

Лабораторна робота №2. Протоколи IoT

Частина 2. Використання WEB API та Web-сокетів

Мета: Навчитись використовувати протоколи WEB-API для побудови розподілених застосунків в Інтернет.

Цілі:

- 1) ознайомитися з роботою HTTP та використанням інструментів Веб-розробника ;
- 2) реалізувати клієнта HTTP в Node-RED
- 3) використати один з відкритих WEB-API, зокрема REST-API в застосунку Node-RED
- 4) використати платформу FRED - хмарний сервіс Node-RED
- 5) використати WEB-сокети в одному з прикладів
- 6) використати WEB-сокети для з'єднання мобільного застосунку та Node-RED
- 7) розробити застосунок в Node-RED для фіксації штрих-кодів в базі даних та виведення по ним інформації на веб-інтерфейс з використанням технологій WEB API

Більшість ресурсів є захищеними та потребують автентифікації, шифрування і т.п. Темі кібер-захисту в IIoT буде присвячена окрема лабораторна робота. У цій роботі усі інтерфейси є відкритими, тому не можуть в чистому вигляді використовуватися в промислових умовах!

2.1. Знайомство з роботою HTTP та використанням інструментів Веб-розробника

Ознайомтеся з принципами роботи HTTP(даються в лекційному матеріалі).

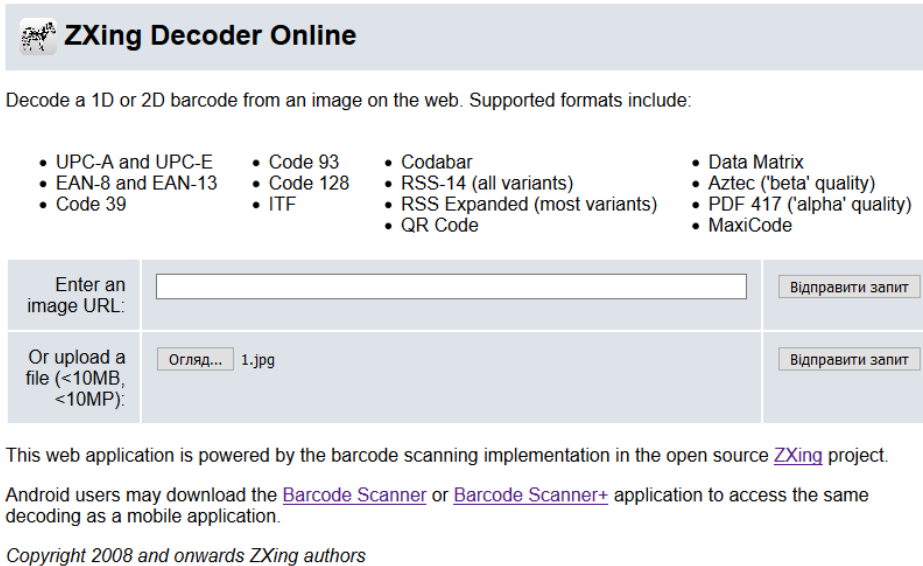
Для перевірки роботи протоколу HTTP, відлагодження та тестування HTML, CSS і JavaScript в завантажених сторінках використовуються різноманітні інструменти Веб-розробника (WEB Developer Tools). Деякі з них є частиною браузерів, таких як Google Chrome або Mozilla FireFox. Якщо Ви звикли працювати з Google Chrome, ознайомитися з вкладками і можливостями Ви можете за наступним [посиланням](#). В даній лабораторній роботі використовується інструменти WEB Developer Tools, що вбудовані в Mozilla FireFox. Зокрема, для аналізу і тестування HTTP запитів використовується Монітор мережі, документація українською мовою для якого доступна за [посиланням](#). У прикладах в лабораторній роботі використовується українська версія FireFox, яку можна завантажити за наступним посиланням <https://www.mozilla.org/uk/firefox/download/thanks/>. При виконанні лабораторної роботи дозволяється використовувати і інші інструменти.

Для перевірки роботи запитів HTTP використовуватимуться сервіси сайту <https://zxing.org/w/decode.aspx>. Ці сервіси дають можливість декодувати штрих-коди та QR-коди що передаються на сайт у вигляді зображення. Слід відмітити, що є багато програм, які можуть це робити в онлайн, дана використовується тільки в якості прикладу.

1. Перевірка роботи WEB-застосунку <https://zxing.org>

Відкрийте сторінку в браузері <https://zxing.org/w/decode.aspx> і перевірте її роботу. Для цього завантажте будь-який файл з зображенням штрих-коду, наприклад [звідси](#) (див. рис.1).

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net



ZXing Decoder Online

Decode a 1D or 2D barcode from an image on the web. Supported formats include:

- UPC-A and UPC-E
- EAN-8 and EAN-13
- Code 39
- Code 93
- Code 128
- ITF
- Codabar
- RSS-14 (all variants)
- RSS Expanded (most variants)
- QR Code
- Data Matrix
- Aztec ('beta' quality)
- PDF 417 ('alpha' quality)
- MaxiCode

Enter an image URL:

Or upload a file (<10MB, <10MP): 1.jpg

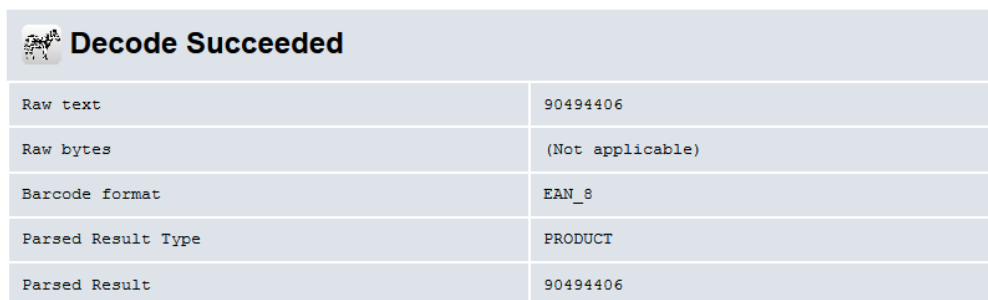
This web application is powered by the barcode scanning implementation in the open source [ZXing](#) project.

Android users may download the [Barcode Scanner](#) or [Barcode Scanner+](#) application to access the same decoding as a mobile application.

Copyright 2008 and onwards ZXing authors

рис.1

Натисніть «Відправити запит», при вдалій обробці запиту буде повернений результат з кодом, подібно до наведеного на рис.2.



Decode Succeeded	
Raw text	90494406
Raw bytes	(Not applicable)
Barcode format	EAN_8
Parsed Result Type	PRODUCT
Parsed Result	90494406

рис.2

2. Робота з Монітором мережі FireFox.

Активуйте в браузері FireFox інструмент «Монітор мережі», або аналогічний в інших браузерах. Робота з Монітором мережі описана за [посиланням](#). Виставте опцію «Вимкнути кеш» з панелі інструментів Монітору мережі. Відкрийте в браузері сторінку <https://zxing.org>. Подивіться список мережних запитів, які були зроблені при завантаженні сторінки, зокрема зверніть увагу на наступні значення:

- скільки запитів було зроблено
- які причини та типи запитів
- які були повернуті результати (стани), скористуйтеся [даним ресурсом Вікіпедії](#) для визначення стану

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

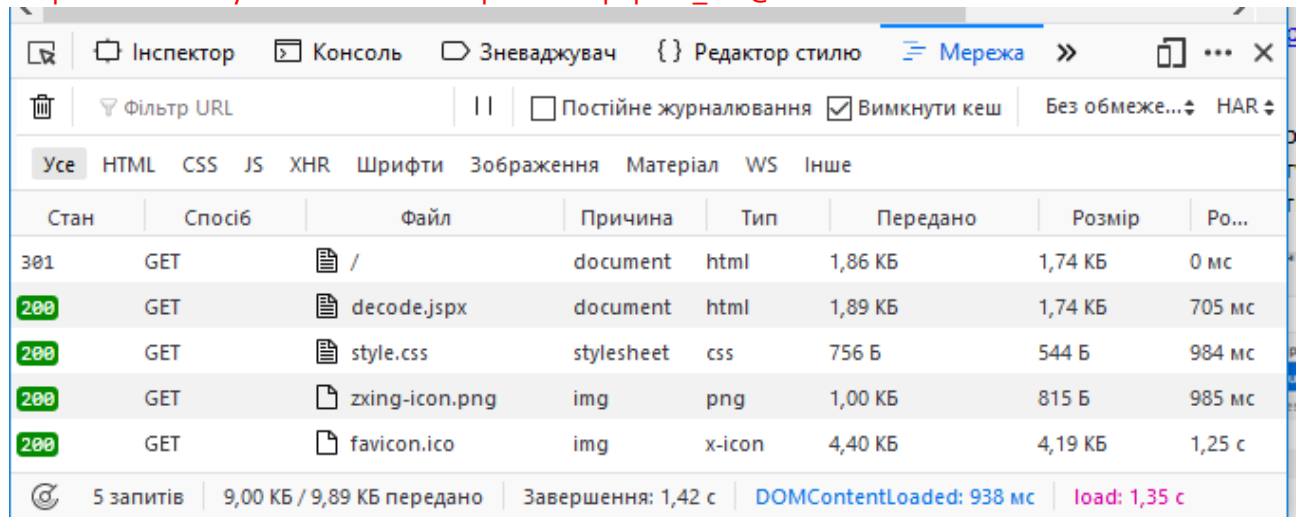


рис.3

3. Аналіз заголовків 1-го мережного запиту та відповіді.

Активуйте панель [Подробиць мережних запитів](#), відкрийте панель «Заголовки» для першого запиту в списку. Ознайомтеся з призначенням використаних в запиті заголовків з ресурсів Інтернет, наприклад з [Вікіпедії](#), сторінкою із [MDN](#) або іншими аналогічними.

Заповніть перші два поля таблиці, наведеної нижче, для пояснення заголовків в запиті. Для зручності розшифровки можете скористатися посиланнями в колонці Заголовок. Інші поля можете розібрати за бажанням.

