МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Факультет прикладної математики та інформатики
Кафедра інформаційних систем

Звіт про проходження виробничої практики

Виконав студент 6 курсу, групи ПМіМ-22с
Кізло Тарас Михайлович

Керівник від кафедри
доцент Бернакевич Ірина Євстахіївна

Львів - 2021

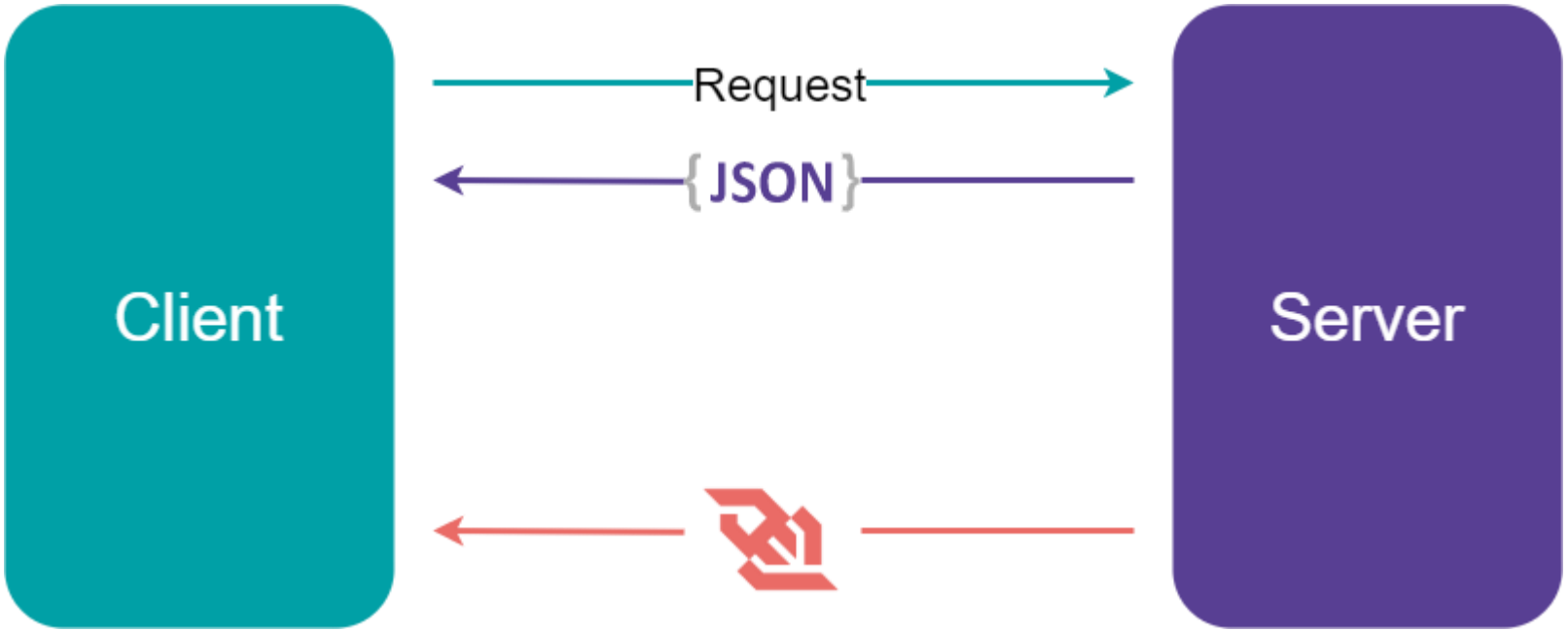
Зміст

- Постановка задачі
- Теоретичні відомості
- Огляд проведеної роботи
- Висновки

Постановка задачі

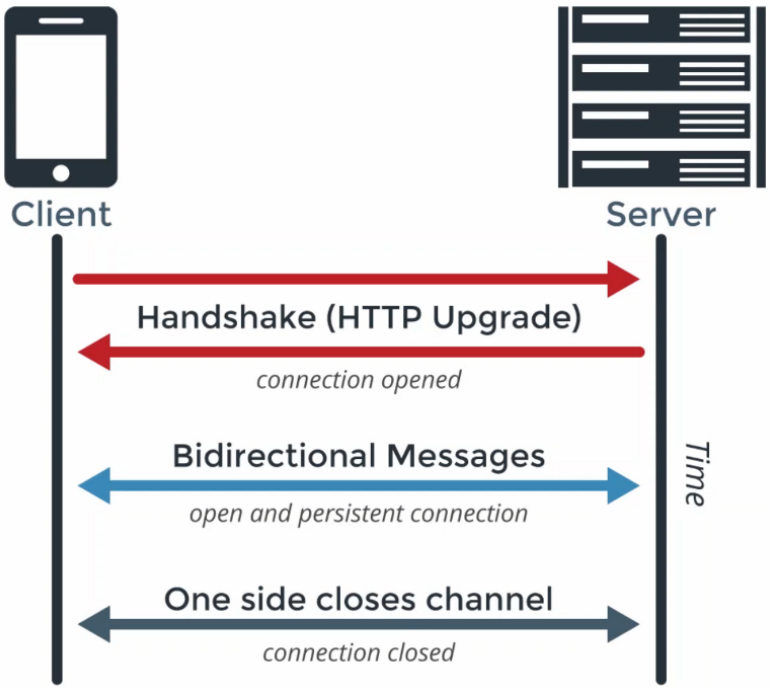
