

## «Комунікація даних користувача із використанням сокетів»

Вінницький національний технічний університет

### **Анотація**

*Дана наукова стаття розглядає використання сокетів для комунікації між користувачами в розробці iOS додатків мовою Swift. Запропоновані методи досліджуються та порівнюються з точки зору ефективності, надійності та безпеки. Розглядаються також різні підходи до реалізації комунікації, такі як використання бібліотек Starscream та Socket.IO.*

**Ключові слова:** сокети, комунікація, iOS розробка, Swift, Starscream, Socket.IO

### **Abstract**

*This scientific article considers the use of sockets for communication between users in iOS app development using the Swift language. The proposed methods are studied and compared in terms of efficiency, reliability, and security. Various approaches to implementing communication are also examined, such as the use of the Starscream and Socket.IO libraries.*

**Keywords:** sockets, communication, iOS development, Swift, Starscream, Socket.IO

### **Вступ**

Сокети забезпечують ефективний спосіб комунікації між клієнтом та сервером у реальному часі. В iOS розробці мовою Swift існують різні бібліотеки та фреймворки, які спрощують роботу з сокетом. Серед них виділяються Starscream та Socket.IO.

Сокети є інтерфейсом, що дозволяє програмам обмінюватися даними через мережу. Вони забезпечують двосторонній зв'язок, який можна використовувати для передачі даних у реальному часі. Сокети можуть працювати на основі різних протоколів, таких як TCP (Transmission Control Protocol) та UDP (User Datagram Protocol).

### **Основна частина**

Процес комунікації за допомогою сокетів зазвичай починається зі створення з'єднання між клієнтом та сервером. Серверний додаток створює сокет, який "слухає" певний порт для вхідних з'єднань. Клієнтський додаток створює сокет, який ініціює з'єднання з сервером через визначений IP-адрес та порт. Після встановлення з'єднання між клієнтом та сервером можна починати обмін даними.

Після встановлення з'єднання клієнт та сервер можуть обмінюватися повідомленнями у реальному часі. Сокети забезпечують безперервний двосторонній канал, через який можна відправляти та отримувати дані. В контексті iOS додатків, це дозволяє реалізувати функції, такі як чати, ігрові додатки з мультиплеєром, або будь-які інші додатки, де важлива миттєва передача інформації. Окрім встановлення основного з'єднання, важливою частиною комунікації через сокети є обробка подій, таких як з'єднання, відправка та отримання даних, а також розрив з'єднання. Наприклад, при успішному встановленні з'єднання клієнт може надіслати повідомлення про авторизацію або запит на отримання даних. Сервер, у свою чергу, може надсилати відповіді, підтвердження або запитувати додаткову інформацію.

У реальних умовах мережа може бути нестабільною, тому важливо забезпечити обробку помилок та автоматичне відновлення з'єднання. Це дозволяє уникнути втрати даних та забезпечує безперервну комунікацію між користувачами. Сучасні бібліотеки для роботи з сокетом, такі як Socket.IO, мають вбудовані механізми для автоматичного відновлення з'єднання у разі його втрати.

Іншим важливим аспектом є безпека комунікації. Сокети можуть передавати чутливі дані, тому важливо забезпечити їх захист від несанкціонованого доступу. Це може бути досягнуто за допомогою шифрування даних та використання безпечних протоколів, таких як TLS (Transport Layer Security).

Бібліотеки, такі як Starscream, підтримують шифрування на основі TLS, що забезпечує додатковий рівень захисту переданих даних.

При використанні сокетів в iOS додатках, розробники мають також враховувати оптимізацію мережевих ресурсів та енергозбереження. Використання сокетів у фоновому режимі може впливати на споживання енергії та продуктивність додатка. Тому важливо налаштувати додаток таким чином, щоб мінімізувати вплив на батарею пристрою, зберігаючи при цьому ефективність комунікації.