Таб.1. Пояснення значень заголовків запиту

Заголовок	Значення	Пояснення значень заголовків
Accept-Encoding	gzip, deflate, br	
Accept-Language	uk-UA,uk;q=0.8,en-US;q=0.5,en;q=0.3	
Cache-Control	no-cache	
Connection	keep-alive	
Host	zxing.org	
Pragma	no-cache	
Upgrade-Insecure-Requests	1	
User-Agent	Mozilla/5.0 (Windows NT 10.0; ...) Gecko/20100101 Firefox/64.0	

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

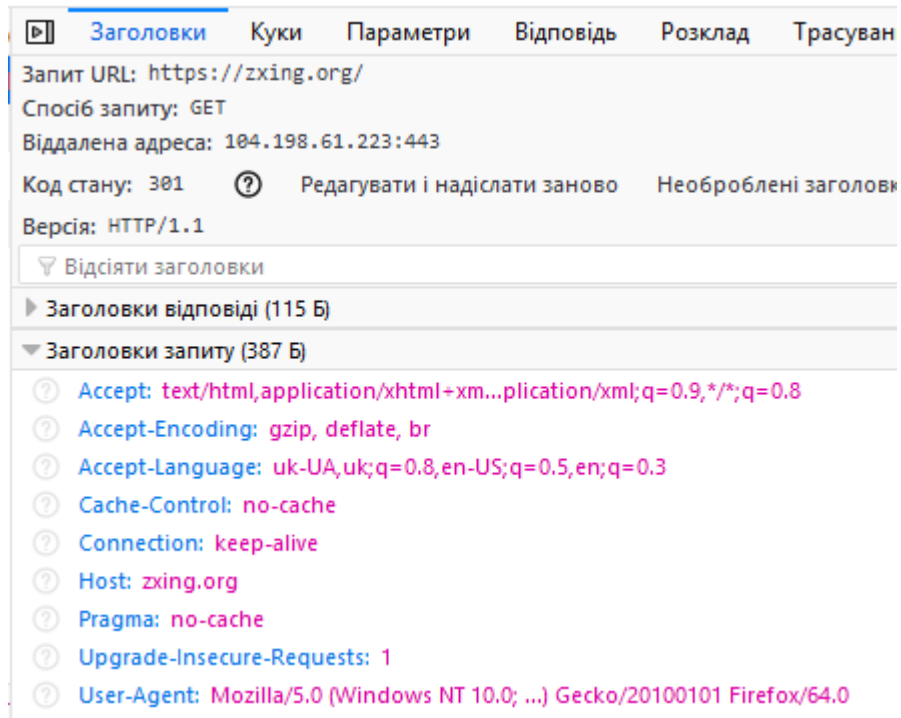


рис.4

Заповніть поля таблиці для заголовку Location для пояснення **відповіді** . Для зручності розшифровки можете скористатися посиланнями в колонці Заголовков.

Таб.2. Пояснення значень заголовків відповіді

Заголовок	Значення	Пояснення значень заголовків
Location	/w/decode.jsx	

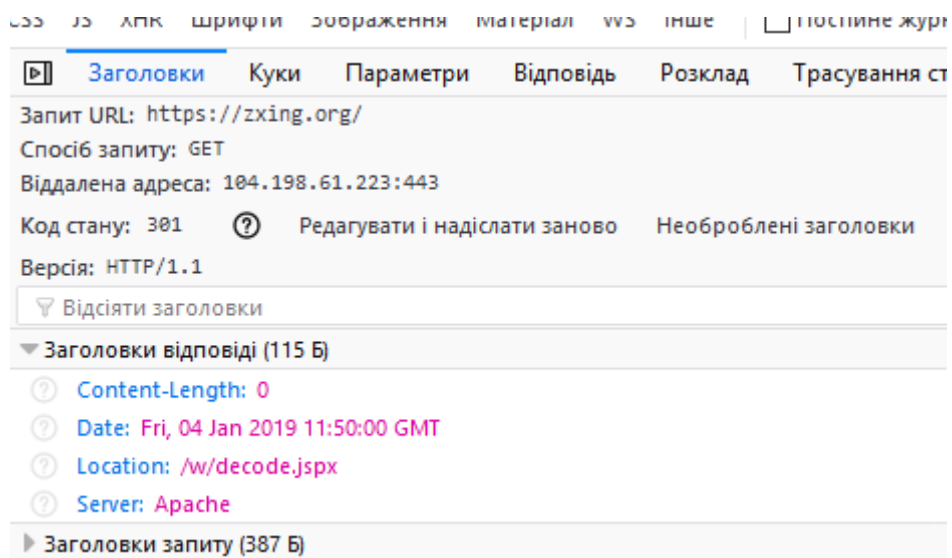


рис.5.

Перегляньте вміст відповіді у вкладці «Відповідь» на перший запит.

Зробіть висновок щодо результату запиту. Впишіть результати в перший рядок таблиці 3.

Таб.3. Пояснення до запитів та відповідей

№	Запит (метод, URL)	Пояснення до запиту, ініціатор	Відповідь (стан – пояснення)	Пояснення до відповіді
1	GET https://zxing.org/	Запит на документ, ініціював	301 - ???	

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

		користувач в рядку браузеру		
2				
3				
4				
5				

4. Аналіз заголовків 2-го мережного запиту та відповіді.

Проаналізуйте другий запит та відповідь на нього. Зверніть увагу на зовнішній вигляд у вікні «Перегляд» у порівнянні з виглядом завантаженої сторінки. Чим вони відрізняються?

Відкрийте корисне навантаження відповіді на панелі «Відповідь» (у форматі HTML). Для того, щоб зручніше проаналізувати вміст завантаженої сторінки, скопіюйте його, відкрийте на іншій закладці браузера посилання <https://www.freeformatter.com/html-formatter.html> та помістіть скопійований текст в поле «Option 1: Copy-paste your HTML document here» після чого натисніть на кнопку «format HTML». У вікні «Formatted HTML:» з'явиться відформатований код сторінки в форматі HTML.

Проаналізуйте HTML код. Якщо ви до цього не знайомилися з мовою розмітки HTML - можете прочитати [цей урок](#).

Зверніть увагу на тег **link** в заголовку HTML (допомога по призначенню тегу [тут](#)), та **img** (допомога по призначенню тегу [тут](#)).

Заповніть 2-й рядок таблиці 3.

5. Аналіз заголовків 3,4 мережного запиту та відповіді.

Проаналізуйте 3-й та 4-й запити та відповіді. За якої причини з'явилися ці запити? Заповніть 3-й та 4-й рядок таблиці 3.

6. Аналіз заголовків 5 мережного запиту та відповіді.

Для розуміння 5-го запиту, прочитайте цю статтю <https://uk.wikipedia.org/wiki/Favicon> . Заповніть 5-й рядок таблиці 3.

7. Аналіз форми в HTML .

Проаналізуйте частину HTML що відповідає за 2-гу форму (отримання коду по зображенню, що відправляється). Про форми в HTML та тег **form**, можна прочитати [за цим посиланням](#).

8. Аналіз заголовків та змісту POST.

Використовуючи кнопку «Очистити» що знаходиться на панелі інструментів Монітору мережі, очистіть список запитів. Завантажте будь-який файл з зображенням штрих-коду, як робили це в п.1. Натисніть «Відправити запит». При вдалій обробці запиту буде повернений результат з кодом.

Використовуючи подробиці мережного запиту **POST** продивіться та проаналізуйте Заголовки. Зробіть копію екранів з заголовками та впишіть пояснення в таблиці 4. Для пришвидшення пошуку призначення заголовків користуйтеся посиланнями, що прив'язані до назв заголовків. Про типи MIME можна прочитати [тут](#).

Таб.4. Пояснення значень заголовків запиту

Заголовок	Значення	Пояснення значень заголовків
Content-Length		
Content-Type		

Використовуючи подробиці мережного запиту POST продивіться та проаналізуйте Параметри. Зробіть копію екрану корисного навантаження запиту.

9. Аналіз відповіді на POST

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

Відкрийте корисне навантаження відповіді на панелі «Відповідь» (у форматі HTML). Для того, щоб зручніше проаналізувати вміст завантаженої сторінки, скопіюйте його, відкрийте на іншій вкладці браузеру посилання <https://www.freeformatter.com/html-formatter.html> та помістіть скопійований текст в поле «Option 1: Copy-paste your HTML document here» після чого натисніть на кнопку «format HTML». У вікні «Formatted HTML:» з'явиться відформатований код сторінки в форматі HTML. Знайдіть місце в документі HTML, де виводиться значення штрих-коду. Зробіть копію екрану і виділіть це місце.

2.2. Реалізація клієнта HTTP в Node-RED

В даній частині лабораторної роботи необхідно в Node-RED розробити програму для визначення штрих-коду по зображенню, використовуючи сервіси <https://zxing.org>. Для розуміння цієї частини лабораторної знадобляться результати, отримані в частині 2.1.

1. Підготовчі роботи.

На диску C:\ створіть директорію «Temp», якщо вона ще не існує. У цій директорії створіть директорію «barcodes», в якій будуть розміщуватися усі необхідні файли для лабораторної роботи. В директорії C:\Temp\barcodes створіть директорію «raw» для розміщення вихідних файлів зображень. Скопіюйте в папку «C:\Temp\barcodes» та «C:\Temp\barcodes\raw» файл з зображенням штрих-коду, що використовувався в попередній частині, наприклад взятий [звідси](#).

2. Читання файлу.

Завантажте Node-RED.

Зробіть статус усіх потоків неактивним.

Створіть новий потік з назвою «Laba2_WEBAPI». Цей потік буде використовуватися для даної частини лабораторної роботи.

З [довідника](#) ознайомтеся з вбудованими функціями роботи з файлами. З розділу «Storage» палітри вузлів вставте вузол «File in». Налаштуйте його таким чином, щоб він зчитував дані з файлу з штрих-кодом та виводив його вміст на панель відлагодження (Debug). Зверніть увагу, що виведення в Debug треба робити як «complete msg object» а не як «msg.payload».

Зробіть розгортання потоку, та з використанням панелі відлагодження подивіться що зміст файлу виводиться як масив байт (рис.6).

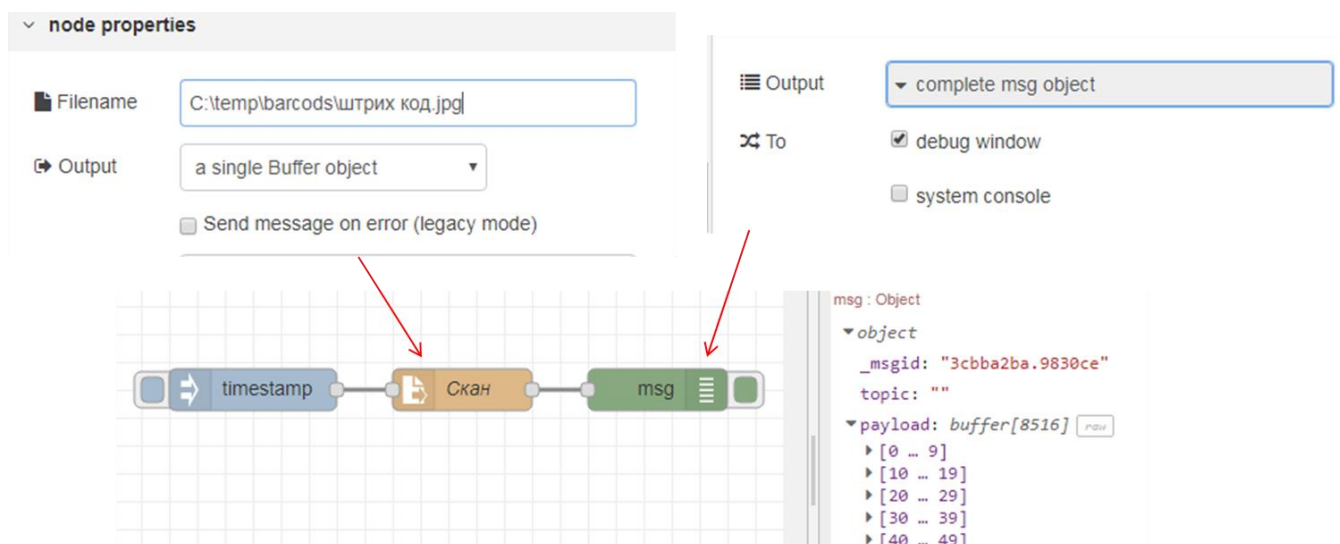


рис.6.

