```python
import tbapy, csv, ast, time

filename = "houston_hopper_2018.csv"

# source: https://towardsdatascience.com/flattening-json-objects-in-python-f5343c794b10
def flatten_json(y):
    def flatten(x, name=''):
        if type(x) is dict:
            for a in x:
                flatten(x[a], name + a + '_')
        elif type(x) is list:
            i = 0
            for a in x:
                flatten(a, name + str(i) + '_')
                i += 1
        else:
            out[name[:-1]] = x

    out = {}
    flatten(y)
    return out

def generate_extra_keys(lower: int, upper: int, *args) -> list:
    keys = []
    for i in range(lower, upper):
        for arg in args:
            keys.append(arg.format(i))
    return keys

start_time = time.time()

# api key-- I don't have any write permissions, so no misgivings about making this public
tba = tbapy.TBA('oJFBtEK2LI8jnZ4CT7snUeVg05zHm882LaB5bTEdAuhx0Lu0A6GKqXLdxYvkq5o2')

# matches I want to record data from
match_keys = tba.team_matches(team=1160, year=2018, keys=True)
match_keys = tba.event_matches(event='2018hop', keys=True)

extra_json_keys = generate_extra_keys(1, 6, 'videos_{0}_key', 'videos_{0}_type') + \
    generate_extra_keys(0, 3, 'alliances_red_dq_team_keys_{0}', 'alliances_blue_dq_team_keys_{0}')
    generate_extra_keys(0, 3, 'alliances_red_surrogate_team_keys_{0}', 'alliances_blue_surrogate_te

# we need to do this process for an arbitrary match so that we can get the keys for the DictWriter
match_arbitrary = tba.match(key=match_keys[0])
dict_arbitrary = ast.literal_eval(match_arbitrary.json())
flattened_arbitrary = flatten_json(dict_arbitrary)
for key in extra_json_keys:
    if key not in flattened_arbitrary:
        flattened_arbitrary[key] = ''
with open(filename, 'w') as f:
    w = csv.DictWriter(f, flattened_arbitrary.keys(), lineterminator = '\n')
    w.writeheader()

    # now we start recording data in earnest
    for match_key in match_keys:
        m = tba.match(key=match_key)
        d = ast.literal_eval(m.json())
        f = flatten_json(d)
        w.writerow(f)

end_time = time.time()
print(time.strftime("%H:%M:%S elapsed", time.gmtime(end_time - start_time)))
```