Университет ИТМО
Кафедра вычислительной техники

Технологии программрования

Лабораторная работа №5

Студенты:
Куклина Мария,
Кириллова Анастасия, P3401
Преподаватель:
Оголюк А.А.

Санкт-Петербург, 2017

1. Листинг

```
import http.server
import json
import os
import urllib
import urllib.parse
import random

from apiclient.discovery import build
from http.server import BaseHTTPRequestHandler, HTTPServer
from urllib.request import urlretrieve

# global variables
names = './names.json';
credits = './credits.txt';

creds = ();
service = None;
people_list = [];

# Global State
current_person_idx = 0;
user_level = 10;
user_score = 0;

# Requests.
def build_service(key):
    return build("customsearch", "v1", developerKey=key);

def execute_request(service, cx_id, request,):
    return service.cse().list(
        q=request,
        cx=cx_id,
        searchType='image',
        num=1,
        imgType='clipart',
        fileType='png',
        safe= 'off'
    ).execute();

def get_image(service, cx_id, objs, idx):
    query = objs[idx]['name'] + ' ' + objs[idx]['description'] + ' photo';
    res = execute_request(service, cx_id, query);
    assert 'items' in res;
    return res['items'][0]['link'];

# Credits.
def get_credits(filename):
    with open(filename) as cfile:
        credits = cfile.read().replace('\n', '').split(' ');
    return (credits[0], credits[1]);
```

```
# Read names list.
def get_object(filename):
    with open(filename) as jfile:
        json_string = jfile.read();
    return json.loads(json_string);

# User's guess.
def get_submit_data(text):
    name = people_list[current_person_idx]['name'];
    answ = text[0];
    global user_score;
    if answ.lower() == name.lower():
        user_score += 1;
    return str(user_score);

# Load new image.
def get_image_data():
    global current_person_idx;
    global people_list;

    new_idx = int(random.randrange(0, user_level));

    if 'path' in people_list[new_idx]:
        image_path = people_list[new_idx]['path'];
    else:
        image_path = get_image(service, creds[1], people_list, new_idx);
        people_list[new_idx]['path'] = image_path;
    current_person_idx = new_idx;
    return image_path;

# Server side shit

def send_data(self, data, memtype):
    self.send_response(200)
    self.send_header('Content-type', memtype)
    self.end_headers()
    self.wfile.write(data)
    return

def load_file(path):
    data = "";
    with open(path, "rb") as ffile:
        data = ffile.read();
    return data;

class testHTTPServer_RequestHandler(BaseHTTPRequestHandler):
    def do_GET(self):
        global user_score;
        qs = urllib.parse.urlparse(self.path);
        qc = urllib.parse.parse_qs(qs.query);

        data = "";
        memtype = "text/html";
```

```
        if "/image" == self.path: # Load new image
            data = bytes(get_image_data(), "utf-8");
            memtype = 'text/plain'
        elif "submit" in qc: # User's guess.
            data = bytes(get_submit_data(qc["submit"]), "utf-8")
            memtype = 'text/plain'
        elif "level" in qc: # Set level and reture game page
            global user_level
            user_level = int(qc["level"][0]);
            data = load_file('game.html');
            user_score = 0;
        else:
            data = load_file('index.html');
            user_score = 0;

        send_data(self, data , memtype);
        return

def run():
  server_address = ('0.0.0.0', 8080)
  httpd = HTTPServer(server_address, testHTTPServer_RequestHandler)
  httpd.serve_forever()

def main():
    global people_list;
    global creds;
    global service;
    global current_person_idx;

    create_dir(path_db);
    creds       = get_credits(credits);
    service     = build_service(creds[0]);
    people_list = get_object(names);
    current_person_idx = 0;
    run();

main()

<html>
        <title>GuessWho: initial page</title>
        <body>
                <fieldset>
                <legend>Select level:</legend>
                <form id="form_id" method="get">
                <br><input id="first_level"  type="radio" value="10" name=
                    "level" checked>1-10 names</input></br>
                <br><input id="second_level" type="radio" value="50" name=
                    "level">1-50 names</input></br>
                <br><input id="third_level"  type="radio" value="100" name
                    ="level">1-100 names</input></br>
                <br><input type="submit" value="Submit"></br>
                </form>
                </fieldset>
```

```
        </body>
</html>

<html>
        <title>Game!</title>
        <head>
                <script type="text/javascript">
                    function get_next_image() {
                var xhttp = new XMLHttpRequest();
                xhttp.onreadystatechange = function() {
                    if (this.readyState == 4 && this.status == 200) {
                        var res = this.responseText;
                        console.log('get res: ' + res);
                        document.getElementById("image").setAttribute('src
                            ', res);
                    }
                };
                xhttp.open("GET", "image", true);
                xhttp.send();
                    }
            function send_data(text) {
                var xhttp = new XMLHttpRequest();
                xhttp.onreadystatechange = function() {
                    if (this.readyState == 4 && this.status == 200) {
                        var res = this.responseText;
                        var prev = document.getElementById("res").
                            innerHTML;
                        document.getElementById("res").innerHTML =
                            parseInt(res);
                    }
                };
                xhttp.open("GET", "?submit=" + text, true);
                xhttp.send();
            }

                        function handle_submit(form) {
                                var t = document.getElementById("text_id")
                                    ;
                                send_data(t.value);
                                get_next_image();
                        }

                        get_next_image();
                </script>
        </head>
        <body>
                <!-- Image -->
                <div id="image_pane">
                        <img id="image" src="" width="400" height="200">
                </div>
                <!-- Textbox -->
                <div id="textbox">
                        <form id="form" onsubmit="handle_submit(this);
                            return␣false;">
```

```
                        <br><input id="text_id" type="text" size=50
                            required="required"></br>
                        <br><input type="submit" value="Submit"></br>
                        </form>
                </div>
                <!-- Score -->
                <div id="score">
                        <br>Score: <span id="res">0</span></br>
                </div>
        </body>
</html>
```