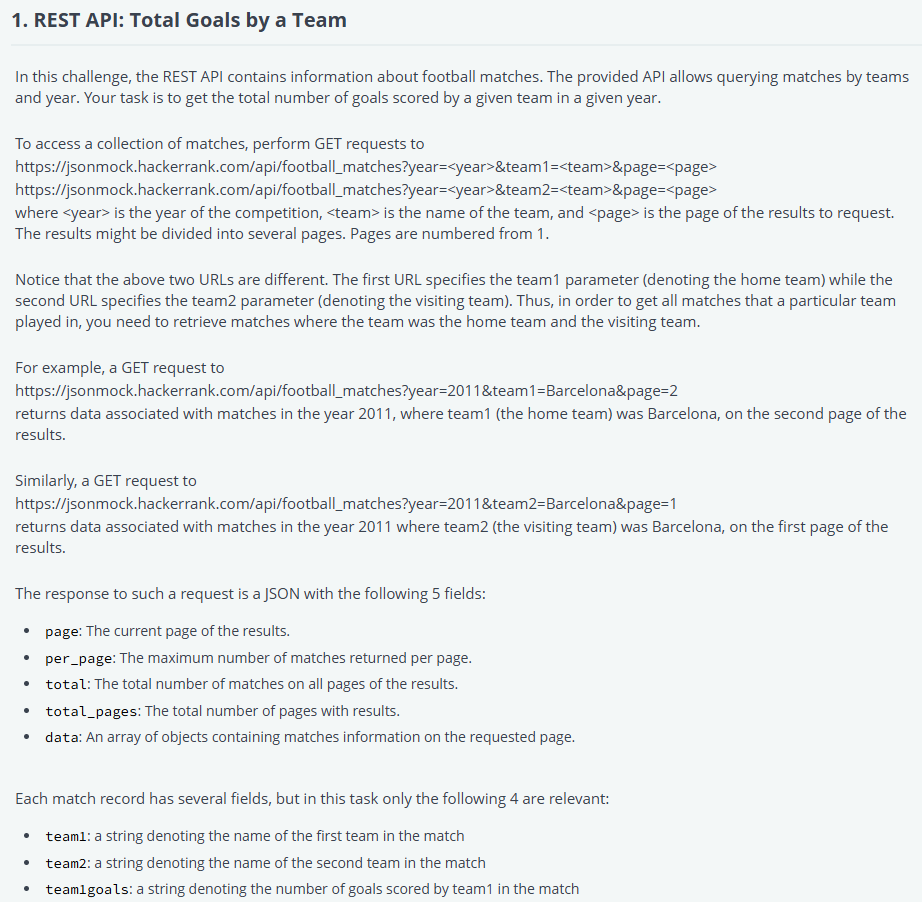
# REST API: Total Goals by a Team



'use strict';

const fs = require('fs');

const fetch = require('node-fetch');

process.stdin.resume();

process.stdin.setEncoding('utf-8');

let inputString = '';

let currentLine = 0;

process.stdin.on('data', function(inputStdin) {

    inputString += inputStdin;

});

process.stdin.on('end', function() {

    inputString = inputString.split('\n');

    main();

});

function readLine() {

    return inputString[currentLine++];

}

/\*

 \* Complete the 'getTotalGoals' function below.

 \*

 \* The function is expected to return an INTEGER.

 \* The function accepts following parameters:

 \*  1. STRING team

 \*  2. INTEGER year

 \*/

async function getTotalGoals(team, year) {

    let totalGoals = 0;

    let page = 1;

    while (true) {

        const url = `https://jsonmock.hackerrank.com/api/football\_matches?year=${year}&team1=${team}&page=${page}`;

        const response = await fetch(url);

        const data = await response.json();

        // Sum the goals from team1 (home team)

        data.data.forEach(match => {

            if (match.team1 === team) {

                totalGoals += parseInt(match.team1goals);

            }

        });

        // Sum the goals from team2 (away team)

        const urlForAway = `https://jsonmock.hackerrank.com/api/football\_matches?year=${year}&team2=${team}&page=${page}`;

        const awayResponse = await fetch(urlForAway);

        const awayData = await awayResponse.json();

        awayData.data.forEach(match => {

            if (match.team2 === team) {

                totalGoals += parseInt(match.team2goals);

            }

        });

        // If we've reached the last page, stop fetching

        if (page >= data.total\_pages) {

            break;

        }

        // Otherwise, move to the next page

        page++;

    }

    return totalGoals;

}

async function main() {

    const ws = fs.createWriteStream(process.env.OUTPUT\_PATH);

    const team = readLine();

    const year = parseInt(readLine().trim(), 10);

