

File Edit View

react_client

EXPLORER

OPEN EDITORS

REACT_CLIENT

- my-coket-client
 - node_modules
 - public
 - src
 - App.css
 - App.js M
 - App.test.js
 - index.css
 - index.js M
 - logo.svg
 - reportWebVitals.js
 - setupTests.js
 - .gitignore
 - package-lock.json M
 - package.json M
 - README.md
 - node_modules

OUTLINE

TIMELINE

JS App.js M

JS index.js M

index.css

my-coket-client > src > JS App.js > [0] App

```

1  import React, { useState, useEffect, useRef } from 'react';
2
3  const App = () => {
4    const [message, setMessage] = useState('');
5    const [name, setName] = useState('');
6    const stompClientRef = useRef(null);
7
8    const fetchGet = async () => {
9      const response = await fetch(
10        'http://localhost:8080/api/messages'
11      );
12      // const data = await response.text();
13      const data = await response.json();
14      console.log(data);
15      setMessage(data);
16    };
17
18    const fetchPost = async () => {
19      const response = await fetch(
20        'http://localhost:8080/api/messages', {
21        method: 'POST',
22        body: JSON.stringify({
23          name
24        })
25      },

```

Ln 7, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

The screenshot shows the VS Code interface with the 'my-coket-client' project open. The Explorer sidebar on the left lists the project files, including 'JS App.js', 'JS App.test.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js', '.gitignore', 'package-lock.json', 'package.json', 'README.md', 'node_modules', and 'package-lock.json'. The main editor area displays the code for 'App.js', which includes a 'fetchPost' function using 'async/await' to send a POST request to 'http://localhost:8080/api/messages'. The code also shows a 'sendMessage' function that calls 'fetchPost'.

```

3  const App = () => {
17
18    const fetchPost = async () => {
19      const response = await fetch(
20        'http://localhost:8080/api/messages', {
21        method: 'POST',
22        body: JSON.stringify({
23          name
24        }),
25        headers: {
26          'Content-type': 'application/json; charset=UTF-8',
27        }
28      });
29    };
30    // const data = await response.text();
31    const data = await response.json();
32    console.log(data);
33    sendMessage(data);
34  };
35  const sendMessage = () => {
36    fetchGet();
37    // fetchPost();
38  };
39

```

Phd Ivan Rovetskyi “Client-Server programming”

