

```

//
// HomeViewController.swift
// PseudoTikTok
//
// Created by Никита Кулагин on 26.01.2025.
//

import UIKit

class HomeViewController: UIViewController {
    // Коллекция для отображения видео
    var collectionView: UICollectionView!
    // Массив с видео
    var videos: [VideoModel] = []

    override func viewDidLoad() {
        super.viewDidLoad()
        view.backgroundColor = .white
        // Настраиваем Collection View
        setupCollectionView()
        // Добавляем кнопку "Обновить" в правый верхний угол
        navigationItem.rightBarButtonItem = UIBarButtonItem(title:
"Обновить", style: .plain, target: self, action:
#selector(refreshFeed))
        // Загружаем видео
        loadVideos()
    }

    // Метод настройки Collection View
    func setupCollectionView() {
        // Определяем высоту ячейки с учетом Tab Bar
        let tabBarHeight = tabBarController?.tabBar.frame.height ??
49
        let itemHeight = view.frame.height - tabBarHeight
        // Создаем layout с вертикальной прокруткой и размером
ячеек
        let layout = UICollectionViewFlowLayout()
        layout.scrollDirection = .vertical
        layout.minimumLineSpacing = 0
        layout.minimumInteritemSpacing = 0
        // Устанавливаем размер ячеек
        layout.itemSize = CGSize(width: view.frame.width, height:
itemHeight)
        // Инициализируем коллекцию
        collectionView = UICollectionView(frame: CGRect(x: 0, y: 0,
width: view.frame.width, height: itemHeight), collectionViewLayout:
layout)
        collectionView.delegate = self
    }
}

```

```

        collectionView.dataSource = self
        collectionView.isPagingEnabled = true
        collectionView.showsVerticalScrollIndicator = false
        // Устанавливаем контентные отступы
        collectionView.contentInset = UIEdgeInsets(top: 0, left: 0,
bottom: tabBarHeight, right: 0)
        // Регистрируем ячейку
        collectionView.register(VideoCell.self,
forCellWithReuseIdentifier: "VideoCell")
        // Добавляем коллекцию на экран
        view.addSubview(collectionView)
    }

    // Метод загрузки видео
    func loadVideos() {
        // Очищаем текущий массив видео
        videos.removeAll()

        // Вызываем метод fetchVideos из NetworkingService
        NetworkingService.shared.fetchVideos { [weak self] result
in
            DispatchQueue.main.async {
                guard let self = self else { return }
                switch result {
                case .success(let fetchedVideos):
                    self.videos = fetchedVideos
                    self.collectionView.reloadData()
                case .failure(let error):
                    // Обрабатываем ошибку, если необходимо
                    print("Ошибка при загрузке видео:
\\(error.localizedDescription)")
                    // Вы можете показать пользователю уведомление
об ошибке
                }
            }
        }
    }

    // Метод для обновления ленты
    @objc func refreshFeed() {
        loadVideos()
    }
}

// Расширение для реализации протоколов UICollectionViewDelegate и
UICollectionViewDataSource
extension HomeController: UICollectionViewDelegate,
UICollectionViewDataSource {

```

```
func collectionView(_ collectionView: UICollectionView,
numberOfItemsInSection section: Int) -> Int {
    // Возвращаем количество видео
    return videos.count
}

func collectionView(_ collectionView: UICollectionView,
cellForItemAt indexPath: IndexPath) -> UICollectionViewCell {
    // Получаем ячейку и настраиваем ее
    let cell =
collectionView.dequeueReusableCell(withReuseIdentifier: "VideoCell",
for: indexPath) as! VideoCell
    let video = videos[indexPath.item]
    cell.configure(with: video)
    return cell
}
}
```