PHASE 2
WEEK 3

DAY 1

ELBRUS/BOOTCAMP

План

1. WebSocket: что это и зачем нужно?
2. Как реализовано?
3. Практика: чат на WS/Socket.io

WebSocket: что это и зачем нужно?

Socket

Сокет (“разъём”)

Программный интерфейс для обмена сообщениями между процессами.

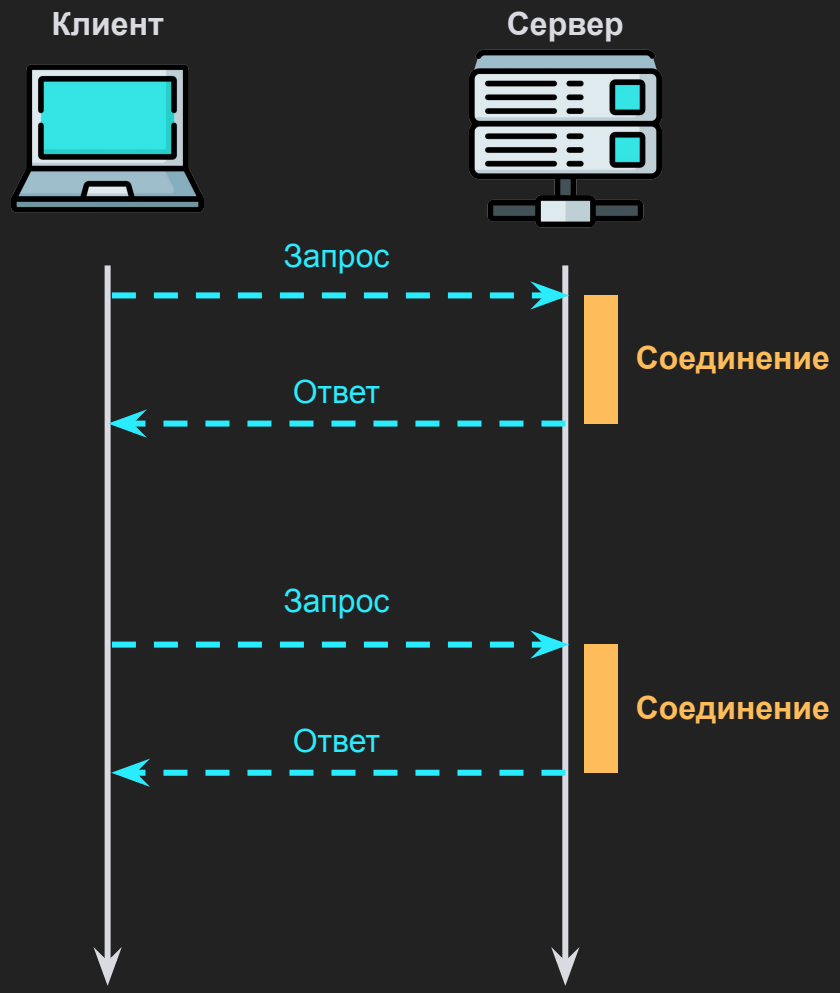
WebSocket: что это и зачем нужно?

WebSocket

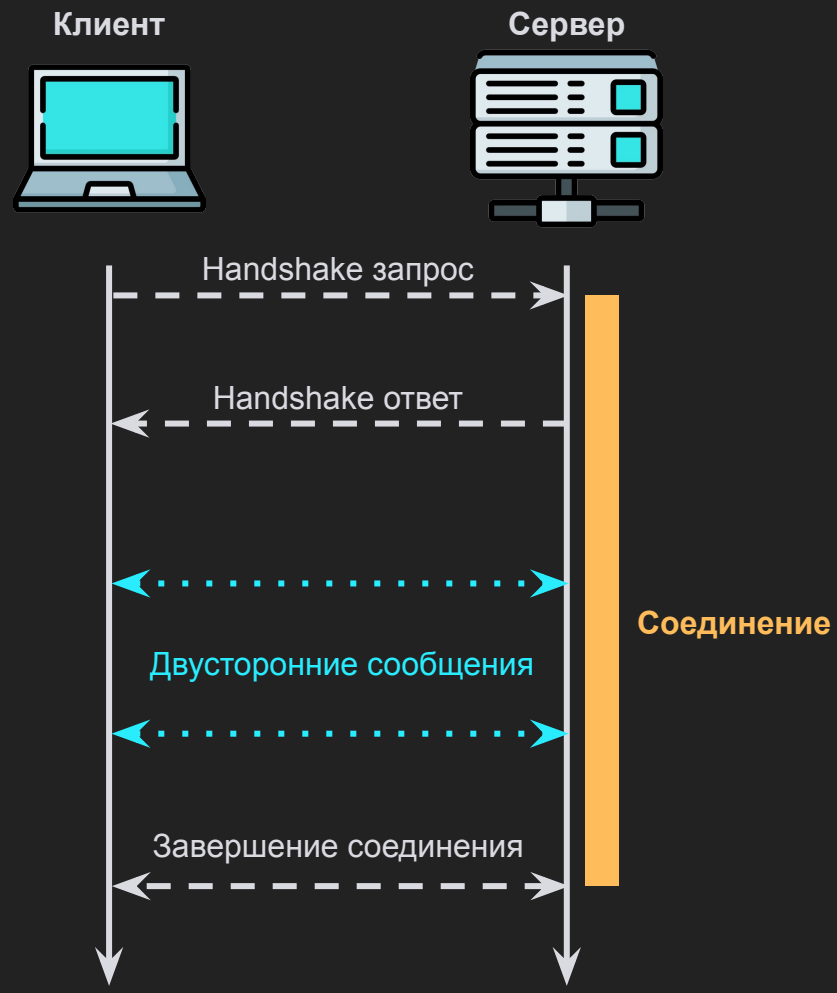
Протокол обмена сообщениями между веб-клиентом и веб-сервером в режиме реального времени.