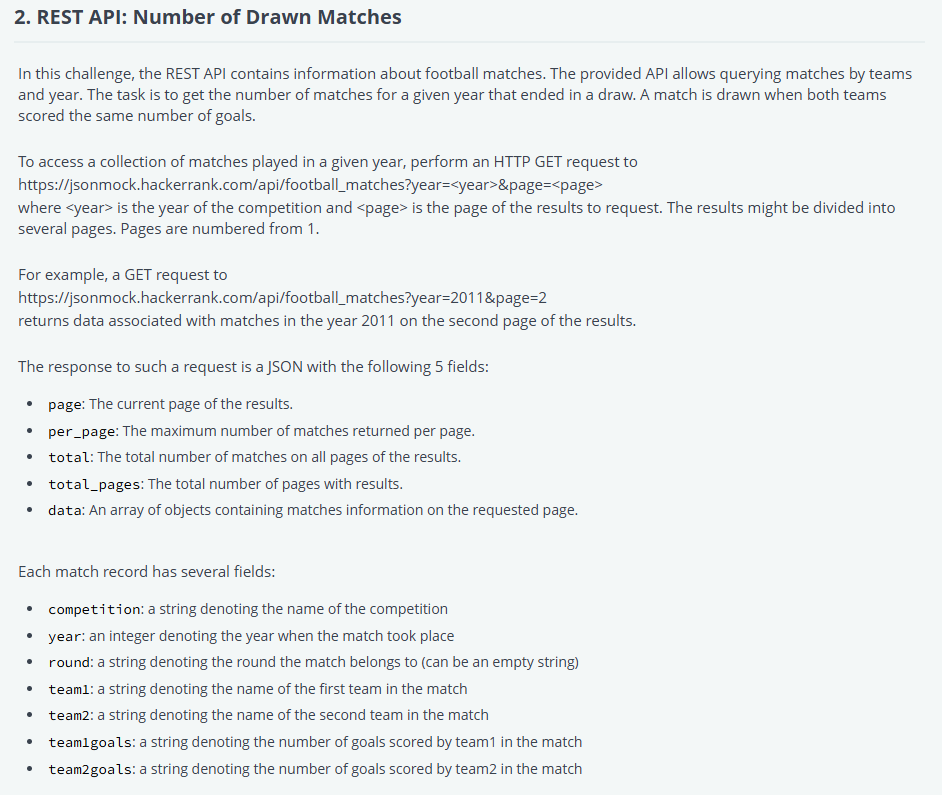
    const result = await getTotalGoals(team, year);

    ws.write(result + '\n');

    ws.end();

}

# REST API: Number of Drawn Matches



'use strict';

const fs = require('fs');

const fetch = require('node-fetch');

process.stdin.resume();

process.stdin.setEncoding('utf-8');

let inputString = '';

let currentLine = 0;

process.stdin.on('data', function(inputStdin) {

    inputString += inputStdin;

});

process.stdin.on('end', function() {

    inputString = inputString.split('\n');

    main();

});

function readLine() {

    return inputString[currentLine++];

}

/\*

 \* Complete the 'getNumDraws' function below.

 \*

 \* The function is expected to return an INTEGER.

 \* The function accepts INTEGER year as parameter.

 \*/

async function getNumDraws(year) {

    let goals=[];

    let ans=0;

    for(let goal=0;goal<=10;goal++)

    {

        const myPromise = fetch(`https://jsonmock.hackerrank.com/api/football\_matches?year=${year}&team1goals=`+goal+`&team2goals=`+goal)

                            .then(res => res.json())

                            .then(data => {

                                console.log(data.total,goal);

                                return data.total

                            });

        goals.push(myPromise);

    }

    await Promise.all(goals).then((array)=>{

        array.forEach( item =>{

            ans+=item;

        })

    })

    return ans;

}

async function main() {

    const ws = fs.createWriteStream(process.env.OUTPUT\_PATH);

    const year = parseInt(readLine().trim(), 10);

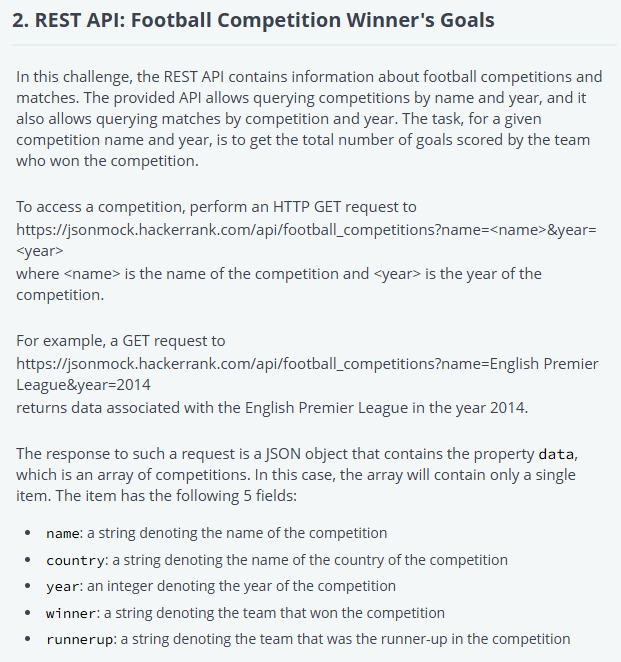
    const result = await getNumDraws(year);

    ws.write(result + '\n');

    ws.end();

}

# REST API: Football Competition Winner’s Goals



import requests

import json

import os

def getWinnerTotalGoals(competition, year):

    goals = 0

    url = 'https://jsonmock.hackerrank.com/api/football\_competitions?name=' +competition+ '&year=' +str(year)

    response = requests.request('GET', url, headers={}, data={})

    winner = json.loads(response.text.encode('utf8'))['data'][0]['winner']

    for team in ['team1', 'team2']:

        url = 'https://jsonmock.hackerrank.com/api/football\_matches?competition=' +competition+ '&year=' +str(year)+ '&' +team+ '=' +winner+ '&page=1'

        response = requests.request('GET', url, headers={}, data={})

        total\_pages = json.loads(response.text.encode('utf8'))['total\_pages']

        for i in range(1, total\_pages+1):

            url = 'https://jsonmock.hackerrank.com/api/football\_matches?competition=' +competition+ '&year=' +str(year)+ '&' +team+ '=' +winner+ '&page=' +str(i)

            response = requests.request('GET', url, headers={}, data={})

            r = json.loads(response.text.encode('utf8'))

            r\_data = r['data']

            for record in r\_data:

                goals += int(record[team+'goals'])

    return goals

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    competition = input()

    year = int(input().strip())

    result = getWinnerTotalGoals(competition, year)

    fptr.write(str(result) + '\n')

    fptr.close()