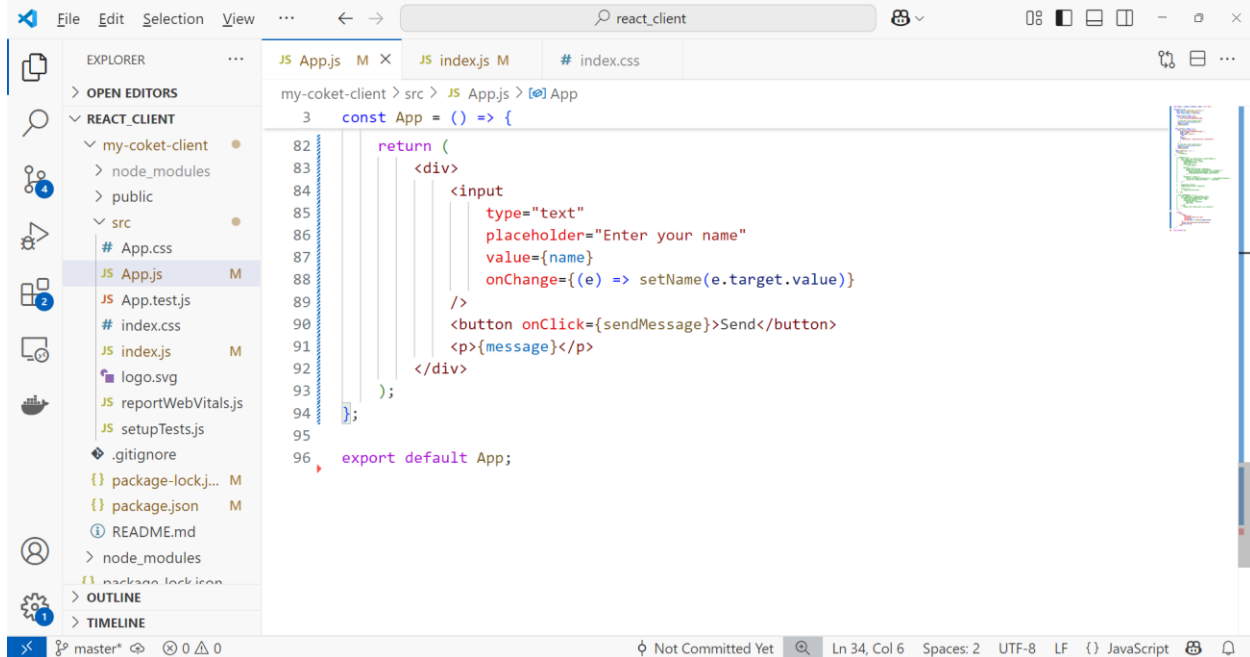
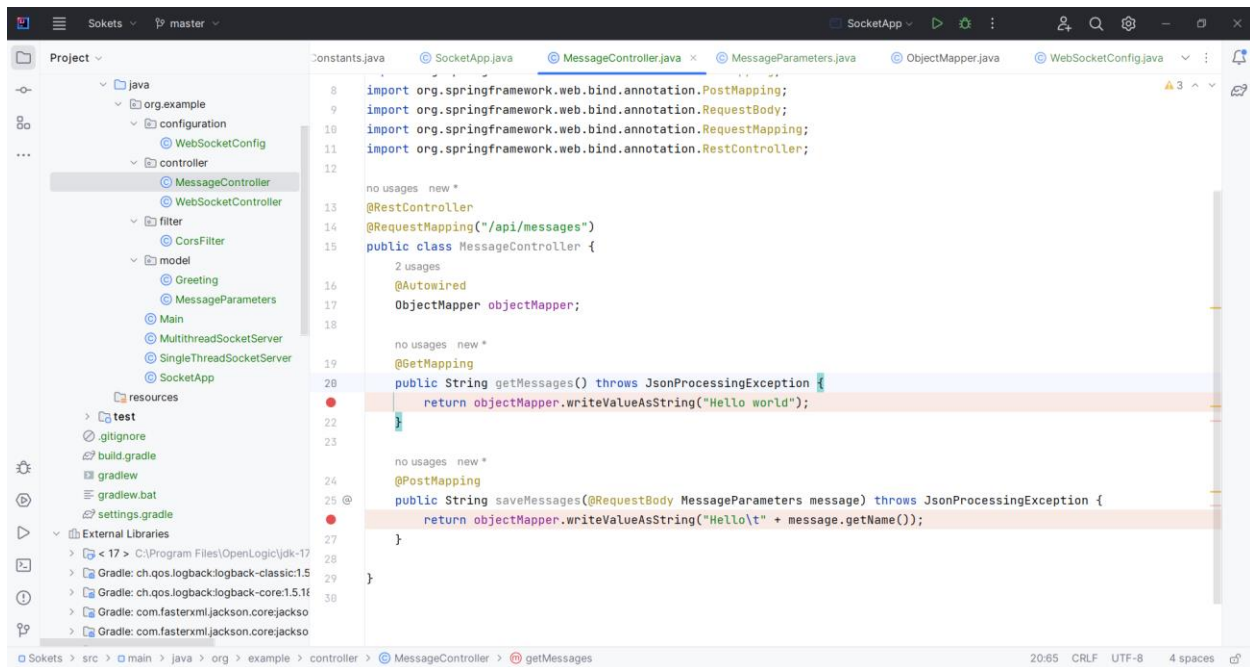


Fig.1. React client for call http-requests

Server part based on Java and Spring Boot.