WebSocket: что это и зачем нужно?

HTTP



WebSocket



WebSocket: что это и зачем нужно?

Варианты применения:

- Чаты
- Онлайн-игры
- Редакторы документов
- Биржи и торговые площадки
- Любые приложения с отслеживанием изменений в реальном времени

WebSocket: как реализовано?

EventEmitter

Класс, доступный из встроенного в Node.js модуля `events`. Позволяет создавать (`emit`) кастомные события.

WebSocket: как реализовано?

Реализации WebSocket API могут быть написаны на разных языках.

Реализации на Node.js используют класс `EventEmitter`.

WebSocket: как реализовано?

Клиентский API:

Инициирует подключение, отправляя handshake-запрос (запрос-рукопожатие)

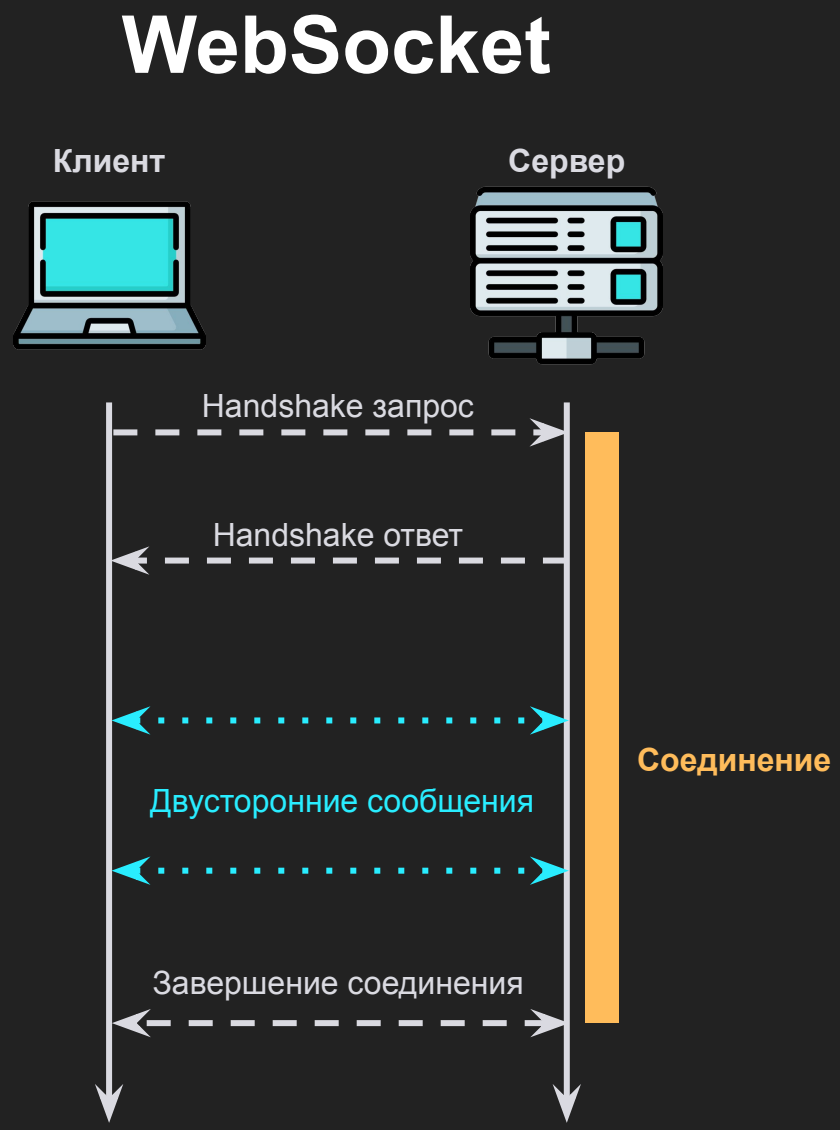
Серверный API:

