

```

//
// NetworkingService.swift
// PseudoTikTok
//
// Created by Никита Кулагин on 24.01.2025.
//

import Foundation
import FirebaseStorage
import FirebaseFirestore

class NetworkingService {

    // Singleton для глобального доступа
    static let shared = NetworkingService()

    private let storageRef = Storage.storage().reference()
    private let databaseRef = Firestore.firestore()

    private init() {}

    // MARK: – Upload Video Method

    /// Метод для загрузки видео в Firebase Storage и сохранения информации в
    Firestore
    func uploadVideo(url: URL, completion: @escaping
    (Result<String, Error>) -> Void) {
        // Получаем текущего пользователя
        guard let userID = Auth.auth().currentUser?.uid else {
            completion(.failure(NSError(domain: "Нет
авторизованного пользователя", code: 0, userInfo: nil)))
            return
        }

        // Создаем уникальный идентификатор для видео
        let videoID = UUID().uuidString

        // Ссылка на место хранения видео в Firebase Storage
        let videoRef = storageRef.child("videos/\(videoID).mp4")

        // Загружаем файл видео в Firebase Storage
        videoRef.putFile(from: url, metadata: nil) { metadata,
error in
            if let error = error {
                print("Ошибка загрузки видео:
\((error.localizedDescription)")
                completion(.failure(error))
            }
        }
    }
}

```

```

        return
    }

    // Получаем URL загруженного видео
    videoRef.downloadURL { url, error in
        if let error = error {
            print("Ошибка получения URL видео:
\\(error.localizedDescription)")
            completion(.failure(error))
            return
        }

        if let downloadURL = url {
            // Сохраняем информацию о видео в Firestore
            self.saveVideoInfo(videoID: videoID, userID:
userID, videoURL: downloadURL.absoluteString, description:
description) { result in
                switch result {
                case .success:
                    completion(.success(videoID))
                case .failure(let error):
                    completion(.failure(error))
                }
            }
        }
    }
}

/// Вспомогательный метод для сохранения информации о видео в Firestore
private func saveVideoInfo(videoID: String, userID: String,
videoURL: String, description: String, completion: @escaping
(Result<Void, Error>) -> Void) {
    let videosCollection = databaseRef.collection("videos")
    let data: [String: Any] = [
        "videoID": videoID,
        "userID": userID,
        "videoURL": videoURL,
        "description": description,
        "timestamp": FieldValue.serverTimestamp(),
        "likesCount": 0
    ]
    // Добавьте другие поля по необходимости
    videosCollection.document(videoID).setData(data) { error in
        if let error = error {
            print("Ошибка сохранения информации о видео:
\\(error.localizedDescription)")
            completion(.failure(error))
        }
    }
}

```

```

        } else {
            completion(.success(()))
        }
    }
}

// MARK: – Fetch Videos Method

/// Метод для получения списка видео из Firestore
func fetchVideos(completion: @escaping (Result<[VideoModel],
Error>) -> Void) {
    let videosCollection = databaseRef.collection("videos")
    videosCollection.order(by: "timestamp", descending:
true).getDocuments { snapshot, error in
        if let error = error {
            print("Ошибка получения списка видео:
\\(error.localizedDescription)")
            completion(.failure(error))
            return
        }

        var videos: [VideoModel] = []

        if let documents = snapshot?.documents {
            for document in documents {

                let data = document.data()
                let videoID = data["videoID"] as? String ?? ""
                let userID = data["userID"] as? String ?? ""
                let videoURL = data["videoURL"] as? String ??
""

                let description = data["description"] as?
String ?? ""

                let likesCount = data["likesCount"] as? Int ??
0

                let timestamp = (data["timestamp"] as?
Timestamp)?.dateValue() ?? Date()

                // Создаем объект VideoModel
                let video = VideoModel(
                    videoID: videoID,
                    userID: userID,
                    videoURL: videoURL,
                    description: description,
                    likesCount: likesCount,
                    timestamp: timestamp
                )
                videos.append(video)
            }
        }
    }
}

```

```
        }  
    }  
    completion(.success(videos))  
}  
}
```