

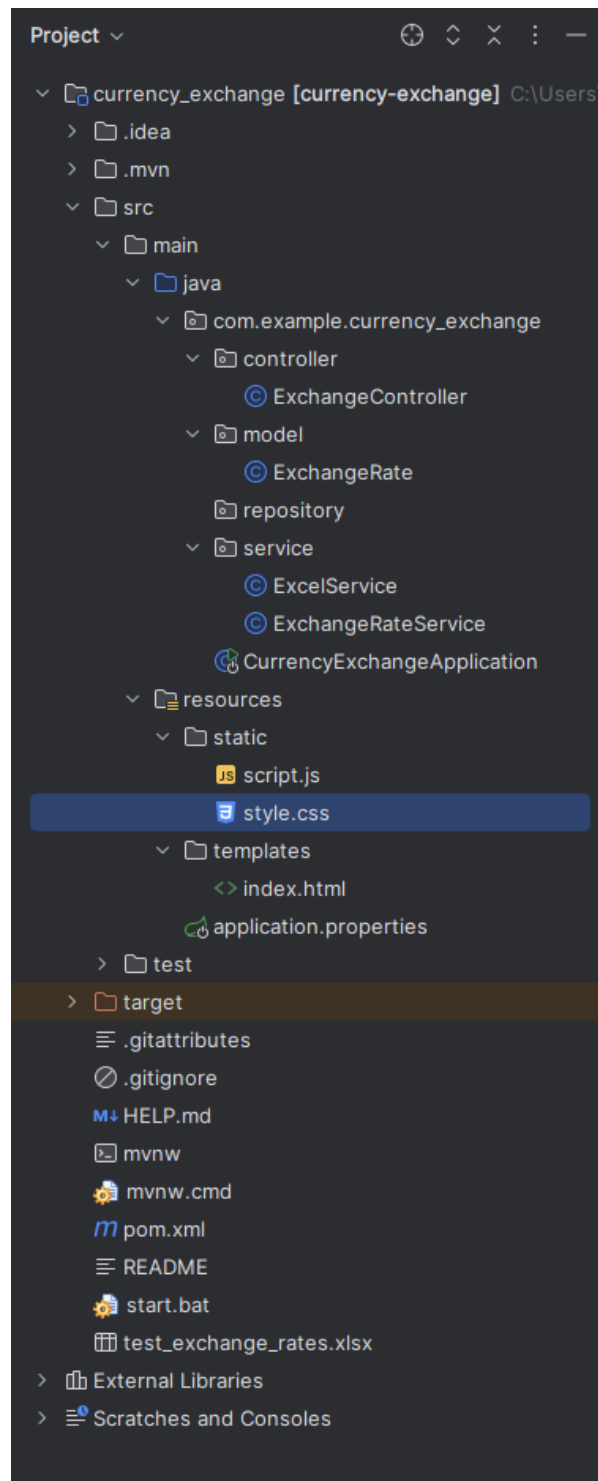
JAVA

Губа Данііл Богданович

124-21-2

Розв'язок

1. Створення програми



ExchangeController

```
package com.example.currency_exchange.controller;

import com.example.currency_exchange.service.ExcelService;
import com.example.currency_exchange.service.ExchangeRateService;
import jakarta.servlet.http.HttpServletResponse;
import org.springframework.web.bind.annotation.*;

import java.io.IOException;
import java.math.BigDecimal;
import java.util.Map;

@RestController
@RequestMapping("/api/exchange")
@CrossOrigin(origins = "*")
public class ExchangeController {
    private final ExchangeRateService exchangeRateService;
    private final ExcelService excelService;

    public ExchangeController(ExchangeRateService
exchangeRateService, ExcelService excelService) {
        this.exchangeRateService = exchangeRateService;
        this.excelService = excelService;
    }

    @GetMapping("/rate")
    public Map<String, BigDecimal> getExchangeRates() {
        return exchangeRateService.getExchangeRates();
    }

    @GetMapping("/convert")
    public BigDecimal convertCurrency(@RequestParam BigDecimal
amount, @RequestParam String currency) {
        Map<String, BigDecimal> rates =
exchangeRateService.getExchangeRates();
        BigDecimal rate = rates.getOrDefault(currency.toUpperCase(),
BigDecimal.ZERO);
        return amount.multiply(rate);
    }

    @GetMapping("/download-excel")
```

```

        public void downloadExcel(HttpServletResponse response,
                                   @RequestParam BigDecimal amount,
                                   @RequestParam String currency) throws
        IOException {
            excelService.exportExchangeRates(response, amount, currency);
        }
    }
}

