

Задание 1: Реализация чата с WebSocket и модулями

Цель: Создать простое чат-приложение с использованием `WebSocket` и модульной структуры.

Требования:

Сервер:

- Используйте `Node.js` + `Express` + библиотеку `ws`
- Обработывайте события подключения, сообщений и отключения

Клиент:

- Реализуйте `ES6`-модули:
 - `chatUI.js` – обработка `DOM` и событий интерфейса
 - `wsClient.js` – обёртка для `WebSocket`-соединения
- Динамически импортируйте модуль уведомлений (`notifications.js`) при первом сообщении

Функционал:

- Отображение статуса подключения
- История сообщений
- Отправка сообщений через форму

Задание 1: Пример серверной части

```
// server.js
import express from 'express';
import WebSocket from 'ws';

const app = express();
const wss = new WebSocket.Server({ port: 8080 });

wss.on('connection', (ws) => {
  ws.on('message', (message) => {
    wss.clients.forEach(client => client.send(message));
  });
});
```

Задание 2: Модульная система для WebSocket-клиента

Цель: Создать переиспользуемый `WebSocket`-клиент с обработкой ошибок.

Требования:

- Реализуйте класс `WebSocketClient` в отдельном модуле:
- Поддержка повторного подключения при разрыве
- Публичные методы: `connect()`, `send()`, `close()`
- События: `onMessage`, `onError`, `onConnect`

В основном модуле приложения:

- Импортируйте `WebSocketClient` как именованный импорт
- Используйте динамический импорт для модуля `логирования` (`logger.js`) только в `development`-режиме

Дополнительно:

- Реализуйте декоратор для кэширования сообщений
- Настройте Webpack для сборки модулей

Задание 2: Пример

```
// WebSocketClient.js
export class WebSocketClient {
  constructor(url) {
    this.url = url;
    this.reconnectInterval = 3000;
  }

  connect() {
    this.ws = new WebSocket(this.url);
    // ...реализация обработчиков
  }
}
```

Задание 3: дашборд с TypeScript модулями

Цель: Создать дашборд с обновлением данных через `WebSocket` и строгой типизацией.

Требования:

Сервер:

Отправляйте случайные **данные** (температура, влажность) каждые 2 сек

Клиент:

- Разбейте на модули:
 - интерфейсы для данных
 - логика отрисовки графиков
 - работа с `WebSocket`

Функционал:

- Автообновление графиков при новых данных
- Кнопка принудительного разрыва/восстановления соединения