Getting Started with DrChrono API with Python and C#

Last modified on 09/09/2025 7:28 pm EDT

Here are Python and C# scripts created by the DrChrono Engineering team to help you get started with connecting to DrChrono APIs.

Python Script

```
import datetime
import requests
import json
import threading
import webbrowser
import urllib.parse

from http.server import BaseHTTPRequestHandler, HTTPServer

"""
This python script demonstrates how to implement the OAUTH2 flow where:
1. a user loads a browser url which logs into drchrono.
2. drchrono will prompt the user to authorize a custom web application to operate using their credentials.
3. once the user authorizes the action they will be redirected to the web application with an authorization code.
4. the web application will receive the request containing the authorization code.
5. the web application will exchange the authorization code for an API token.
6. the web application will use the API token to retrieve a list of appointments on behalf of the user.

This script requires python runtime 3.6+ and python "requests" library
Always keep in mind client_secret and access_token are sensitive information and subject to HIPAA least-use policy

Example output:
c:\temp>python.exe thescript.py
Starting web server at http://localhost:8000
https://app.drchrono.com/o/authorize/?scope=calendar%3Aread%20patients%3Aread%20clinical%3Aread&response_type=code&redirect_uri=http://localhost:8000&client_id=XXX
Received GET with URL /?code=YYY
Found auth code YYY
POSTing to token endpoint using client-id/client-secret/auth-code/redirect-uri to get a token
Server responded with {"access_token": "ZZZ", "token_type": "Bearer", "expires_in": 172800, "refresh_token": "AAA", "scope": "calendar:read patients:read clinical:read"}
Token expires at 2024-04-18 09:15:29
Using the token to load appointments
Server responded with {"previous":null,"results":[{...}]}
"""


# this should be a URL to a web server which you control.
# when the user logs in to drchrono they will be redirected to this location with the authorization_code in the querystring.
# this authorization code can be used to log on to the API as the user.
redirect_uri = 'http://localhost:8000'
# you must set up an API application inside drchrono to retrieve these values
client_id = 'DRCHRONO_PROVIDED'
client_secret = 'DRCHRONO_PROVIDED'


class MyWebServer(BaseHTTPRequestHandler):
    """ Replace this with your web application listening on a specific URL """
    def do_GET(self):
        """ listening for a GET request which happens after user logs on and is redirected """
```

```
            listening for a GET request which happens after user logs on and is redirected
        try:
            print(f'Received GET request with URL {self.path}')
            authorization_code = self.path.split('code=')[1]
            print(f'Found auth code {authorization_code}')
            print(f'POSTing to token endpoint using client-id/client-secret/auth-code/redirect-uri to get a token')
            response = requests.post(f'https://app.drchrono.com/o/token/',
                params={
                    'grant_type': 'authorization_code',
                    'client_id': client_id,
                    'client_secret': client_secret,
                    'redirect_uri': redirect_uri,
                    'code': authorization_code,
                },
            )
            response.raise_for_status()
            print(f'Server responded with {response.text}')
            token = response.json()
            expires_in = datetime.datetime.now() + datetime.timedelta(seconds=token['expires_in'])
            print(f'Token expires at {expires_in.strftime("%Y-%m-%d %H:%M:%S")}')
            print(f'Using the token to load appointments')
            response = requests.get(
                f'https://app.drchrono.com/api/appointments?since=2024-01-01',
                headers={
                    'Authorization': f'Bearer {token["access_token"]}',
                    'Content-Type': 'application/json',
                },
            )
            response.raise_for_status()
            print(f'Server responded with {response.text}')
        finally:
            raise SystemExit


print(f'Starting disposable web server at {redirect_uri}')
server_thread = threading.Thread(target=lambda: HTTPServer(('localhost', 8000), MyWebServer).serve_forever())
server_thread.start()

# scopes allow you as the client to limit the capabilities the API token will have
permitted_scopes = ['calendar:read', 'patients:read', 'clinical:read']
scope_string = urllib.parse.quote(" ".join(permitted_scopes), safe='')
# this is a url link which the user would click on to initiate the OAUTH2 process
browser_url = f'https://app.drchrono.com/o/authorize/?scope={scope_string}&response_type=code&redirect_uri={re
direct_uri}&client_id={client_id}'
webbrowser.open(browser_url, new=2)

# the web server awaits the user to perform the logon process and get redirected back to the web server with the aut
h token
server_thread.join()
```