```

ExchangeRate

```

package com.example.currency_exchange.model;

import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import lombok.Getter;
import lombok.Setter;
import java.math.BigDecimal;

@Getter
@Setter
@JsonIgnoreProperties(ignoreUnknown = true)
public class ExchangeRate {
    private String ccy;
    private String base_ccy;
    private BigDecimal buy;
    private BigDecimal sale;

    public String getCcy() {
        return ccy;
    }

    public BigDecimal getSale() {
        return sale;
    }
}

```

ExcelService

```

package com.example.currency_exchange.service;

import org.apache.poi.ss.usermodel.*;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import org.springframework.stereotype.Service;

import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.math.BigDecimal;
import java.util.Map;

@Service
public class ExcelService {
    private final ExchangeRateService exchangeRateService;

    public ExcelService(ExchangeRateService exchangeRateService) {
        this.exchangeRateService = exchangeRateService;
    }

    public void exportExchangeRates(HttpServletResponse response,
        BigDecimal amount, String currency) throws IOException {
        Workbook workbook = new XSSFWorkbook();
        Sheet sheet = workbook.createSheet("Exchange Rates");

        // Заголовки
        Row headerRow = sheet.createRow(0);
        headerRow.createCell(0).setCellValue("Валюта");
        headerRow.createCell(1).setCellValue("Курс продажи (UAH)");

        // Получаем курсы валют
        Map<String, BigDecimal> rates =
exchangeRateService.getExchangeRates();
        int rowNum = 1;

        for (Map.Entry<String, BigDecimal> entry : rates.entrySet())
        {
            Row row = sheet.createRow(rowNum++);
            row.createCell(0).setCellValue(entry.getKey());

            row.createCell(1).setCellValue(entry.getValue().doubleValue());
        }
    }
}

```

```

        // Добавляем информацию о транзакции
        Row transactionHeaderRow = sheet.createRow(rowNum++);

        transactionHeaderRow.createCell(0).setCellValue("Конвертация");

        Row transactionRow = sheet.createRow(rowNum++);
        transactionRow.createCell(0).setCellValue("Сумма:");

        transactionRow.createCell(1).setCellValue(amount.doubleValue());

        transactionRow = sheet.createRow(rowNum++);
        transactionRow.createCell(0).setCellValue("Валюта:");
        transactionRow.createCell(1).setCellValue(currency);

        // Вычисляем результат конвертации
        BigDecimal rate = rates.getDefault(currency.toUpperCase(),
        BigDecimal.ZERO);
        BigDecimal convertedAmount = amount.multiply(rate);

        transactionRow = sheet.createRow(rowNum++);
        transactionRow.createCell(0).setCellValue("Результат в
        UAH:");

        transactionRow.createCell(1).setCellValue(convertedAmount.doubleValue
        ());

        // Устанавливаем заголовки ответа и отправляем файл
        response.setContentType("application/vnd.openxmlformats-
        officedocument.spreadsheetml.sheet");
        response.setHeader("Content-Disposition", "attachment;
        filename=exchange_rates.xlsx");

        workbook.write(response.getOutputStream());
        workbook.close();
    }
}

```

ExchangeRateService

```
package com.example.currency_exchange.service;

import org.springframework.stereotype.Service;

import java.math.BigDecimal;
import java.util.HashMap;
import java.util.Map;

@Service
public class ExchangeRateService {
    public Map<String, BigDecimal> getExchangeRates() {
        Map<String, BigDecimal> rates = new HashMap<>();
        rates.put("PLN", BigDecimal.valueOf(10.54));
        rates.put("EUR", BigDecimal.valueOf(45.40));
        return rates;
    }
}
```

Index.html

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Обмін валют</title>
    <link rel="stylesheet" href="/style.css">
</head>
<body>
<div class="container">
    <h1>Обмін валют</h1>

    <p>Курс злотих: <span id="pln-rate">...</span></p>
    <p>Курс євро: <span id="eur-rate">...</span></p>

    <h2>Конвертація</h2>
    <input type="number" id="amount" placeholder="Сума">
    <select id="currency">
        <option value="PLN">PLN</option>
```

```

        <option value="EUR">EUR</option>
    </select>
    <button onclick="convertCurrency()">Конвертувати</button>

    <h3>Результат: <span id="result">...</span></h3>
    <br>
    <button onclick="downloadExcel()">Скачати звіт</button>
</div>

<script src="/script.js"></script>
</body>
</html>

```

Style.css

```

body {
    margin: 0;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background: linear-gradient(to right, #e0eafc, #cfdef3);
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
}

