DataFrame(j['data'])
itables.show(egfr_df, column_filters="header",
              buttons=["pageLength", "copyHtml5", "csvHtml5", "excelHtml5"])
summary_df = egfr_df[['nct_id', 'official_title']]
itables.show(summary_df, column_filters="header",
              buttons=["pageLength", "copyHtml5", "csvHtml5", "excelHtml5"])
time.sleep(2)
```

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>