- Побудова взаємодії «сервер-клієнт»
- Реалізація ідемпотентності повідомлень
- Реалізація at-least-once семантики
- Реалізація логіки повторного підключення при розривах з’єднання

Взаємодія між клієнтом та сервером



WebSocket

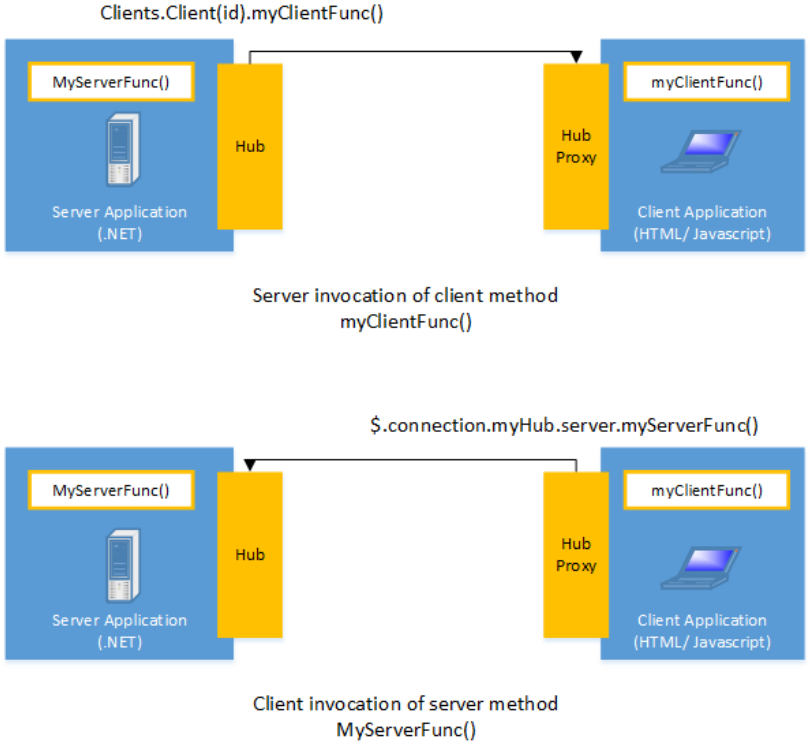
- Двонаправлений повнодуплексний канал зв'язку[1]
- Встановлення з’єднання відбувається через HTTP з вимогою перейти на WebSocket[2]



[1] https://en.wikipedia.org/wiki/WebSocket
[2] https://en.wikipedia.org/wiki/WebSocket#Protocol_handshake