3. Формування запиту.

Для роботи в Node-RED HTTP-клієнту використовується вузол «HTTP request». Ознайомтеся з його роботою з [довідника](#).

Для формування запиту необхідно сформулювати на вході вузла «HTTP request» його заголовки та тіло. З попередньої частини лабораторної роботи видно, що для отримання штрих-коду по зображенню необхідно сформулювати запит з методом POST, який буде багато-частинним, тобто

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

складатися з розділів, що відокремлюються межами. У даному випадку розділ є тільки один, який включає вміст файлу зображення. Для пришвидшення виконання лабораторної роботи імпортуйте в потік підготовлений код для вузлу «Формувати запит», файл імпорту завантажить [звідси](#). Уважно ознайомтеся з його вмістом.

Модифікуйте програму, як це показано на рис.7. Зробіть розгортання, активуйте ініціювання повідомлення в Inject і подивіться на вміст об'єкту на панелі відлагодження. Порівняйте результат з заголовками та вмістом запиту POST з п.1.8 попередньої частини («Знайомство з роботою HTTP»). Що відрізняється і чому? Результат порівняння і пояснення впишіть в звіт до лабораторної роботи.

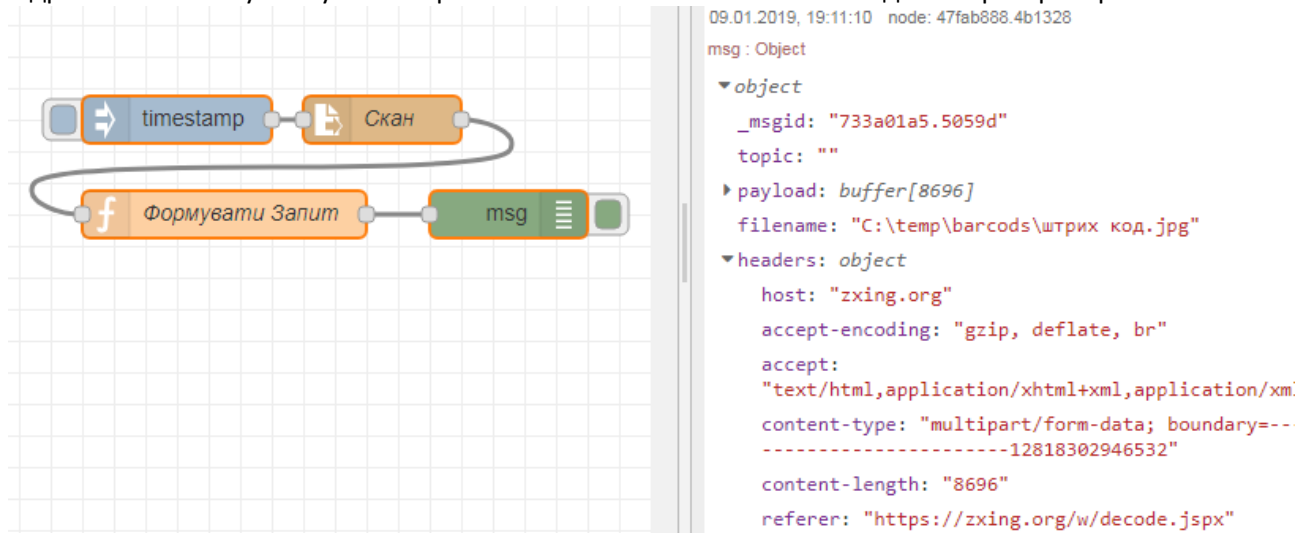


рис.7.

4. Відправлення запиту.

З розділу «function» палітри вузлів вставте вузол «HTTP request». Модифікуйте програму, як це показано на рис.8. Зверніть увагу, що для кращого тестування бажано вивести як вхідне так і вихідне повідомлення «HTTP request».

Зробіть розгортання, активуйте Inject, через кілька секунд на панелі відлагодження з'явиться інформація про об'єкт-відповідь. Порівняйте результат з заголовками та вмістом відповіді POST з п.9 попередньої частини («Знайомство з роботою HTTP»). Результат виконання вважається позитивним, якщо у відповіді буде напис «Decode Succeeded». Знайдіть значення штрих-коду.

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

The screenshot shows a Node-RED workflow on the left and a decoded payload on the right. The workflow consists of the following nodes: 'timestamp', 'Скан' (Scan), 'Формувати Запит' (Formulate Request), 'http request', and 'msg'. A red arrow points from the 'http request' node to the decoded payload. The payload is an HTML document with a table containing barcode information.

```
payload: "<!DOCTYPE html><html><head><meta charset='UTF-8'><meta content='none' name='robots'><title>Decode Succeeded</title><link type='text/css' href='style.css' rel='stylesheet'><style type='text/css'>+ td { vertical-align: top; padding: 0.1in; background-color: #EEEEEE; font-family: monospace; }+ </style></head><body><div id='main'><div id='header'><h1><img alt='' id='icon' src='zxing-icon.png'> Decode Succeeded</h1></div><table id='result'><tr><td>Raw text</td></tr><tr><td><pre>90494406</pre></td></tr><tr><td>Raw bytes</td><td><pre>(Not applicable)</pre></td></tr><tr><td>Barcode format</td><td>EAN_8</td></tr><tr><td>Parsed Result Type</td><td>PRODUCT</td></tr><tr><td>Parsed Result</td><td><pre>90494406</pre></td></tr></table></div></body></html>"
```

рис.8.

5. Розбирання запиту по частинам.

Для витягування потрібних елементів з HTML-контенту в Node-RED використовується вузол «HTML». Ознайомтеся з його роботою з [довідника](#).

З розділу function палітри вузлів вставте «HTML». Модифікуйте програму, як це показано на рис.9. Селектор вибирає елемент HTML з ID="result" у якого вибирає усі-елементи нащадки з тегом «pre».

The screenshot shows a modified Node-RED workflow. The nodes are: 'timestamp', 'Скан', 'Формувати Запит', 'http request', 'вибірка' (Selection), and 'msg'. A red arrow points from the 'вибірка' node to the output console. The console shows the output of the 'HTML' node, which is an array of pre elements extracted from the HTML document.

```
09.01.2019, 20:12:01 node: 47fab888.4b1328  
msg: Object  
  object  
    _msgid: "9d8a25ad.cf41e8"  
    topic: ""  
    payload: array[3]  
      0: "90494406"  
      1: "(Not applicable)"  
      2: "90494406"
```

рис.9.

Зробіть розгортання, активуйте Inject, через кілька секунд на панелі відлагодження з'явиться інформація про масив вибірки.

Додайте вузол-функцію для присвоєння значення msg.ID рівним номеру штрих-коду при позитивній відповіді, та "ERROR" при помилці обробки (рис.10). Зробіть розгортання, активуйте Inject та перевірте роботу програми.

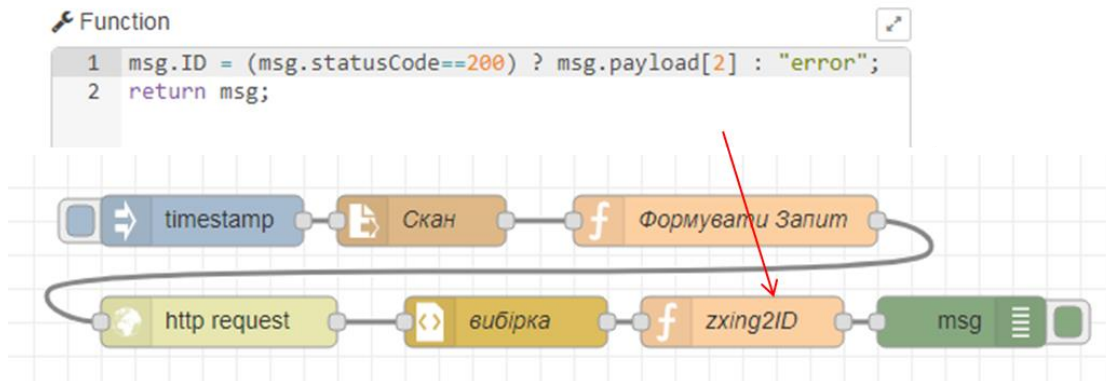


рис.10.

2.3. Використання відкритого WEB API

Ознайомтеся з принципами роботи WEB-API та REST (даються в лекційному матеріалі).

Багато застосунків в Інтернеті мають відкритий API інтерфейс для доступу до різних ресурсів як сервісів чи даних. Більшість з них є платними і надаються за підпискою. Деякі з них мають можливість обмеженого користування на певний період чи на певну продуктивність.

Дана частина лабораторної роботи призначена для знайомства з тестовими утилітами та для самостійної побудови WEB-клієнта для доступу через API з Node-RED.

1. Знайомство з сервісами IPAPI.

Веб-застосунок [IPAPI](https://ipapi.co/) дає можливість визначити деталі місця розташування за IP адресою. Це можна зробити через сторінку Веб-інтерфейсу, або через відкритий API-інтерфейс (обмеження на безкоштовне використання до 1000 запитів/день). Повний опис API доступний за [посиланням](#).

Зайдіть на сторінку за посиланням <https://ipapi.co/>. Ознайомтеся зі змістом сторінки. Зверніть увагу на ту інформацію, яка надається по IP-адресі, а також на значення Вашої білої адреси IP, вірніше від якої Ваш пристрій спілкується в Інтернеті. Слід розуміти, що у більшості випадків видима IP-адреса – це одна з адрес провайдера, що надає послуги Інтернету, тому координати будуть саме цього провайдера.

Подивіться на приклад запиту і відповіді в форматі JSON.

2. Робота з онлайн утилітами для API-тестування.

Для тестування API Ви можете користуватися будь якою утилітою, наприклад <https://apitester.com/> або <http://restninja.io/>

Відкрийте сторінку однієї з утиліт. Для перевірки роботи IPAPI введіть в поле адреси <https://ipapi.co/8.8.8.8/json/> у метод – «GET», і натисніть «SEND» (або TEST). Проаналізуйте відповідь

Повторіть те саме зі своїм білим IP. Порівняйте отримані результати з тими, що показані на сторінці <https://ipapi.co>

3. Створення клієнту для IPAPI в Node-RED.

Створіть новий потік і реалізуйте програму, як показано на рис.11. Вузол "Http request" повинен робити запит на отримання даних про свою адресу IP, тобто в форматі <https://ipapi.co/json/>. Результат запиту вивести у вікно налагоджування. Усі інші налаштування зробіть **самостійно**. Зверніть увагу, що **IPAPI може повернути негативну відповідь**. Подивіться поля заголовків <https://apitester.com/> в запиті, і заповніть їх аналогічно у вузлі «headers».

