```python
import requests  # for getting URL
import json  # for parsing json
from datetime import datetime  # datetime parsing
import pytz  # timezone adjusting
import csv  # for making csv files
import pandas as pd
import numpy as np
import os


def first_date(row):
  return row['days'][0]


def get_sleep_stat(row, sleep_stat: str):
  if row['sleep.sleeps'] == []:
    return np.nan
  return row['sleep.sleeps'][0][sleep_stat]


def  get_access_token(username: str, password: str):
    # GET ACCESS TOKEN
    # Post credentials
    r = requests.post("https://api-7.whoop.com/oauth/token", json={
    "grant_type": "password",
    "issueRefresh": False,
    "password": password,
    "username": username })
    if r.status_code != 200:
        print("Fail - Credentials rejected.")
        return 1
    # print("Success - Credentials accepted")
    return r

def get_user_data_raw(access_token, start_date='2000-01-01T00:00:00.000Z',
end_date='2030-01-01T00:00:00.000Z', url='https://api-
7.whoop.com/users/{}/cycles'):

    ################################################################
    # Exit if fail
    r = access_token

    # Set userid/token variables
    userid = r.json()['user']['id']
    access_token = r.json()['access_token']

    # GET DATA
    # Download data
    url = url.format(userid)

    params = {
    'start': start_date,
    'end': end_date
    }

    headers = {
    'Authorization': 'bearer {}'.format(access_token)
    }
    r = requests.get(url, params=params, headers=headers)

```

```
    # Check if user/auth are accepted
    if r.status_code != 200:
        # print("Fail - User ID / auth token rejected.")
        return 1

    # print("Success - User ID / auth token accepted")
    # print(json.dumps(r.json()))
    data_raw = r.json()
    return data_raw


def get_user_data_df(access_token,
                     start_date='2000-01-01T00:00:00.000Z',
                     end_date='2030-01-01T00:00:00.000Z',
                     url='https://api-7.whoop.com/users/{}/cycles'):
    # get raw whoop data
    data_raw = get_user_data_raw(access_token, start_date, end_date, url)
    # convert json to pandas df
    df = pd.json_normalize(data_raw)

    df['date'] = df.apply (lambda row: first_date(row), axis=1)
    df['sleep.sws.duration'] = df.apply (lambda row: get_sleep_stat(row, 'slowWaveSleepDuration'), axis=1)
    df['sleep.quality.duration'] = df.apply (lambda row: get_sleep_stat(row, 'qualityDuration'), axis=1)
    df['sleep.light.duration'] = df.apply (lambda row: get_sleep_stat(row, 'lightSleepDuration'), axis=1)
    df['sleep.rem.duration'] = df.apply (lambda row: get_sleep_stat(row, 'remSleepDuration'), axis=1)
    df['sleep.wake.duration'] = df.apply (lambda row: get_sleep_stat(row, 'wakeDuration'), axis=1)
    df['respiratoryRate'] = df.apply (lambda row: get_sleep_stat(row, 'respiratoryRate'), axis=1)
    df['sleep.efficiency'] = df.apply (lambda row: get_sleep_stat(row, 'sleepEfficiency'), axis=1)
    df['sleep.consistency'] = df.apply (lambda row: get_sleep_stat(row, 'sleepConsistency'), axis=1)
    return df

```