SignalR

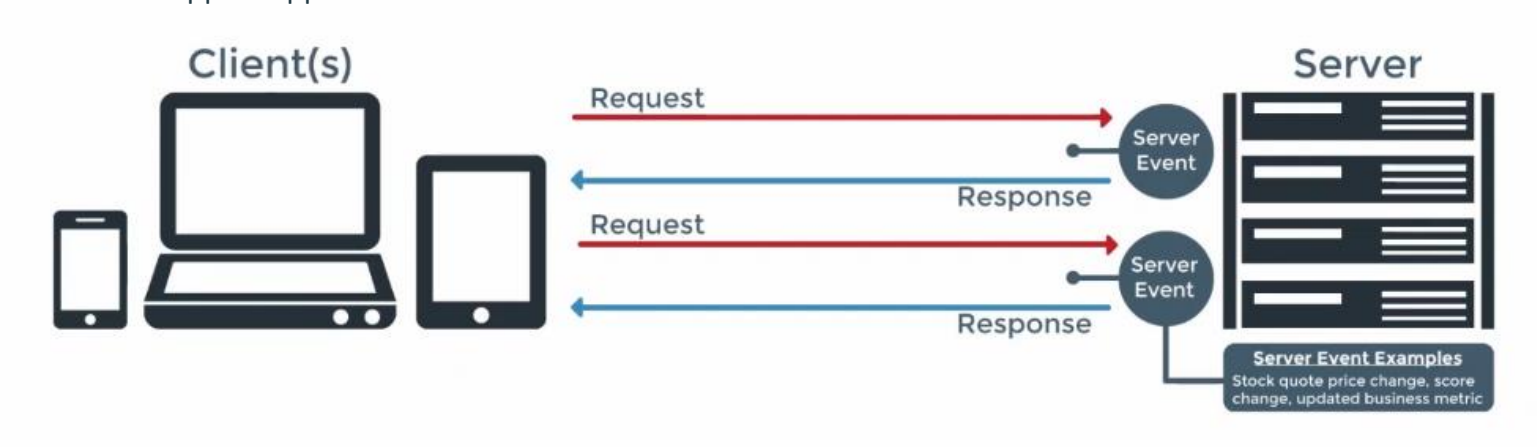
- API для для створення викликів віддалених процедур (RPC)[3]
- Реалізує взаємодію в реальному часі



[3] https://docs.microsoft.com/ru-ru/aspnet/signalr/overview/getting-started/introduction-to-signalr

SignalR Transports

- **WebSocket** — протокол двонаправленого повнодуплексного зв'язку
- **Server Sent Events** — технологія взаємодії «сервер-клієнт» на основі HTTP протоколу
- **Forever Frame** — підхід, який полягає в створені прихованого IFrame на клієнті для отримання та виконання JavaScript коду
- **Long Polling** — підхід, який полягає в опитуванні сервера з певним інтервалом на наявність нових подій за допомогою HTTP запитів



[4] https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr#transports-and-fallbacks

SignalR реалізація на сервері

- Оголошення точки з’єднання

```
public class ChatHub : Hub
{
    public async Task SendServer(string message)
    {
        await this.Clients.All.SendAsync("SendClient", message);
    }
}
```

- Реєстрація точки з’єднання

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapHub<ChatHub>("/chat");
});
```

SignalR реалізація на клієнті

- Встановлення з’єднання

```
const hubConnection = new signalR.HubConnectionBuilder()
    .withUrl("/chat")
    .build();
```

- Обробка повідомлень на клієнті

```
hubConnection.on("SendClient", function (data) {
    console.log("Виклик процедури на клієнті");
});
```

- Виклик процедури на серері

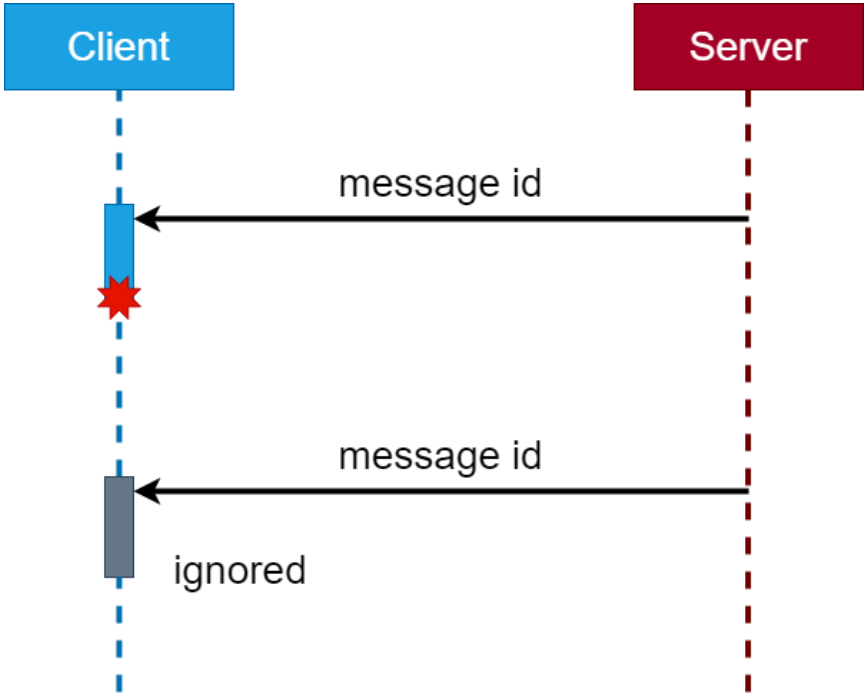
```
let message = "Hello world";