.container {
    background: white;
    padding: 30px;
    border-radius: 16px;
    box-shadow: 0 8px 20px rgba(0, 0, 0, 0.1);
    max-width: 400px;
    width: 90%;
    text-align: center;
}

h1, h2, h3 {
    color: #2c3e50;
    margin-bottom: 15px;
}

p {

```

```
        font-size: 1.1em;
        color: #34495e;
        margin: 10px 0;
    }

    input[type="number"],
    select {
        padding: 10px;
        font-size: 1em;
        width: 100%;
        margin-bottom: 15px;
        border: 1px solid #ccc;
        border-radius: 8px;
        box-sizing: border-box;
        transition: border-color 0.3s;
    }

    input[type="number"]:focus,
    select:focus {
        border-color: #3498db;
        outline: none;
    }

    button {
        padding: 10px 20px;
        background-color: #3498db;
        color: white;
        font-size: 1em;
        border: none;
        border-radius: 8px;
        cursor: pointer;
        transition: background-color 0.3s, transform 0.2s;
        width: 100%;
        margin-top: 10px;
    }

    button:hover {
        background-color: #2980b9;
        transform: scale(1.02);
    }
```



```
#result {  
    font-weight: bold;  
    color: #2ecc71;  
}
```

Script.js

```
const apiUrl = "http://localhost:8080/api/exchange";  
  
async function fetchRates() {  
    try {  
        const response = await fetch(`${apiUrl}/rate`);  
        if (!response.ok) throw new Error("Ошибка запроса");  
  
        const data = await response.json();  
  
        document.getElementById("pln-rate").textContent = data.PLN +  
        " грн";  
        document.getElementById("eur-rate").textContent = data.EUR +  
        " грн";  
    } catch (error) {  
        document.getElementById("pln-rate").textContent = "Ошибка";  
        document.getElementById("eur-rate").textContent = "Ошибка";  
    }  
}  
  
async function convertCurrency() {  
    const amount = document.getElementById("amount").value;  
    const currency = document.getElementById("currency").value;  
  
    try {  
        const response = await  
fetch(`${apiUrl}/convert?amount=${amount}&currency=${currency}`);  
        if (!response.ok) throw new Error("Ошибка конвертации");  
  
        const convertedAmount = await response.json();  
        document.getElementById("result").textContent =  
convertedAmount + " грн";  
    } catch (error) {  
        document.getElementById("result").textContent = "Ошибка";  
    }  
}
```

```

    }

    async function downloadExcel() {
        const amount = document.getElementById("amount").value;
        const currency = document.getElementById("currency").value;

        if (!amount) {
            alert("Введіть суму для конвертації!");
            return;
        }

        const url = `http://localhost:8080/api/exchange/download-excel?amount=${amount}&currency=${currency}`;

        fetch(url)
            .then(response => response.blob())
            .then(blob => {
                const link = document.createElement("a");
                link.href = window.URL.createObjectURL(blob);
                link.download = "transaction.xlsx";
                document.body.appendChild(link);
                link.click();
                document.body.removeChild(link);
            })
            .catch(error => console.error("Помилка скачування:", error));
    }

    fetchRates();

```

Start.bat

```

@echo off
cd /d %~dp0
echo Cleaning and running Spring Boot application...
mvn clean spring-boot:run

echo.
echo Press any key to exit...
pause

```

2. Запуск програми

Запускаємо файл Start.bat

```
[INFO] Compiling 1 source file with javac [debug parameters release 17] to target\test-classes
[INFO]
[INFO] <<< spring-boot:3.4.3:run (default-cli) < test-compile @ currency-exchange <<<
[INFO]
[INFO] --- spring-boot:3.4.3:run (default-cli) @ currency-exchange ---
[INFO] Attaching agents: []

  ____ _
 / ___ \| | | |
| |   \| |_| |
 \___/\__|_____|

:: Spring Boot ::                (v3.4.3)
```

Запускаємо браузер та вводимо <http://localhost:8080/>

3. Результат виконання

Обмін валют

Курс злотих: 10.54 грн
Курс євро: 45.4 грн

Конвертація

Конвертувати

Результат: **200.26 грн**

Скачати звіт

Обмін валют PLN

Обмін валют

Курс злотих: 10.54 грн
Курс євро: 45.4 грн

Конвертація

Конвертувати

Результат: 200.26 грн

Скачати звіт

Обмін валют EUR

Обмін валют

Курс злотих: 10.54 грн
Курс євро: 45.4 грн

Конвертація

Конвертувати

Результат: 862.6 грн

Скачати звіт

Клікаємо на кнопку [Скачати звіт](#)

Завантажуємо файл Excel та відкриваємо

	A	B	C	D
1	Валюта	Курс продажу (UAH)		
2	PLN	10,54		
3	EUR	45,4		
4	Конвертація			
5	Сумма:	19		
6	Валюта:	PLN		
7	Результат	200,26		
8				

Бачимо наступне

1. Курс валют (Який беремо з API <<Приват Банк>>)
2. Сума конвертації
3. Результат конвертації в ГРН. (Останній запит)