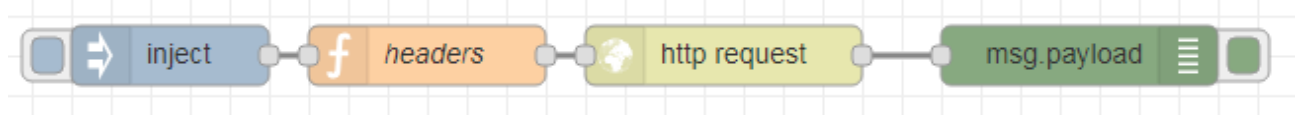


рис.11.

Зробіть копію фрагменту екрану виводу у вікні відлагодження і збережіть для звіту.

2.4. Робота з FRED - хмарним сервісом Node-RED

На хмарних платформах, зокрема IBM, інструмент Node-RED можна використовувати як середовище для розробки програмних додатків. Є також можливість скористатися окремим хмарним сервісом FRED <https://fred.sensetecnic.com/>. Ця платформа має безкоштовну ліцензію, обмежену 50-ма вузлами та потребує перезапуску (працює добу, після чого треба перезапускати). Для користування платформою потрібна безкоштовна реєстрація.

Даний сервіс буде використовуватися в лабораторних роботах в тому випадку, коли необхідно буде мати серверний додаток в Інтернет.

1. Реєстрація на FRED.

Зайдіть на <https://fred.sensetecnic.com/> ознайомтеся з умовами ліцензії.

Зайдіть на <https://users.sensetecnic.com/register>, зареєструйтеся:

- введіть необхідні дані в поля реєстрації,
- виберіть PLAN Fred Short.
- виставте одну із опцій «How would you describe your use of FRED? (select one)»
- виставте одну або кілька опцій «What industry or area of interest are you using FRED for? (select all that apply)»
- виставте опцію «I agree to the Terms of Use»
- виставте опцію «Я не робот»
- натисніть “Create An Account”
- через кілька хвилин на пошту прийде лист від info@sensetecnic.com, в якому буде посилання «Verify your email address» для активації акаунту

2. Запуск FRED.

Зайдіть на <https://fred.sensetecnic.com/> і якщо Node-RED не виконується, запустіть його натисканням “Start Instance”. Сховайте ліву бокову панель кнопкою «Toggle sidebar» (стрілка в лівому нижньому кутку екрану)

Видаліть усі вузли в потоці, які вставлені туди при створенні акаунту.

Створіть невеличку програму, наприклад, яка складається з двох з'єднаних вузлів «inject» та «debug», та перевірте її роботу.

2.5. Використання WEB-сокетів

Ознайомтеся з принципами роботи WEB-сокетів (даються в лекційному матеріалі).

В даній частині лабораторної роботи WEB-сокети будуть використовуватися для наступних цілей:

- зв'язку між локальним (ПК) та віддаленим (FRED) Node-RED, тобто в якості подовжувача зв'язку між потоками
- зв'язку Node-RED з застосунками в Інтернет
- зв'язку Node-RED з застосунками на мобільних пристроях

При роботі з HTTP (у тому числі WEB-сокетами) FRED, як посередник, передає всі повідомлення з Інтернету на екземпляр Node-RED. Щоб відправити запит до конкретного екземпляру, FRED використовує ім'я користувача в заголовку для **приватних** вузлів введення API або URL для **загальнодоступних** вузлів вводу Websockets.

Для створення загальнодоступного серверного вузла WebSocket, необхідно в конфігураційному вузлі WebSocket в URL-адресі прослушування вказати префікс «/public/». Тоді можна буде отримати доступ до websocket в

wss:// {ім'я користувача} .fred.sensetecnic.com/api/public/{custom_endpoint_name}.

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

Наприклад, вхідний вузол websocket, налаштований на шлях /public /data, створений користувачем mike, буде доступний як

`wss: //mike.fred.sensetecnic.com/api/public/data`

Для того, щоб клієнт Websockets отримував доступ до захищених вузлів вводу WebSocket у потоках, необхідно надати своє ім'я користувача та пароль, використовуючи базову аутентифікацію, або ключ API з вашими запитам. Щоб зробити запит до захищеного вузла вводу веб-сокету, ім'я користувача та ключ API повинні бути додані до заголовків запиту: x-auth-user та x-auth-key. В даній лабораторній роботі не будуть використовуватися захищені приватні з'єднання. З деталями роботи з Веб-сокетами на платформі FRED, можна ознайомитися [тут](#).

1. Створення серверного WEB-сокету в FRED.

В Node-RED на платформі FRED створіть вхідний серверний "WebSocket In" та з'єднайте його з вузлом "Debug" (рис.12)

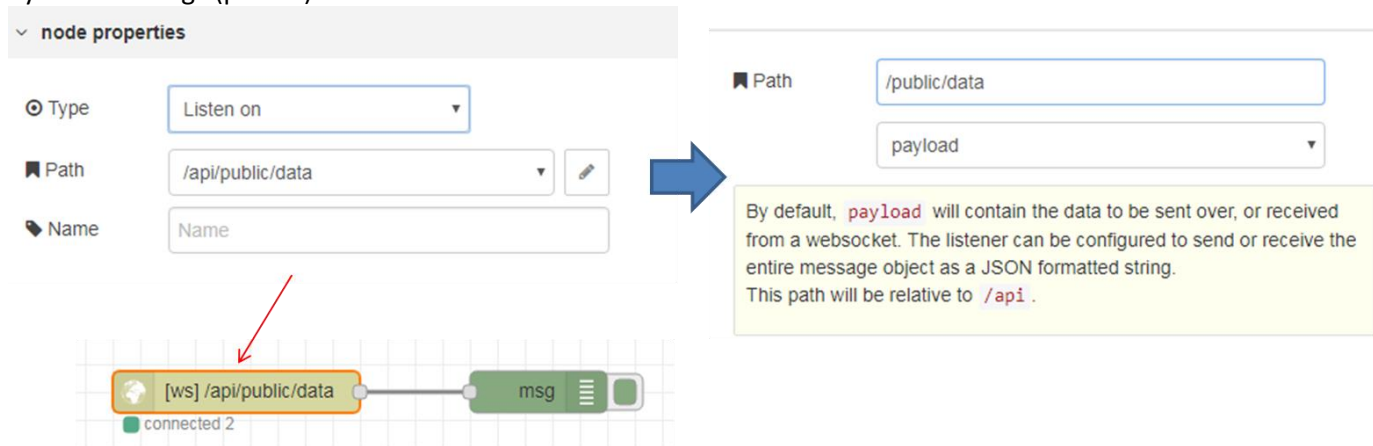


рис.12.

Відкрийте Веб-інтерфейс Dashboard FRED, скопіюйте адресу, вона повинна мати вигляд: `https://account_name.fred.sensetecnic.com/api/ui/` де `account_name` – ім'я вашої реєстрації

2. Створення клієнтського WEB-сокету в локальному Node-RED.

Запустіть Node-RED на локальному ПК. Створіть новий потік, в ньому розмістіть два вузли – клієнтський "WebSocket out" та "Inject" (рис.13). Налаштуйте відповідно до вашого серверного сокету. Перевірте роботу з'єднання. Зробіть копію екрану потоку на FRED, на якій видно адресу вашого аккаунту і наявність з'єднання – зелений напис «connected». Додайте його до звіту.

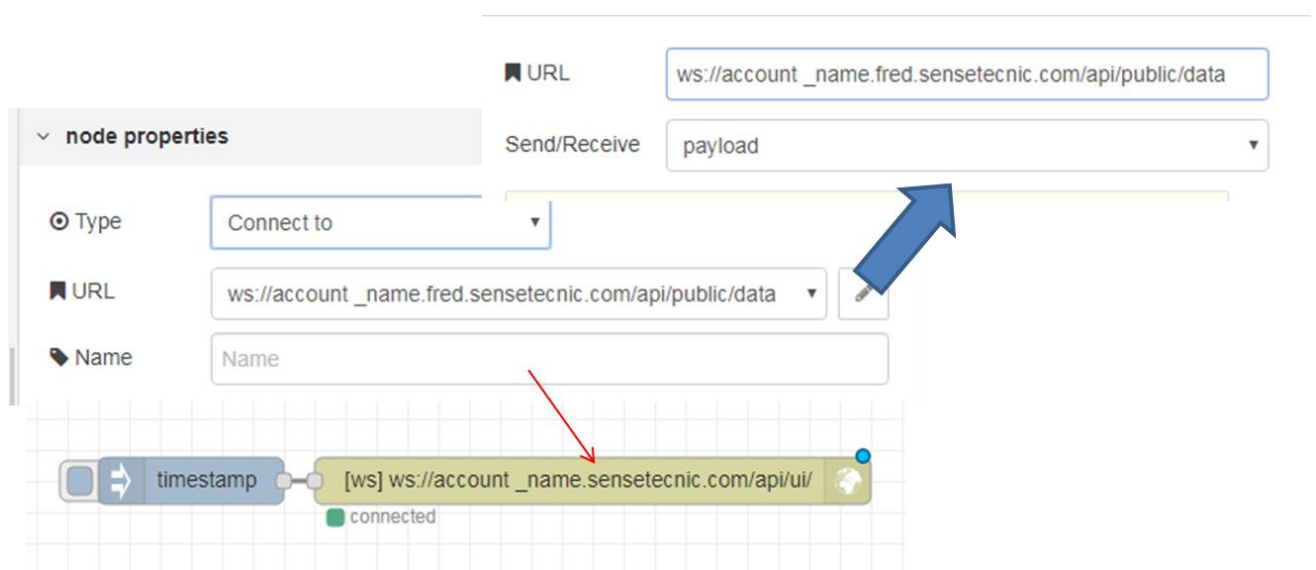


рис.13.

3. Перевірка роботи сервісів WEB-сокету через онлайн утиліти.

10.12.2018

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

Самостійно зробіть тестування клієнтського WEB-сокету в локальному Node-RED, під'єднавшись до <wss://echo.websocket.org>. Цей сервер передбачає ехо-відповіді на запити, тому для перевірки його роботи необхідно як вхідний так і вихідний клієнтські сокети.

Самостійно зробіть тестування створеного раніше серверного WEB-сокету на FRED через [sevic https://www.websocket.org/echo.html](https://www.websocket.org/echo.html)

Наступні частини не є обов'язковими для виконання!

2.6. Використання WEB-сокетів для з'єднання мобільного застосунку та Node-RED

В даній частині лабораторної роботи використовується безкоштовна версія застосунку [Wireless Barcode Scanner](#) для пристроїв з Android. Даний застосунок дає можливість відсканувати і розпізнати штрих-код і передати його по наступним мережним протоколам:

- Bluetooth,
- TCP/IP,
- UDP
- HTTP
- WebSocket

В даній частині в якості віддаленого сканера пропонується використати пристрій з Android, який буде передавати сканований штрих-код по WebSocket через WiFi. Мобільний пристрій і ПК повинні бути в одній мережі.

1. Налаштування дозволів використання вхідних портів на ПК.

