

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №6
з дисципліни
“Програмування мобільних систем / Розроблення клієнтських додатків
для мобільних платформ”

Виконав:
студент групи ІВ-83
ЗК ІВ-8328
Чорноморець Влад

Варіант: 1

```
>>> 8328%6 + 1  
1
```

```
>>> 8328%2 + 1  
1
```

Варіант 1

Предметна область – фільми

URL-адреса для отримання даних –
`http://www.omdbapi.com/?apikey=API_KEY&s=REQUEST&page=1`, де
API_KEY – 7e9fe69e, **REQUEST** – запит із поля для пошуку.

Зверніть увагу на те, що використання ключа **API_KEY** обмежене до 1000 запитів на день, якщо буде вичерпано кількість запитів на день для наведеного ключа, то можна отримати власний ключ, зареєструвавшись за [посиланням](#).

Варіант 1

Предметна область – фільми

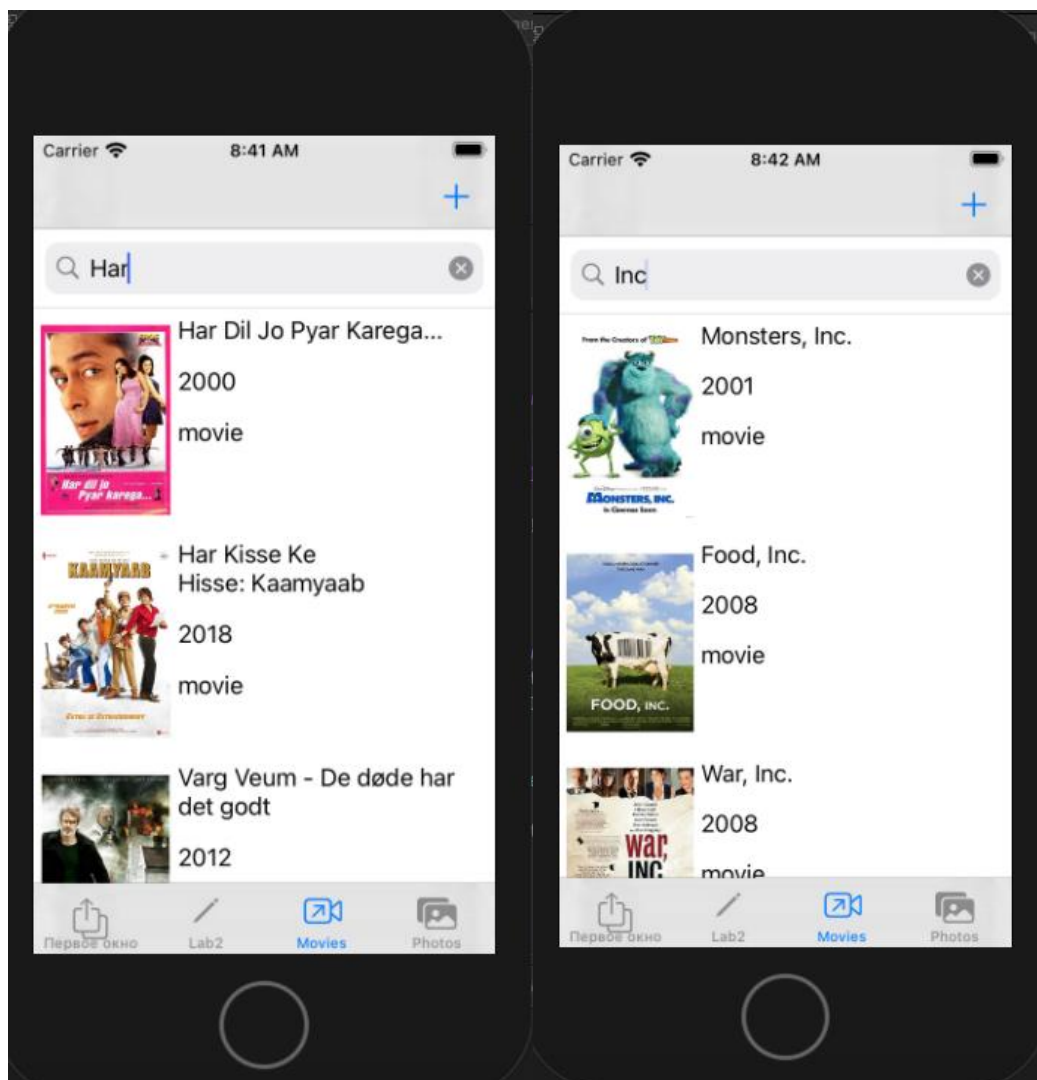
URL-адреса для отримання даних –
`http://www.omdbapi.com/?apikey=API_KEY&i=IDENTIFIER`, де **API_KEY**
– 7e9fe69e, **IDENTIFIER** – ідентифікатор відповідного фільму (поле
imdbID із сутності фільму).

