Constants.swift

//

//  Constants.swift

//  Project-part2

//

//  Created by Garry Singh on 11/06/17.

//  Copyright © 2017 Garry Singh. All rights reserved.

//

import Foundation

struct Constants

{

    static let APIBaseURL = "https://api.flickr.com/services/rest/"

    struct ParameterKeys

    {

        static let Method = "method"

        static let APIKey = "api\_key"

        static let Extras = "extras"

        static let Format = "format"

        static let NoJSONCallback = "nojsoncallback"

        static let Tags = "tags"

        static let Text = "text"

        static let token = "auth\_token"

        static let Signature = "api\_sig"

    }

    struct ParameterValues{

        static let APIKey = "da1683fbc9cb177dd116d5454db1079a"

        static let ResponseFormat = "json"

        static let DisableJSONCallback = "1"

        static let PhotosMethod = "flickr.photos.search"

      // static let GalleryID = "66911286-72157677375993701"

        static let MediumURL = "url\_m"

        static let TagKeywords = "MIT"

        static let TextValue = "Auckland"

        static let TokenValue = "72157681886417924-bb11bb25c36d6915"

        static let SignatureValue = "22b469a917e8743668fbbc15afbf38dd"

    }

   // auth\_token=72157681886417924-bb11bb25c36d6915    &api\_sig=22b469a917e8743668fbbc15afbf38dd

}

ViewController.swift

//

//  ViewController.swift

//  Project-part2

//

//  Created by Garry Singh on 11/06/17.

//  Copyright © 2017 Garry Singh. All rights reserved.

//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var locationTextField: UITextField!

    @IBOutlet weak var nameTextField: UITextField!

    @IBOutlet weak var imageView: UIImageView!

    override func viewDidLoad() {

        super.viewDidLoad()

        // Do any additional setup after loading the view, typically from a nib.

    }

    @IBAction func btnSearch(\_ sender: UIButton) {

      //

       // api\_sig=8d0b464c6bbc2c48736cf8f3589cf3e1

        grabAnImage()

    }

    private func grabAnImage()

    {

        let textKeyword = locationTextField.text! // entered value "Auckland"

        let tag = nameTextField.text!     /// the vakue entered is "MIT"

        let methodParameters = [ Constants.ParameterKeys.Method:Constants.ParameterValues.PhotosMethod,

                                 Constants.ParameterKeys.APIKey:Constants.ParameterValues.APIKey,

            Constants.ParameterKeys.Tags:tag,

            Constants.ParameterKeys.Text:textKeyword,

            Constants.ParameterKeys.Extras:Constants.ParameterValues.MediumURL,

            Constants.ParameterKeys.Format:Constants.ParameterValues.ResponseFormat,

            Constants.ParameterKeys.NoJSONCallback:Constants.ParameterValues.DisableJSONCallback,

            Constants.ParameterKeys.token:Constants.ParameterValues.TokenValue,

            Constants.ParameterKeys.Signature:Constants.ParameterValues.SignatureValue

        ]

      let stringURL = Constants.APIBaseURL + escapedParameters(parameters: methodParameters)

        //This is the flickr link to get photos

        /\*

         let s = "<https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=8a743f2eab9475d5d8df7c56965d328c&tags=MIT&text=Auckland&extras=url_m&format=json&nojsoncallback=1&auth_token=72157681886417924-bb11bb25c36d6915&api_sig=22b469a917e8743668fbbc15afbf38dd>"

        \*/

 let url = URL(string: stringURL)!

        let request = URLRequest(url: url)

        let task = URLSession.shared.dataTask(with: request){

            (data,response,error) in

            if error == nil

            {

                if let data = data

                {

                    let parsedResult :[String:AnyObject]!

                    do

                    {

                        parsedResult = try! JSONSerialization.jsonObject(with: data, options: .allowFragments) as! [String:AnyObject]

                        }

                    print(parsedResult)

                }

            }

        }

        task.resume()

    }

    private func escapedParameters(parameters : [String:String]) -> String

    {

        if parameters.isEmpty{

            return ""

        }

        else

        {

            var keyValuePairs = [String]()

            for (key, value) in parameters

            {

                //convert the value into string

                let stringValue = "\(value)"

               let escapedValue = stringValue.addingPercentEncoding(withAllowedCharacters: .urlQueryAllowed)

                keyValuePairs.append(key + "=" + "\(escapedValue!)")

            }

            return "?\(keyValuePairs.joined(separator: "&"))"

        }

    }

}