Типово, вхідні порти ПК заблоковані брандмауером Windows. Тому необхідно налаштувати брандмауер Windows для можливості використання вхідного порту з мережі, а саме:

- відключити правила, що забороняють доступ до порту 1880 та роботи Node.js
- додати правило, що дозволяє доступ до порту 1880

Відкрийте налаштування брандмауера Windows через «Панель керування»-«Адміністрування» - «брандмауер Windows». Відключіть усі правила що забороняють доступ JavaScript Node.js, додайте та активуйте правило для вхідних підключень по порту 1880 (рис.14)

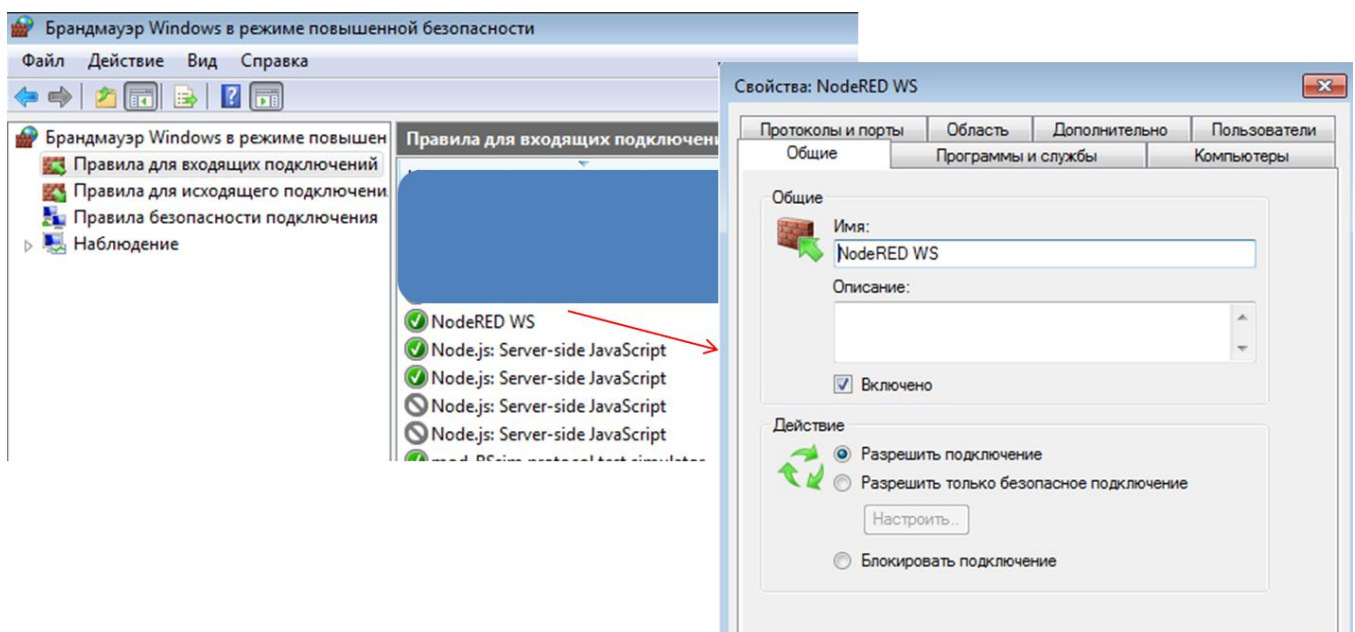


рис.14.

2. Конфігурація WebSocket server в застосунку Node-RED.

10.12.2018

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

Відкрийте програму, що була створена в частині 2.2 «Реалізація клієнта HTTP в Node-RED» (Потік «Laba2_WEBAPI»). Вставте вузол «Websocket In», що буде очікувати з'єднання та отримувати по ньому дані (рис.15).

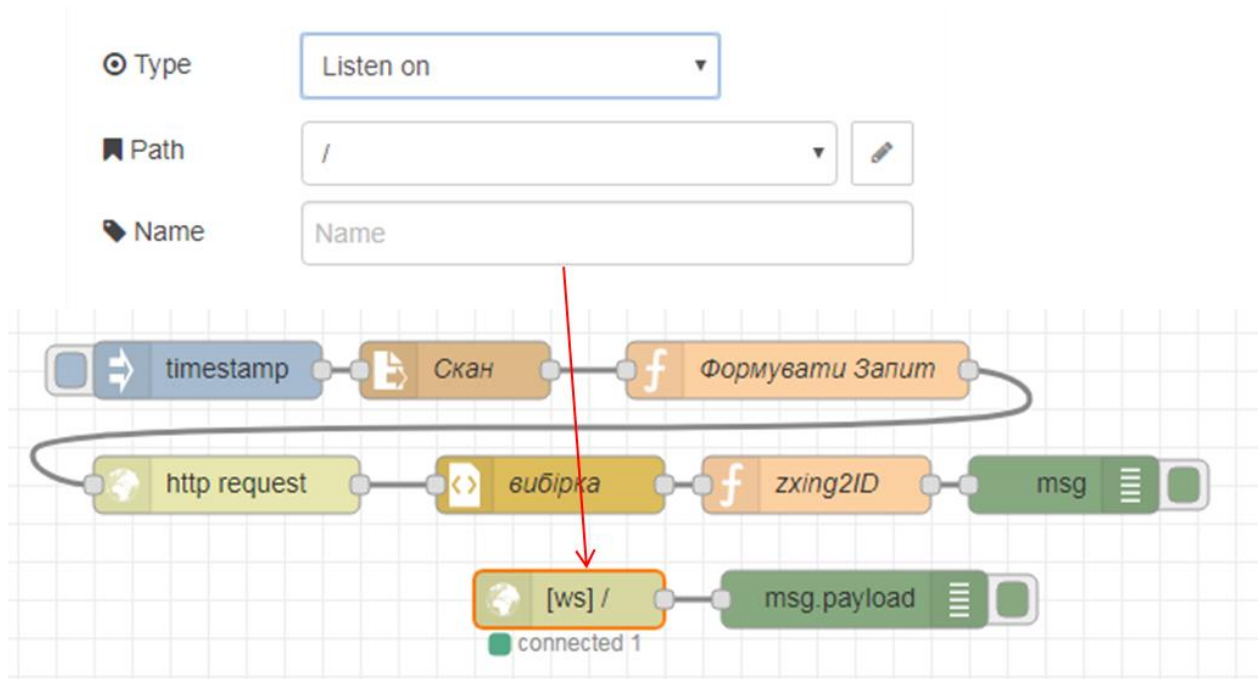


рис.15.

3. Встановлення та налаштування мобільного застосунку.

Встановіть на мобільний пристрій безкоштовну версію [Wireless Barcode Scanner](#) .

Визначте IP-адресу ПК, через яку буде проводитися доступ до нього з мережі (можна подивитися у властивостях мережної карти).

Запустіть застосунок Wireless Barcode Scanner. Відкрийте конфігурацію через пункт меню «setting». Вставте наступні налаштування в розділі Data link:

- Data transfer – вкл.
- Connection type – WebSocket client
- Configuration – Host=xx.xx.xx.xx, Port=1880
де xx.xx.xx.xx – номер IP ПК з Node-RED

Вийдіть з конфігурації застосунку в режим сканування. Якщо з'єднання відбулося, внизу екрана з'явиться зелена літера «i», якщо ні – червона «!» .

4. Перевірка роботи.

У Wireless Barcode Scanner в робочому режимі відскануйте штрих-код шляхом натискання кнопки «Scan».

На панелі відлагодження Node-RED повинен з'явитися буфер відсканованих символів, що закінчуються двома символами – кінця і переведення рядку , тобто 0xd та 0xa (рис.16).

```
msg.payload : buffer[15]
  ▼ buffer[15]
    ▶ [0 ... 9]
    ▼ [10 ... 14]
      10: 0x36
      11: 0x30
      12: 0x38
      13: 0xd
      14: 0xa
```


5. Визначення ID.

Добавте функцію яка буде перетворювати отриманий буфер в символний рядок ID (рис.17).

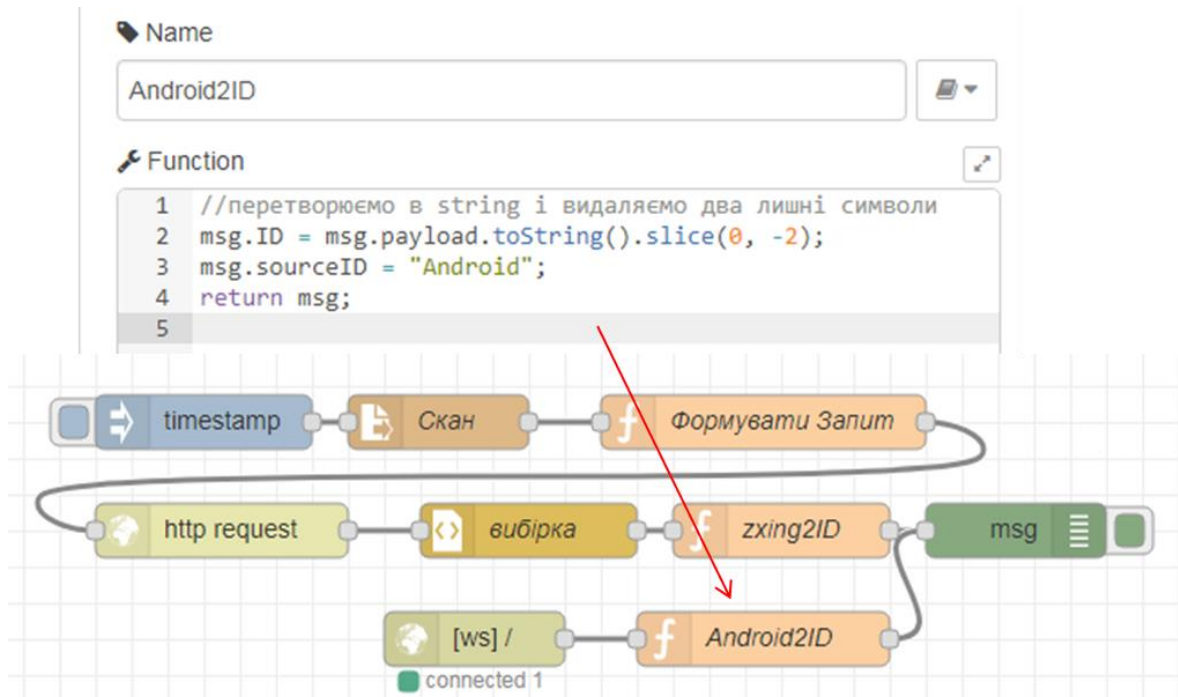


рис.17

Перевірте роботу, проконтролюйте щоб ID в повідомленні співпадав з ID на штрих-коді.

2.7. Доробка застосунку в Node-RED для фіксації штрих-кодів в базі даних та виведення по ним інформації.

У даній частині лабораторної роботи необхідно доробити розроблену програму таким чином, щоб вона виконувала наступні функції:

- ведення бази даних (текстовий файл) по продуктам за зчитуваними штрих-кодами; в БД записується:
 - країна походження продукту: визначається автоматично за номером штрих-коду EAN-13
 - джерело надходження в базу номеру штрих-коду: ім'я файлу зображення штрих-коду або напис «Android»
 - дата та час надходження в базу номеру штрих-коду
 - найменування продукту: вноситься вручну
 - опис продукту: вноситься вручну
 - посилання на PDF-файл документації: вноситься вручну
- перевірка наявності в директорії нових файлів з розширенням *.jpg і *.png, автоматичне їх сканування і перейменування (для уникнення повторної обробки)
- внесення в БД тільки тих штрих-кодів, які в ній відсутні
- при скануванні існуючого в БД штрих-коду на веб-інтерфейс повинна бути виведена інформація за вказаним продуктом включно з PDF-файлом документації
- помилка читання повинна бути зафіксована в журналі подій

1. Формування бази країн по EAN-коду.

