



Bài 15

Socket IO 1

Module: Web backend development with NodeJS

Mục tiêu



- Trình bày được khái niệm giao thức WebSocket
- Cài đặt được thư viện Socket.io trong Node.js
- Triển khai được thư viện Socket.io
- Triển khai được Socket.io phía server (Node.js)
- So sánh được giao thức WebSockets và HTTP
- Triển khai được Socket.io phía client (JavaScript và HTML)
- Triển khai được kỹ thuật phát sự kiện (emit events)
- Triển khai được kỹ thuật lắng nghe sự kiện



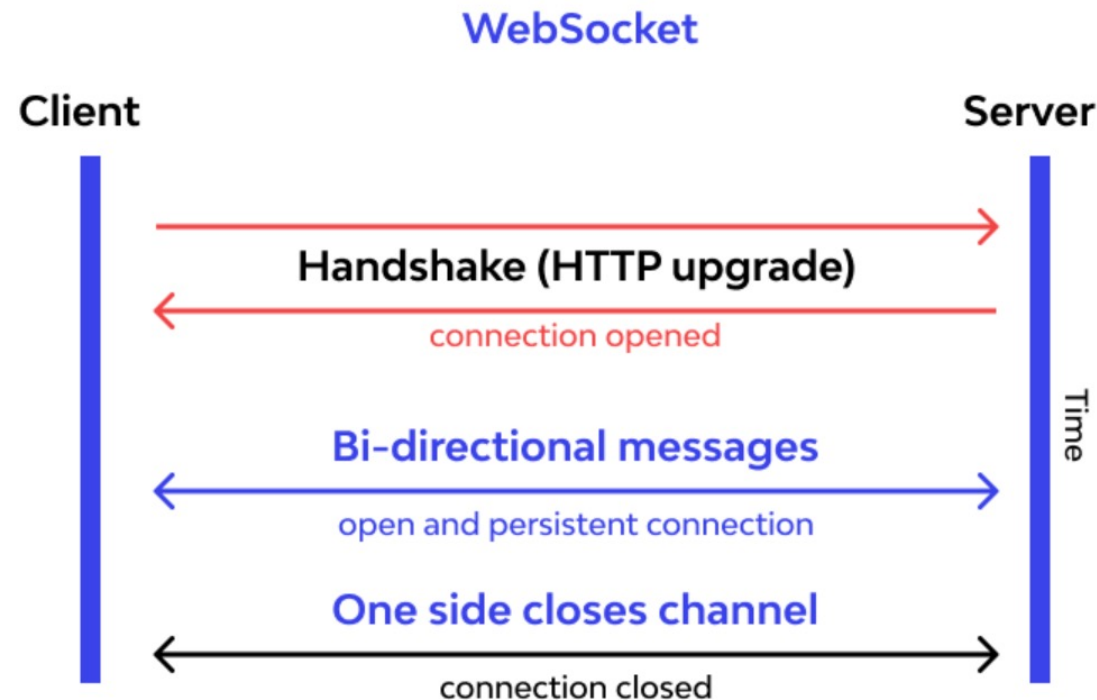
Giao thức WebSocket

Thảo luận

Giao thức WebSocket



- Là giao thức hỗ trợ giao tiếp hai chiều giữa client và server để tạo một kết nối trao đổi dữ liệu
- Hoạt động trên giao thức TCP
- Kết nối được ngắt khi client hoặc server ngắt kết nối



Khi nào nên sử dụng Websocket



- Phát triển ứng dụng web thời gian thực
- Tạo ứng dụng trò chuyện trực tuyến
- Phát triển game online



Thư viện Socket.IO

Thảo luận

- [Socket.io](#) là một thư viện JavaScript cho phép giao tiếp theo thời gian thực, hai Socket.io bao gồm hai API:
- chiều và theo hướng sự kiện giữa client và server..
 - API máy chủ node.js.
 - Một API ứng dụng khách JavaScript cũng có thể được chạy từ node.js.
- Socket.io sử dụng engine.io để thiết lập kết nối và trao đổi dữ liệu giữa client và server.

Ưu điểm socket.io



- Bảo mật cao
- Kết nối tự động tới server
- Cho phép tạo kênh và phòng
- Mã hoá nhị phân

Cài đặt socket.io trên server



Cài đặt thư viện bằng câu lệnh

```
npm install socket.io
```



Cài đặt socket.io trên client

- Mặc định máy chủ socket.io sẽ thêm thư viện cho client tại `/socket.io/socket.io.js`
- Biến `io` sẽ được đăng ký dưới dạng toàn cục

```
<script src="/socket.io/socket.io.js"></script>
<script>
    const socket = io();
</script>
```

- Có thể sử dụng link CDN:

```
<script src="https://cdn.socket.io/4.5.0/socket.io.min.js"></script>
```

Gửi/nhận Event trong Socket.io



- **Socket.io** hoạt động dựa trên các sự kiện.
- Có một số sự kiện dành riêng, có thể được truy cập bằng cách sử dụng đối tượng socket ở phía **server** như: connect, message, disconnect, leave ...
- Bên phía **clients** thì chúng ta có các dạng như: connect, connect_error, ...

Gửi/nhận Event trong socket.io



- API Socket.IO được lấy cảm hứng từ Node.js EventEmitter, có nghĩa là bạn có thể phát ra các sự kiện ở một bên và đăng ký trình nghe ở phía còn lại.

Server

client

```
socket.emit("my-event-a");
```



```
socket.on("my-event-a", () => {  
  // ...  
});
```

```
socket.on("my-event-b", () => {  
  // ...  
});
```



```
socket.emit("my-event-b");
```

Gửi Event



- Để gửi một event chúng ta sử dụng phương thức emit():
- Cú pháp:

```
socket.emit("name-event", data);
```

Nhận Event



- Nhận dữ liệu từ một event đã tạo trước đó bằng cách sử dụng phương thức `on` và truyền vào tham số đầu tiên là tên event và tham số tiếp theo là một *callback function* để bắt data:
- Cú pháp

```
socket.on([eventName], function (from, msg) {  
    //code here  
});
```

Ví dụ



- Phía server gửi event:

```
io.on("connection", (socket) => {  
    socket.emit("hello", "world");  
});
```

- Phía client nhận event:

```
socket.on("hello", (arg) => {  
    console.log(arg); // world  
});
```



Demo cài đặt và sử dụng Socket.io

Thảo luận

So sánh websocket và HTTP



- Giống nhau:
 - Đều là những giao thức hỗ trợ việc truyền tải thông tin giữa máy chủ và máy trạm.
- Khác nhau:
 - WebSocket là giao thức đóng, 2 chiều trong khi HTTP giao thức 1 chiều
 - WebSocket giải quyết độ trễ của HTTP, có thể sử dụng trong các ứng dụng real-time



Tóm tắt bài học

- WebSocket là giao thức hỗ trợ giao tiếp hai chiều giữa client và server để tạo một kết nối trao đổi dữ liệu
- Sử dụng WebSocket có các trường hợp xử lý độ trễ thấp, các ứng dụng chat, game online theo thời gian thực
- Socket.io là thư viện giúp làm việc với socket trở lên dễ dàng hơn

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: Socket IO 2