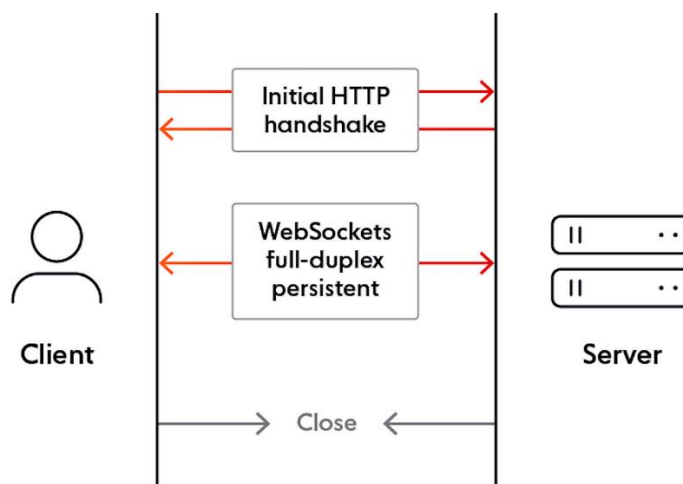


Рисунок 1 – Комунікація даних користувачів із застосуванням сокетів

Starscream є популярною бібліотекою для роботи з WebSocket в iOS додатках на Swift. Вона забезпечує двосторонню комунікацію в реальному часі, що робить її ідеальною для чатів, ігрових додатків та інших сервісів, де потрібна швидка передача даних. Основними особливостями Starscream є простота використання та налаштування, підтримка протоколу WebSocket, що дозволяє встановлювати постійне з'єднання між клієнтом та сервером, та обробка подій, таких як підключення, отримання повідомлень та відключення.

Socket.IO є ще однією популярною бібліотекою для реалізації сокет-з'єднань в iOS додатках. Вона надає більш високу абстракцію та додаткові можливості, такі як автоматичне відновлення з'єднання та крос-платформенність. Socket.IO підтримує різні транспортні протоколи, включаючи WebSocket та HTTP Long Polling, що робить її більш надійною у нестабільних мережах. Додатково, вона підтримує функції, такі як кімнати та простори імен, які допомагають організувати комунікацію між різними групами користувачів.

При порівнянні бібліотек Starscream та Socket.IO можна виділити наступні аспекти. Starscream зазвичай має кращу ефективність завдяки більш низькому рівню абстракції. Водночас, Socket.IO забезпечує більшу надійність завдяки підтримці різних транспортних протоколів та автоматичному відновленню з'єднання. З точки зору простоти використання, Starscream є простішим для базових задач, тоді як Socket.IO надає більше функціональних можливостей для складних сценаріїв.

Уявімо, що у вас є додаток для обміну повідомленнями. Використовуючи сокети, ви можете забезпечити миттєву доставку повідомлень між користувачами. При відкритті додатка клієнтський додаток встановлює з'єднання з сервером через сокет. Коли один користувач надсилає повідомлення, сервер отримує це повідомлення і відправляє його всім підключеним клієнтам. Завдяки двосторонній природі сокетів, нові повідомлення відразу з'являються у користувачів, забезпечуючи плавний та інтерактивний досвід.

## Висновки

У цій статті були розглянуті та порівняні два підходи до реалізації комунікації між користувачами в iOS додатках мовою Swift за допомогою сокетів: Starscream та Socket.IO. Кожен з цих підходів має свої переваги та недоліки, що робить їх придатними для різних задач та сценаріїв. Вибір бібліотеки залежить від конкретних вимог до ефективності, надійності та простоти реалізації комунікації.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Reynaud, M. (2019). Starscream: An efficient WebSocket library for Swift. GitHub. Retrieved from <https://github.com/daltoniam/Starscream>
2. Socket.IO (2020). Real-time engine. Retrieved from <https://socket.io>
3. Apple Inc. (2020). Swift Programming Language. Apple Developer Documentation.
4. Nguyen, D. (2018). Real-time communication with WebSockets in iOS. Medium. Retrieved from <https://medium.com>
5. Smith, J. (2017). Using Socket.IO with Swift. Ray Wenderlich. Retrieved from <https://www.raywenderlich.com>

**Велянський Сергій Андрійович** – студент групи ІКІ-23м, факультет інформаційних технологій і комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, [sergiy.velyanskiy48@gmail.com](mailto:sergiy.velyanskiy48@gmail.com).

**Снігур Анатолій Васильович** – канд. техн. наук, доцент, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, м. Вінниця.

**Velianskyi Serhii A.** – student of group 1KI – 23m, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, [sergiy.velyanskiy48@gmail.com](mailto:sergiy.velyanskiy48@gmail.com).

**Snigur Anatoliy V.** – Cand. Sc. (Eng.), Assistant Professor of the Computer Techniques Chair, Vinnytsia National Technical University, Vinnytsia.