[Завантажте та інсталюйте Notepad++](#), який Вам знадобиться для перегляду та коректної правки текстових файлів.

Відкрийте потік LABA2_WEBAPI. Імпортуйте в потік частину програми [з цього посилання](#). Вона матиме вигляд, як на рис.18.

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net



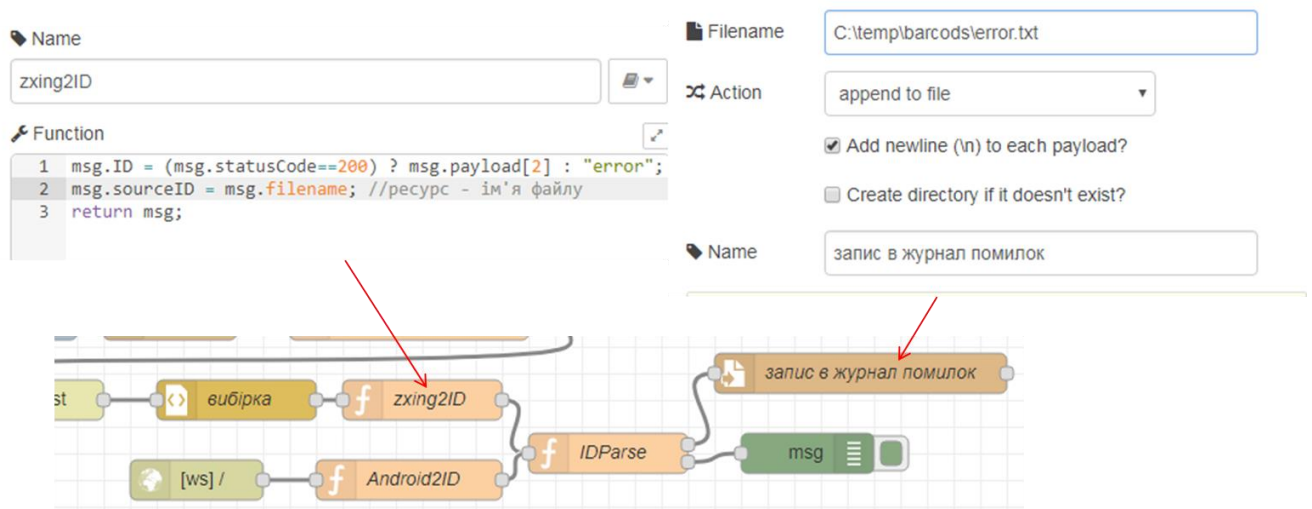
рис.18.

Зробіть розгортання проекту. Активуйте відображення повідомлень з Msg (воно за замовчуванням в прихованому режимі). При старті системи, або при активації кнопкою елементу Inject "Start" у вікно налагодження буде виведено масив, в якому індекс відповідає за номер країни, а значення – за назву. Про деталі відповідності номеру країни можна прочитати [за цим посиланням](#).

Подивіться на вміст функції EAN. При старті значення кодів також будуть записуватися в контекст потоку, у його властивість «arEAN», значення якої можна буде прочитати у будь-якому вузлі цього потоку.

2. Модифікація структури запису по продукту та виведення помилок сканування.

Імпортуйте в потік вузол-функцію «IDparse» [з цього посилання](#). Модифікуйте програму щоб вона виглядала так, як на рис.19.



```
1 msg.ID = (msg.statusCode==200) ? msg.payload[2] : "error";
2 msg.sourceID = msg.filename; //ресурс - ім'я файлу
3 return msg;
```

рис.19.

Зробіть розгортання програми, активуйте повідомлення вприскуванням Inject. Тепер після обробки запиту на панелі відлагодження повинен з'явитися об'єкт з властивістю prodinfo, в якій будуть поля та значення, що описані в переліку функцій.

3. Запис в базу даних (текстовий файл).

Модифікуйте програму так, як показано на рис.20.

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

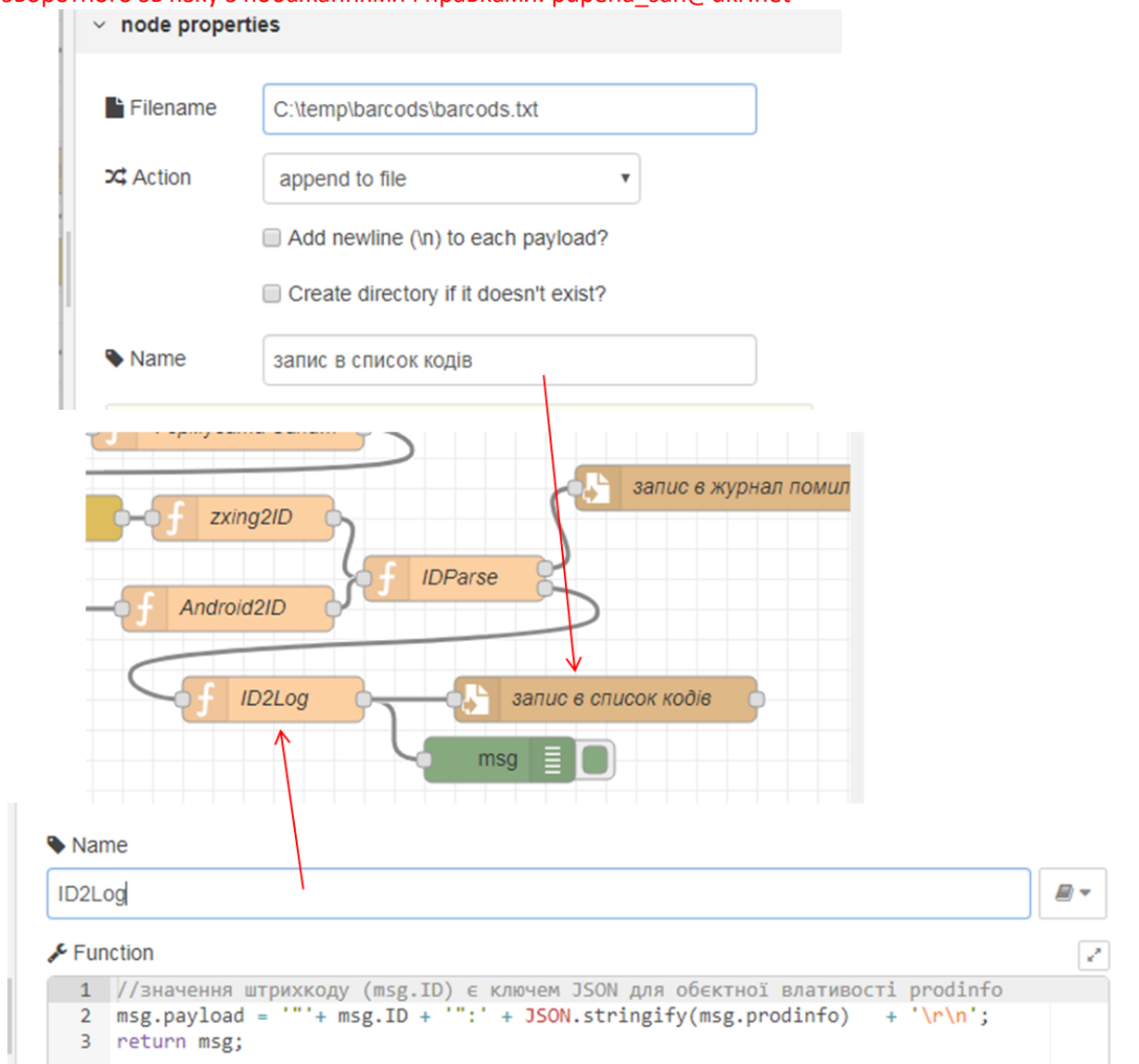


рис.20

Розгорніть та запустіть на виконання, активуйте перерахунок потоку. В файл barcodes.txt повинен записатися об'єкт з записами prodingo.

Примітка. Якщо в файлі некоректно відображається кирилиця (характерно для Windows 7 x86) необхідно зробити наступні кроки:

- встановити в Node-RED модуль contrib-iconv (рис.21)
- модифікувати програму, як показано на рис.22

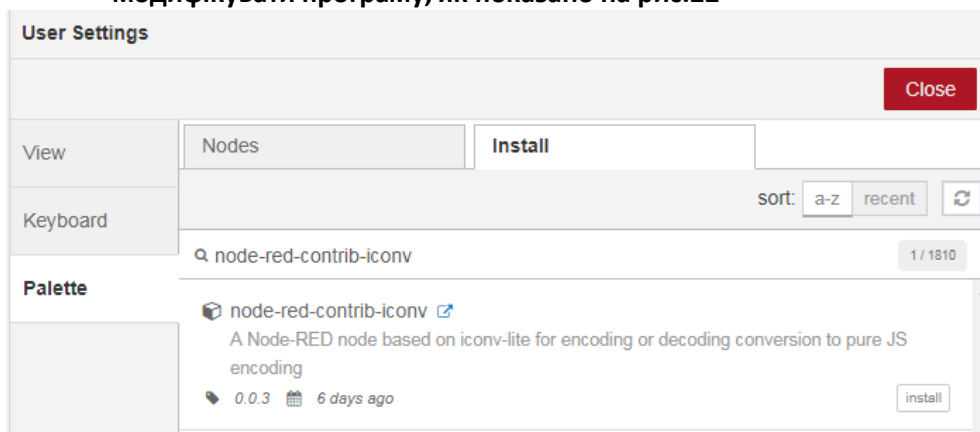


рис.21

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

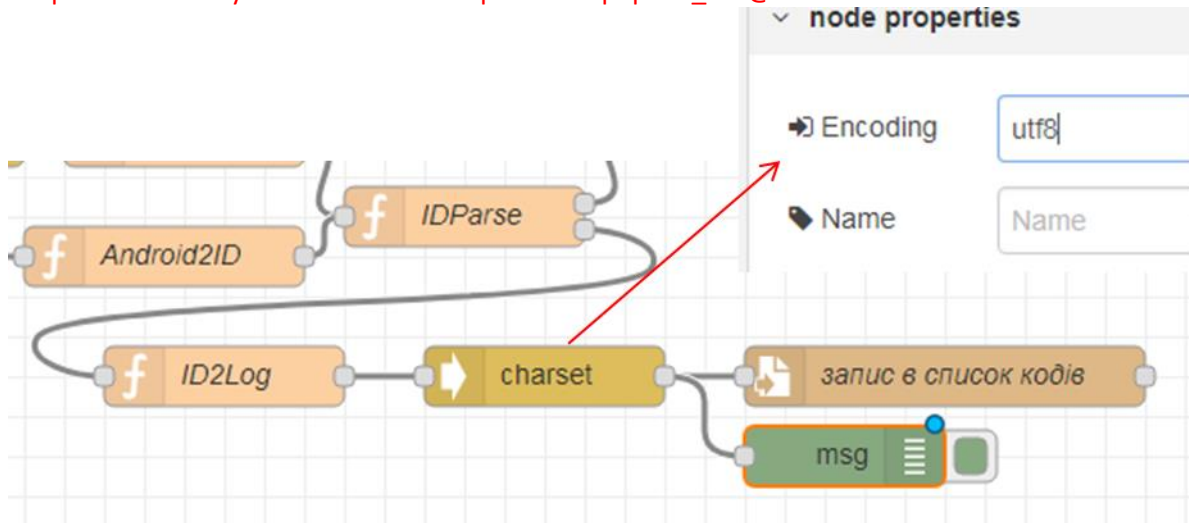


рис.22

4. Перевірка наявності в директорії файлів з розширенням *.jpg і *.png.