Phd Ivan Rovetskyi “Client-Server programming”

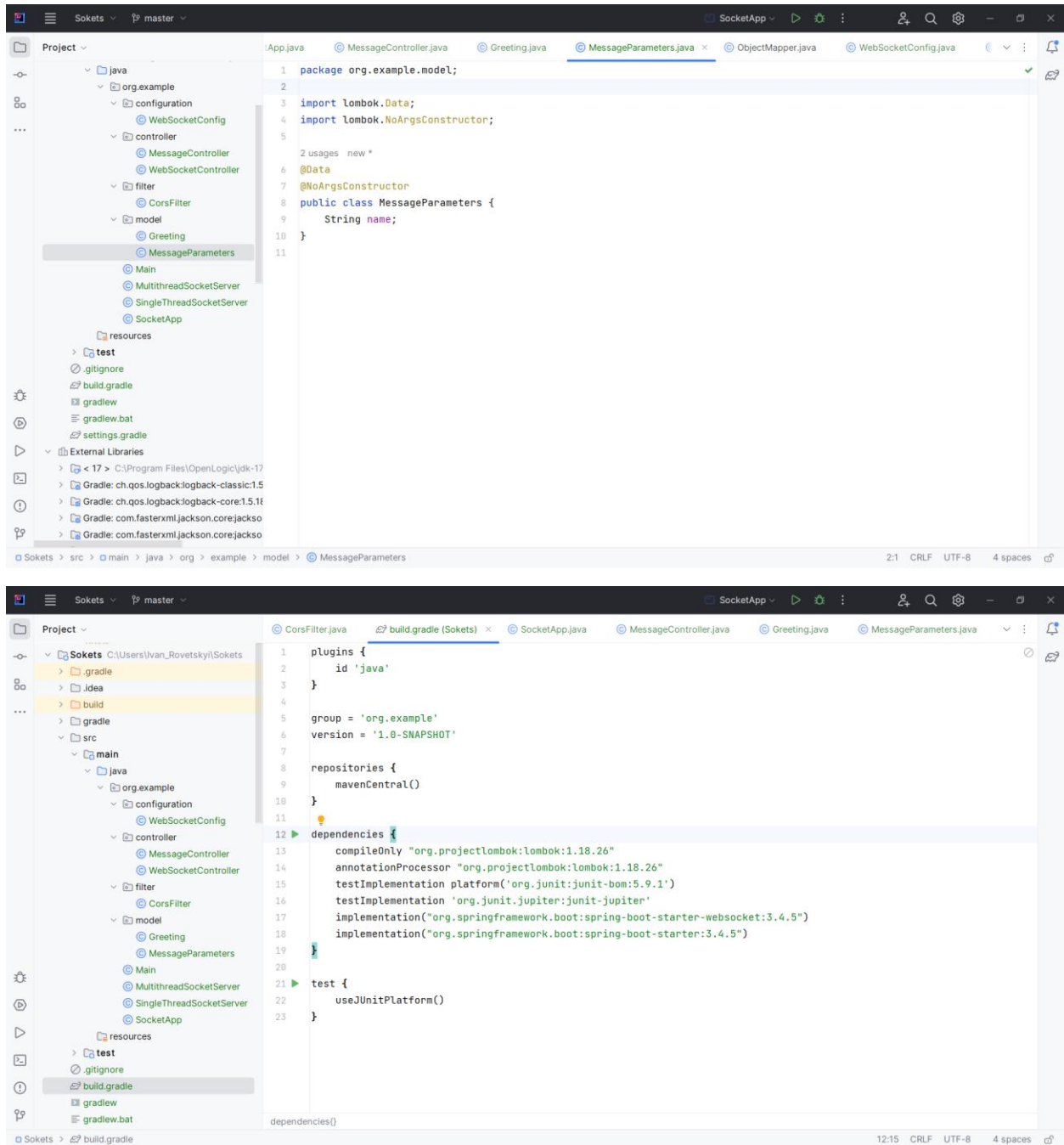


Fig.1. Java server for accept http-requests