hubConnection.invoke("SendServer", message);
```

Ідемпотентна обробка повідомлень

Ідемпотентність — це властивість, при якій повторна дія над об’єктом не змінює його стану[5]

Рішення:

- Відстежуємо та ігноруємо вже оброблені події
- Підтримка очищення застарілих подій



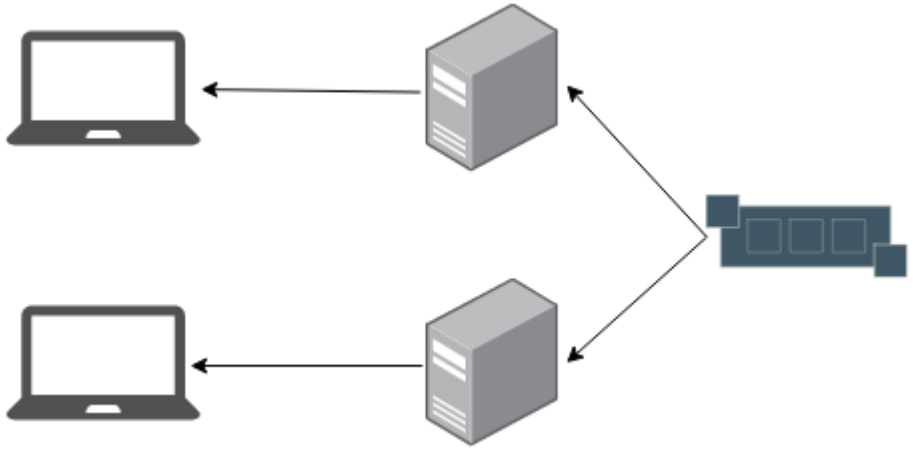
[5] https://en.wikipedia.org/wiki/Idempotence

Реалізація at-least-once семантики

At-least-once семантика — підхід при якому повідомлення буде оброблене клієнтом, як мінімум один раз[6]

Рішення:

- Додати журнал оброблених подій клієнтами
- Рішення для горизонтального масштабування серверів[7]



[6] https://doc.akka.io/docs/akka/current/general/message-delivery-reliability.html?language=scala#the-general-rules
[7] https://docs.microsoft.com/en-us/aspnet/signalr/overview/performance/scaleout-in-signalr

Реалізація логіки повторно підключення при розриві з’єднання

- Періодичність — при невдалій спробі
- Збільшенням інтервалу — знаходження прийнятної швидкості[8]
- Випадковість — щоб уникнути DDoS атаки

[8] https://en.wikipedia.org/wiki/Exponential_backoff

Висновки

- Переваги
  - інтерактивна взаємодія
  - синхронізація даних між клієнтами
- Недоліки
  - складність розробки та відлагодження
  - узгодженість даних
  - погано горизонтально масштабується

Використана література

- [1] WebSocket — Available from: https://en.wikipedia.org/wiki/WebSocket
- [2] WebSocket. Protocol handshake — Available from: https://en.wikipedia.org/wiki/WebSocket#Protocol_handshake
- [3] Introduction to SignalR — Available from: https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr
- [4] Introduction to SignalR. Transports and fallbacks — Available from: https://docs.microsoft.com/en-us/aspnet/signalr/overview/getting-started/introduction-to-signalr#transports-and-fallbacks
- [5] Idempotence — Available from: https://en.wikipedia.org/wiki/Idempotence
- [6] Message-delivery-reliability. The general rules — Available from: https://doc.akka.io/docs/akka/current/general/message-delivery-reliability.html?language=scala#the-general-rules
- [7] Introduction to Scaleout in SignalR — Available from: https://docs.microsoft.com/en-us/aspnet/signalr/overview/performance/scaleout-in-signalr
- [8] Exponential_backoff — Available from: https://en.wikipedia.org/wiki/Exponential_backoff

Дякую за вашу увагу