API Bulk Endpoints

Last modified on 09/09/2025 7:47 pm EDT

DrChrono APIs have endpoints to grab data in bulk. Normally, our API response has a page size limit of 250, but our bulk APIs can obtain up to 1000 results.

Our current list of bulk APIs.

Appointments

Patients

Line Item

Transactions

Prescription Messages

Eligibility Checks

Clinical Notes

Clinical Note Field Values

Steps to perform request

1. POST a request with the filter and pagination options
2. Make a GET request with the "uuid" field from the POST response to fetch the results list once available
   -> While the list is being generated, the GET response will have "status": "In progress"

   Note: Higher page sizes will take longer to generate.

The list generated is cached in DrChrono for 1 hour. If the list is expired, the UUID will return an invalid message. The POST request will need to be sent to generate a new list.

Example

In this example, we will use the Appointments API. The response includes a "uuid" value that will be used in the GET request to retrieve the result list.

POST REQUEST:

```
POST https://app.drchrono.com/api/appointments_list?verbose=true&since=2024-02-01&page=1&page_size=1000&order_by=updated_at
```

RESPONSE:

```
HTTP Status 201

{
"status": "In progress",
"uuid": "dfeb7fa2-13dd-4b73-902d-c86ffd8d7f54",
"description": "Call https://app.drchrono.com/api/appointments_list?uuid=dfeb7fa2-13dd-4b73-902d-c86ffd8d7f54 to get the list"
}
```

GET REQUEST:

```
GET https://app.drchrono.com/api/appointments_list?uuid=dfeb7fa2-13dd-4b73-902d-c86ffd8d7f54
```

RESPONSE:

```
{
"status": "Complete",
"pagination": {
"count": 4428,
"pages": 5,
"page_size": 1000,
"page": 1
},
"uuid": "dfeb7fa2-13dd-4b73-902d-c86ffd8d7f54",
"results": [
{...}
]
}
```

To paginate to the next page you will need to make another POST request using the query parameter, "page=integer" [1 .. 1000].

EXPIRED RESULT RESPONSE

If the requested UUID is expired, the detail will specify the error.

```
{
"detail": "Invalid uuid specified"
}
```

Python Example Code

This script can be used for any of our bulk API endpoints. The goal is to obtain all the data by iterating through all the pages.

```
import json
import time
import requests
import textwrap

token = '6pxCIEj162QS80Ae6btWniZihkHHJv'
patient_data = []
mode='appointments'

def print_roundtrip(response, *args, **kwargs):
""" print full req and resp """
format_headers = lambda d: '\n'.join(f'{k}: {v}' for k, v in d.items())
```

```
print(textwrap.dedent('''
---------- request ----------
{req.method} {req.url}
{reqhdrs}
{req.body}
---------- response ----------
{res.status_code} {res.reason} {res.url}
{reshdrs}
{res.text}
''').format(
req=response.request,
res=response,
reqhdrs=format_headers(response.request.headers),
reshdrs=format_headers(response.headers),
))

hooks = { 'response': print_roundtrip }

def get_page(page=1, page_size=1000, since='2023-06-01'):
""" call the {records}_list asynchronously to POST a batch request and poll for results """
response = requests.post(
f'https://app.drchrono.com/api/{mode}_list?page_size={page_size}&page={page}&since={since}&verbose=true',
headers={
'Authorization': f'Bearer {token}',
'Content-Type': 'application/json'},
hooks=hooks,
)
response.raise_for_status()
handle = response.json()['uuid']
# await the request to be complete. check periodically.
while response.json()['status'] != 'Complete':
time.sleep(10)
response = requests.get(f'https://app.drchrono.com/api/{mode}_list?uuid={handle}',
headers={f'Authorization': f'Bearer {token}'},hooks=hooks,
)
response.raise_for_status()
return response

print('fetching page 1 of UNKNOWN')
response = get_page(1)
print('initial submission complete. record counts: ' + json.dumps(response.json()['pagination']))
patient_data.extend(response.json()['results'])
total_records = int(response.json()['pagination']['count'])
total_pages = int(response.json()['pagination']['pages'])
current_page = int(response.json()['pagination']['page']) + 1
print(f'gathered {[x["id"] for x in response.json()["results"]]}')

while current_page <= total_pages:
print(f'fetching page {current_page} of {total_pages}')
response = get_page(current_page)
patient_data.extend(response.json()['results'])
print(f'gathered {[x["id"] for x in response.json()["results"]]}')
current_page = current_page + 1

print(f'total records gathered: {len(patient_data)}. records reported by api: {total_records}')
print(f'{[x["id"] for x in patient_data]}')
```