Зверніть увагу на те, що використання ключа **API_KEY** обмежене до 1000 запитів на день, якщо буде вичерпано кількість запитів на день для наведеного ключа, то можна отримати власний ключ, зареєструвавшись за [посиланням](#).

Варіант 1

REQUEST – “yellow+flowers”
COUNT – 27

Скріншот роботи додатка





Лістинг коду

NetworkManager.swift

```
//  
// NetworkManager.swift  
// LabWork1.1  
//  
// Created by Vladislav on 24.05.2021.  
//  
  
import Foundation  
import UIKit  
  
protocol NetworkManagerDelegate {  
    func getMovies(with name: String, completion: @escaping (Data?, Error?) -> Void)  
    func getMovieDetails(with identifier: String, completion: @escaping (Data?, Error?) -> Void)  
    func getPhotos(with request: String, count: Int, completion: @escaping (Data?, Error?) -> Void)  
    func getImage(with coverURL: String, completion: @escaping (UIImage?, Error?) -> Void)  
    func cancel()  
}  
  
final class NetworkManager: NSObject, NetworkManagerDelegate, URLSessionDelegate  
{  
    static let sharad = NetworkManager()  
  
    lazy var urlsession: URLSession = {  
        let config = URLSessionConfiguration.default  
        return URLSession(configuration: config)  
    }()  
  
    var datatask:URLSessionTask?  
}  
  
extension NetworkManager {  
    func getMovies(with name: String, completion: @escaping (Data?, Error?) -> Void) {  
        let urlString =  
            "http://www.omdbapi.com/?apikey=7e9fe69e&s=\(name)&page=1"  
  
        guard let components = URLComponents(string: urlString), let url =  
            components.url else {return}  
        DispatchQueue.global(qos: .background).async {[weak self] in  
            self?.datatask = self?.urlsession.dataTask(with: URLRequest(url:  
url)) { (data, response, error) in  
                guard let httpresponse = response as? HTTPURLResponse,  
                    let data = data,  
                    httpresponse.statusCode == 200 else {return}  
                DispatchQueue.main.async {  
                    completion(data, nil)  
                }  
            }  
            self?.datatask?.resume()  
        }  
    }  
}
```

```

    }
    func getMovieDetails(with identifier: String, completion: @escaping (Data?,
Error?) -> Void) {
        let urlString =
"http://www.omdbapi.com/?apikey=7e9fe69e&i=\(identifier)"
        guard let components = URLComponents(string: urlString), let url =
components.url else {return}
        DispatchQueue.global(qos: .background).async { [weak self] in
            self?.datatask = self?.urlsession.dataTask(with: URLRequest(url:
url)) { (data, response, error) in
                guard let httpResponse = response as? HTTPURLResponse,
                    let data = data,
                    httpResponse.statusCode == 200 else {return}
                print(httpResponse.statusCode)
                DispatchQueue.main.async {
                    completion(data, nil)
                }
            }
            self?.datatask?.resume()
        }
    }

    func getPhotos(with req: String, count: Int, completion: @escaping (Data?,
Error?) -> Void) {
        let urlString = "https://pixabay.com/api/?key=19193969-
87191e5db266905fe8936d565&q=\(req)&image_type=photo&per_page=\(count)"
        guard let components = URLComponents(string: urlString), let url =
components.url else {return}
        DispatchQueue.global(qos: .background).async { [weak self] in
            self?.datatask = self?.urlsession.dataTask(with: URLRequest(url:
url)) { (data, response, error) in
                guard let httpResponse = response as? HTTPURLResponse,
                    let data = data,
                    httpResponse.statusCode == 200 else {return}
                DispatchQueue.main.async {
                    completion(data, nil)
                }
            }
            self?.datatask?.resume()
        }
    }

    func cancel() {
        datatask?.cancel()
    }

    func getImage(with imageURL: String, completion: @escaping (UIImage?,
Error?) -> Void) {
        guard let url = URL(string: imageURL) else {return}
        DispatchQueue.global(qos: .background).async { [weak self] in
            self?.datatask = self?.urlsession.downloadTask(with: url) {
(tempURL, response, error) in
                if let tempURL = tempURL,
                    let data = try? Data(contentsOf: tempURL),
                    let image = UIImage(data: data) {
                        DispatchQueue.main.async {
                            completion(image, nil)
                        }
                    }
            }
        }
    }

```

```
        } else {
            completion(nil, error)
        }
    }
    self?.datatask?.resume()
}
}
```

Висновок:

В цій лабораторній роботі була доповнена робота 5. Тепер дані у вкладках Movies та Photos беруться з інтернету по варіанту.