Завантажте [файли зі штрих-кодами](#) в директорію «C:\Temp\barcodes\raw» вони будуть використовуватися як файли з зображеннями для сканування. **Для пришвидшення сканування можете обробити файли так, щоб на них залишилися тільки зображення штрих-кодів.**

Інсталюйте модуль «node-red-contrib-fs-ops», в ньому знаходяться два вузли, які знадобляться для лабораторної роботи “dir” і “move”. Познайомтеся з принципами їх роботи з [довідника](#).

Перевірте роботу алгоритму вибірки файлів за фільтрами *.jpg та *.png за допомогою фрагмента програми, показано на рис.23.

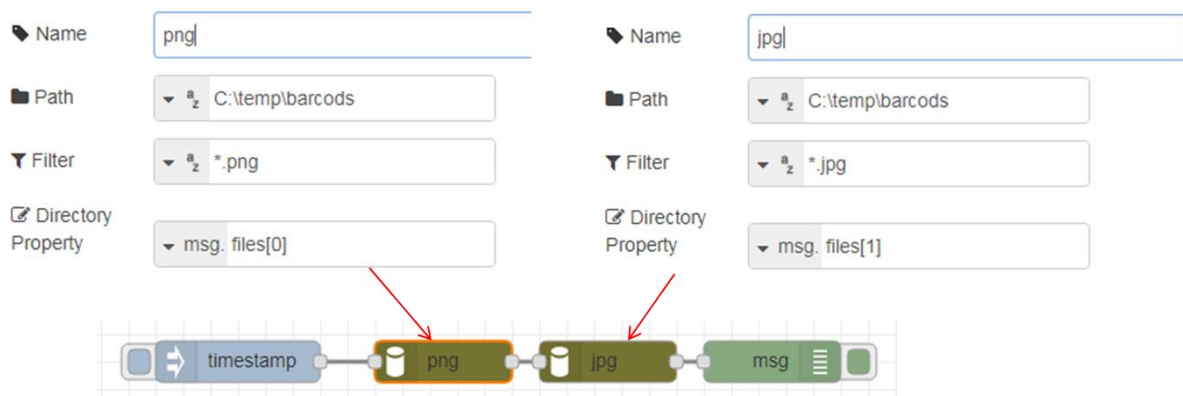


рис.23

Зробіть розгортання, перекопіюйте декілька файлів з директорії «C:\Temp\barcodes\raw» в «C:\Temp\barcodes», запустіть фрагмент на виконання і подивіться на вміст панелі відлагодження.

5. Пошук графічних файлів в директорії.

Імпортуйте вузол-функцію “find” [з цього посилання](#). Цей вузол об'єднує масиви назв «*.jpg» та «*.png» і шукає в них перший файл, в якого назва не містить послідовність літер «cmplt» (ця послідовність використовуватиметься для маркування оброблених файлів). Модифікуйте фрагмент програми, як це показано на рис.24. Перевірте його роботу.



рис.24

6. Автоматичне сканування і перейменування знайдених файлів.

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

Модифікуйте програму, щоб вона мала вигляд як на рис.25. Зверніть увагу, що назву файлу з вузлу «Скан» треба видалити, тільки в такому разі назва файлу буде братися з властивості повідомлення "msg.filename". Перевірте як працює програма.

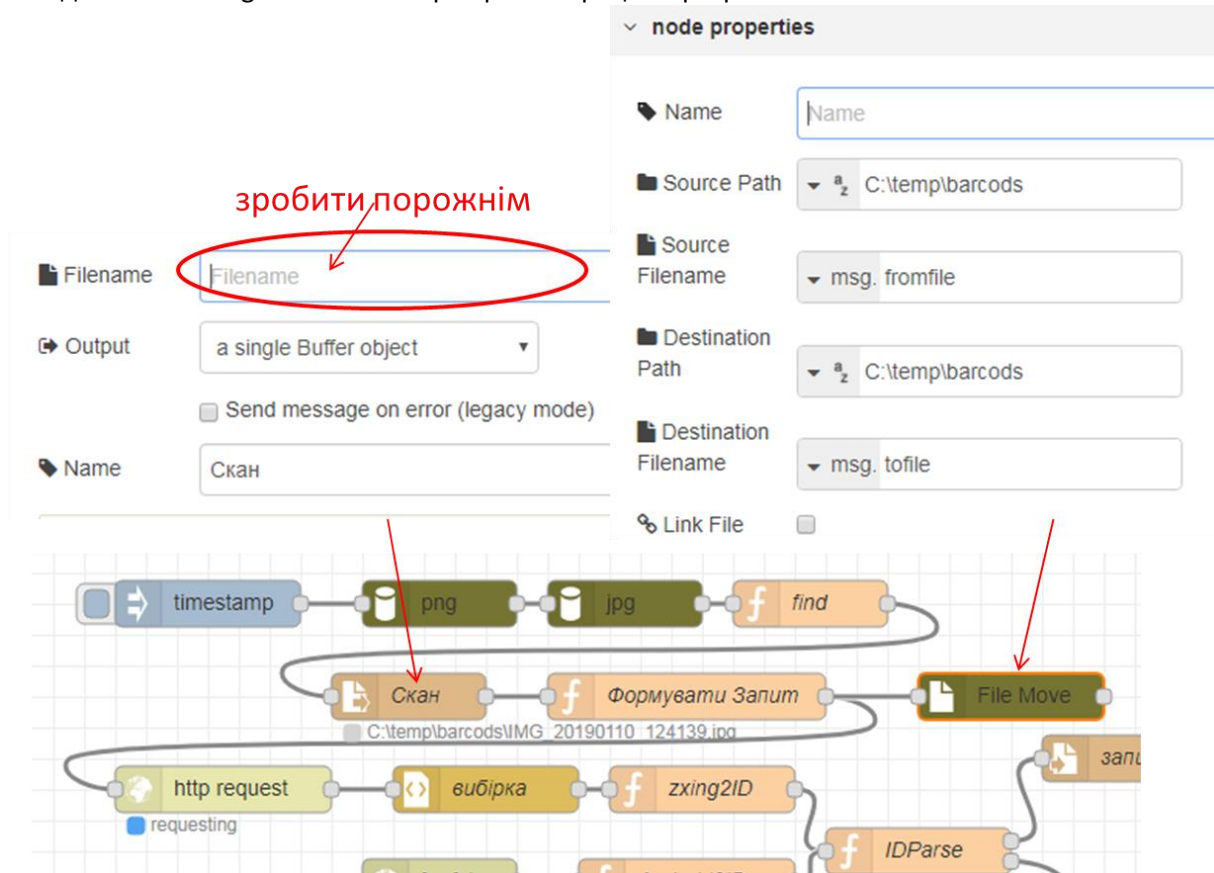


рис.25

Модифікуйте програму так, щоб вузол «Timestamp» генерував повідомлення кожні 2 секунди. Таким чином кожні 2 секунди буде скануватися один із графічних файлів і перейменовуватися з префіксом «смплт_».

7. Синхронізація сканування та періодичності обробки файлів.

Видаліть усі графічні та текстові файли з директорії «C:\Temp\barcodes». Скопіюйте усі графічні файли з директорії «C:\Temp\barcodes\raw» в «C:\Temp\barcodes». Прослідкуйте з якою частотою обробляються запити у вузлі «http request» та «File move». Node-RED дозволяє розсинхронізувати потоки таким чином, що на вхід вузла може прийти декілька повідомлень, які він буде обробляти поступово, буферизуючи їх. Однак в деяких випадках така обробка небажана. Тому в даному пункті необхідно зробити синхронізацію: пошук нових файлів проводити тільки у випадку, коли не відбувається очікування відповіді на запит «http request». Контроль стану вузла можна проводити за допомогою вузла «Status».

Модифікуйте програму так, щоб відбувалася синхронізація між знаходженням нового файлу і відсутності очікування відповіді від серверу (рис.26). Також модифікуйте функцію «find» (рис.27).