C# Script

```
using System.Net;
using System.Text.Json;
using System.Web;
using System.Net.Http.Headers;
using System.Text.Json.Nodes;

/*
This C# script demonstrates how to implement the OAUTH2 flow where:
1. a user loads a browser url which logs into drchrono.
2. drchrono will prompt the user to authorize a custom web application to operate using their credentials.
3. once the user authorizes the action they will be redirected to the web application with an authorization code.
```

```
4. the web application will receive the request containing the authorization code.
5. the web application will exchange the authorization code for an API token.
6. the web application will use the API token to retrieve a list of appointments on behalf of the user.

This script requires dotnet runtime version 6+. Due to the custom http listener the user must be running with adminis
trator.
Always keep in mind client_secret and access_token are sensitive information and subject to HIPAA least-use policy

Example output:
c:\temp>ConsoleApp1.exe
Starting web server at http://localhost:8000
https://app.drchrono.com/o/authorize/?scope=calendar%3Aread%20patients%3Aread%20clinical%3Aread&response_t
ype=code&redirect_uri=http://localhost:8000&client_id=XXX
Received GET with URL /?code=YYY
Found auth code YYY
POSTing to token endpoint using client-id/client-secret/auth-code/redirect-uri to get a token
Server responded with {"access_token": "ZZZ", "token_type": "Bearer", "expires_in": 172800, "refresh_token": "AAA",
"scope": "calendar:read patients:read clinical:read"}
Token expires at 2024-04-18 09:15:29
Using the token to load appointments
Server responded with {"previous":null,"results":[{...}]}
*/

// this should be a URL to a web server which you control.
// when the user logs in to drchrono they will be redirected to this location with the authorization_code in the querystri
ng.
// this authorization code can be used to log on to the API as the user.
string redirect_uri = "http://localhost:8000";
// you must set up an API application inside drchrono to retrieve these values
string client_id = "DRCHRONO_PROVIDED";
string client_secret = "DRCHRONO_PROVIDED";


// this is a disposable web server for the purpose of example which you should replace with your web application
// it will await the user request when they are redirected from login
var listener = new HttpListener();
listener.Prefixes.Add("http://*:8000/");
listener.Start();

// scopes allow you as the client to limit the capabilities the API session will have (HIPAA least-use policy)
// in this example we limit the scopes to read appointment data
var permittedScopes = new[] { "calendar:read", "patients:read", "clinical:read" };
var scopeString = HttpUtility.UrlEncode(string.Join(" ", permittedScopes));
// this is a link which a user would click on to initiate the OAUTH2 process
var browserUrl = $"https://app.drchrono.com/o/authorize/?scope={scopeString}&response_type=code&redirect_uri=
{redirect_uri}&client_id={client_id}";
// in this example we open the user's default browser to the link
System.Diagnostics.Process.Start(new System.Diagnostics.ProcessStartInfo { FileName = browserUrl, UseShellExecut
e = true });

// wait here for the user to log in and authorize the OAUTH2 session
var context = listener.GetContext();
Console.WriteLine($"Received GET request with URL {context.Request.Url}");
var authorizationCode = HttpUtility.ParseQueryString(context.Request.Url.Query)["code"];
Console.WriteLine($"Found auth code {authorizationCode}");

using (var httpClient = new HttpClient())
{
    Console.WriteLine("POSTing to token resource using client-id/client-secret/auth-code/redirect-uri to get a token");
    var response = httpClient.PostAsync($"https://app.drchrono.com/o/token/", new FormUrlEncodedContent(new[]
    {
        new KeyValuePair<string, string>("grant_type", "authorization_code"),
        new KeyValuePair<string, string>("client_id", client_id),
        new KeyValuePair<string, string>("client_secret", client_secret),
```

```
        new KeyValuePair<string, string>("redirect_uri", redirect_uri),
        new KeyValuePair<string, string>("code", authorizationCode)
    })).Result;
    response.EnsureSuccessStatusCode();
    dynamic tokenData = JsonSerializer.Deserialize<JsonNode>(response.Content.ReadAsStringAsync().Result);
    var expiresIn = DateTime.Now.AddSeconds(tokenData["expires_in"].GetValue<int>());
    Console.WriteLine($"Token expires at {expiresIn.ToString("yyyy-MM-dd HH:mm:ss")}");

    Console.WriteLine("Using the token to load appointments");
    using (var requestMessage = new HttpRequestMessage(HttpMethod.Get, $"https://app.drchrono.com/api/appointm
ents?since=2024-01-01"))
    {
        requestMessage.Headers.Authorization = new AuthenticationHeaderValue("Bearer", tokenData["access_token"].
GetValue<string>());
        response = httpClient.SendAsync(requestMessage).Result;
    }
    response.EnsureSuccessStatusCode();
    Console.WriteLine($"Server responded with {response.Content.ReadAsStringAsync().Result}");
}
```