Слушает событие подключения. Отправляет handshake-ответ или отвергает подключение

WebSocket: как реализовано?

WS-соединение устанавливается **рукопожатием**.

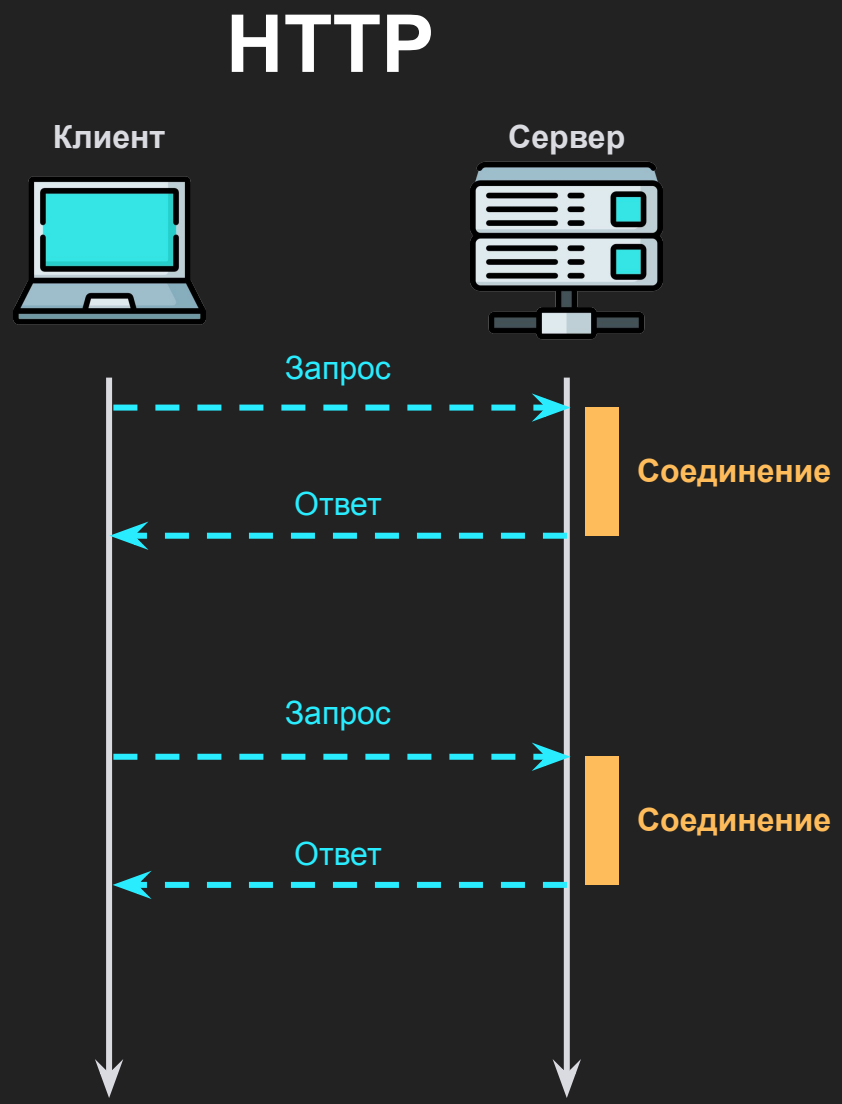
Клиент и сервер могут посылать друг другу кастомные события.



WebSocket: как реализовано?

Данные передаются в обоих направлениях.

Подключение не разрывается, новых HTTP-запросов для обмена не нужно.



WebSocket: как реализовано?

Node.js реализации

- WS library
  https://github.com/websockets/ws
- Socket.io
  https://socket.io/

WebSocket: практика

Socket.io

- Документация по серверу: https://socket.io/docs/v4/server-initialization/
- Документация по клиенту: https://socket.io/docs/v4/client-initialization/

WebSocket: практика

Стек:

- WS или Socket.io
- Express
- Nodemon для разработки