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

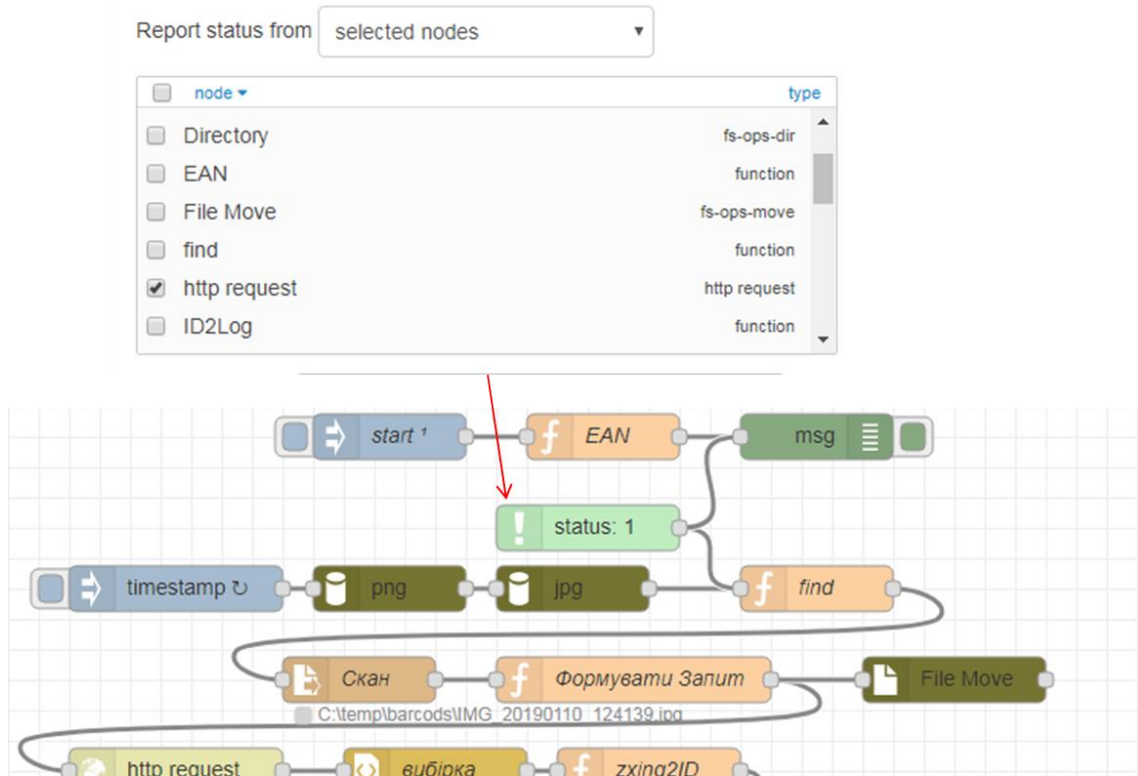


рис.26

```

1 //отримання значення з контексту "Занятість", або ініціалізація
2 var busy = context.get ("busy") || false;
3 //якщо отримали повідомлення від вузла Status
4 if (msg.status!==undefined) {
5     //якщо значення тексту статусу "httpin.status.requesting"
6     if (msg.status.text ==="httpin.status.requesting")
7         busy = true;// виставляємо "Занятість"
8     else busy = false;//якщо інше значення тексту статусу
9     context.set ("busy", busy);//записати значення в контекст вузла
10 }
11 //зчитувати назви файлів тільки коли не Занятий і повідомлення прийшло від Directory
12 if (!busy && msg.files !== undefined) {
13     var path = "C:\\temp\\barcodes\\";//шлях
14     //об'єднуємо масиви назв файлів *.png і *.jpg в один
15     var files = msg.files[0].concat(msg.files[1]);
16     msg.files = undefined;//видаляємо старі масиви
17     //перебираємо назви файлів в масиві
18     for (var i=0; i < files.length; i++){
19         if (files[i].indexOf('cmplt')== -1) { //якщо в імені НЕ є 'cmplt'
20             msg.filename = path + files[i]; //записуємо в filename повне ім'я з шляхом
21             msg.fromfile = files[i]; //в fromfile ім'я файлу
22             msg.tofile = "cmplt_" + files[i]; //в tofile ім'я для перейменування
23             return msg;
24         }
25     }
26 }

```

рис.27

7. Внесення в БД тільки тих штрих-кодів, які в ній відсутні.

Імпортуйте вузол-функцію «Наявна в БД?» з цього посилання. Модифікуйте програму як це показано на рис.28. Вузол "Directory" перевіряє наявність файлу «barcodes.txt» в директорії. Якщо файлу не існує, вузол «switch» направляє повідомлення тим саме шляхом, як і до модифікації програми. Якщо файл є, повідомлення ініціює вузол «Читання», який зчитує зміст БД. Вузол-функція «Наявна в БД» перевіряє, чи є запис в БД, і якщо ні направляє повідомлення стандартним шляхом.

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

Якщо запис в базі існує, то на другому (не підключеному) виході формується повідомлення з об'єктом, зчитаним з БД. Цей вихід буде використовуватися в подальшому.

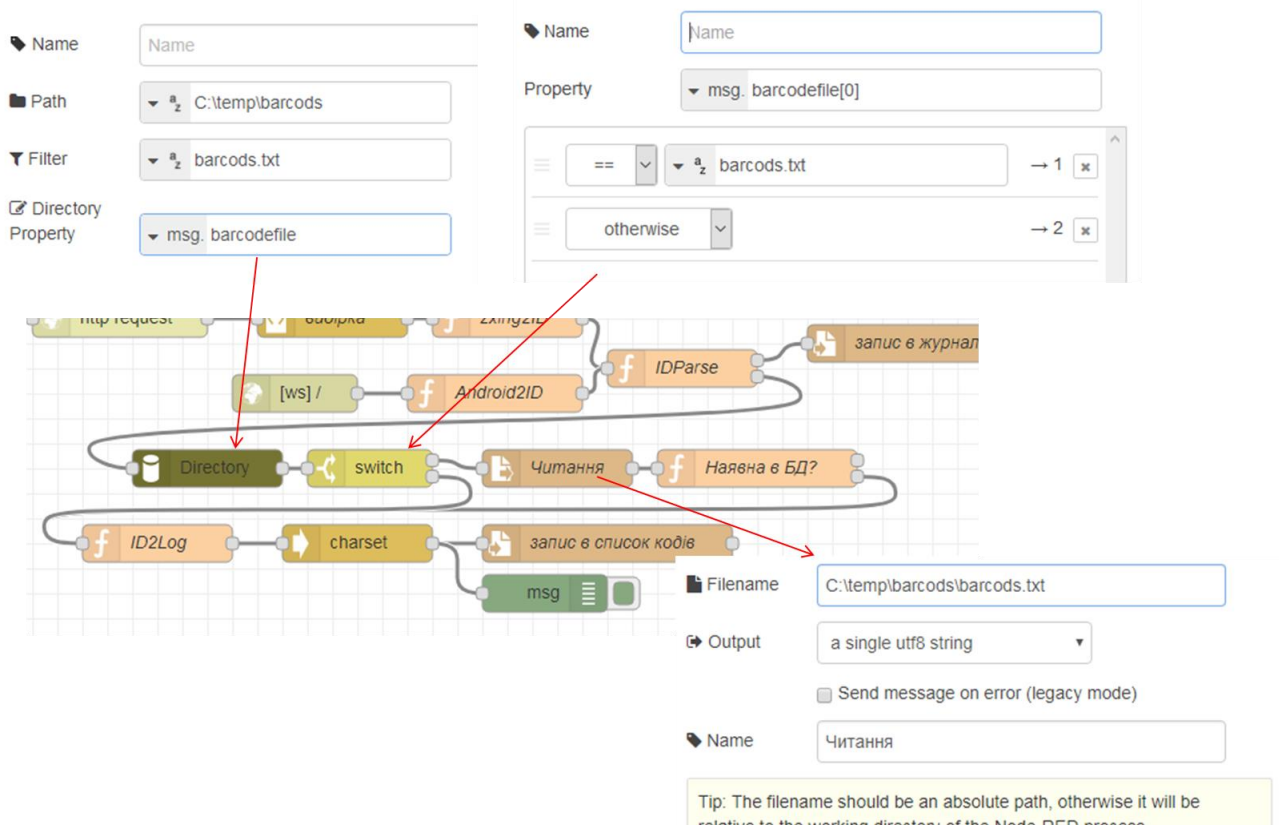


рис.28

Видаліть усі графічні та текстові файли з директорії «C:\Temp\barcodes». Скопіюйте усі графічні файли з директорії «C:\Temp\barcodes\raw» в «C:\Temp\barcodes». Зробіть розгортання потоку і запустіть на виконання. Після обробки всіх файлів, скопіюйте ще раз один з них, в БД не повинен з'явитися новий запис.

7. Виведення інформації на ВЕБ-інтерфейс при скануванні існуючого в БД штрих-коду.

Імпортуйте вузли для ВЕБ-інтерфейсу з [цього посилання](#). Модифікуйте програму як показано на рис.29. Зробіть розгортання.

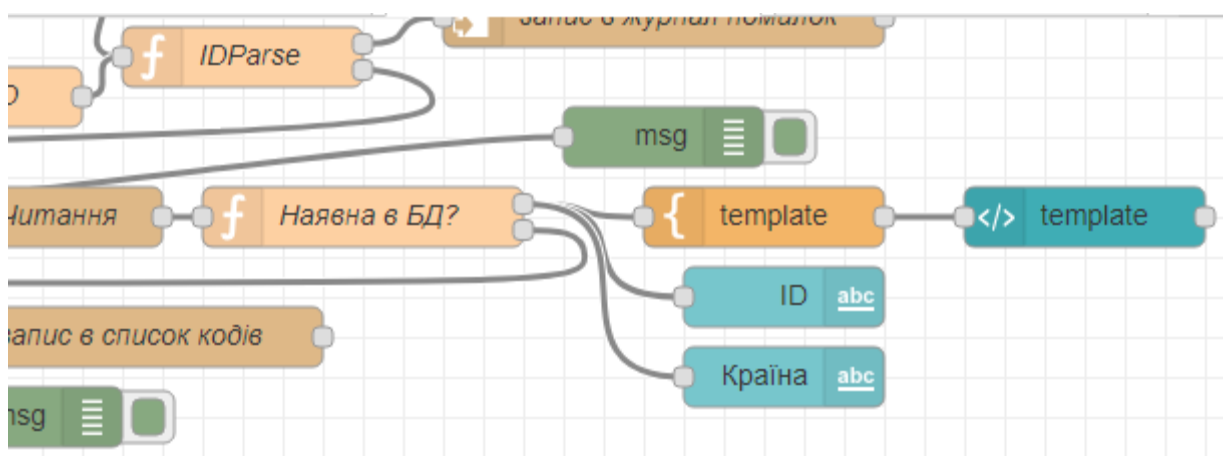


рис.29

Відкрийте Веб-інтерфейс на закладці «Штрих-коди». Скопіюйте один із файлів зображень штрих-коду що є в БД в директорію «C:\Temp\barcodes» або зробіть його скан з мобільного застосунку. На Веб-інтерфейсі повинен з'явитися його код, та країна походження. Повторіть те саме з іншим файлом.

Робоча чернетка. Автор - Олександр Пупена, доц. ІАСУ НУХТ. Поширюється для тестування і зворотного зв'язку з побажаннями і правками. pupena_san@ukr.net

8. Ручна модифікація БД та виведення додаткової інформації на ВЕБ-інтерфейс при скануванні існуючого в БД штрих-коду.

Для двох або більше штрих-кодів із файлу «barcodes.txt» знайдіть назви, опис та посилання на файл PDF з Інтернету. Заповніть ці поля в БД, використовуючи Notepad++. Наприклад:

```
"3606480071058":{"Country":"Франція","sourceID":"C:\\temp\\barcodes\\IMG_20190110_124139.jpg","sourceDoc":"http://elprivod.nmu.org.ua/ua/student/techdoc/pch/atv12_%D1%80%D1%83%D0%BA_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7.pdf","name":"ATV12H037M2","descr":"variable speed drive ATV12 - 0.37kW - 0.55hp - 200..240V - 1ph - with heat sink","inputdate":"2019-1-11 23:34:54"}
```

```
"3389110146608":{"Country":"Франція","sourceID":"C:\\temp\\barcodes\\IMG_20190110_124417.jpg","sourceDoc":"https://www.alliedelec.com/m/d/68afa7d7064e3bef2fc703440bf8f780.pdf","name":"XS630B1P AL2","descr":"inductive sensor XS6 M30 - L62mm - brass - Sn15mm - 12..48VDC - cable 2m","inputdate":"2019-1-11 23:35:24"}
```

Збережіть зміни. Скопіюйте файл з зображенням або відскануйте його мобільним застосунком Wireless Barcode Scanner. Дочекайтеся коли його буде оброблено, після чого на Веб-інтерфейсі повинен з'явитися документ, який було вказано в БД. Зробіть те саме з іншим файлом.

Самостійні завдання (необов'язкове для виконання).

- 1) Виведіть на Веб-інтерфейс всю інформацію про товар.
- 2) Наведена програма має певні вади. Зокрема, вона не класифікує штрих-коди по типу, тому невірно в ряді випадків визначає країну походження. Зробіть модифікацію так, щоб країна походження визначалася тільки для кодів EAN-13. Можна також класифікувати типи штрих-кодів та обробляти їх окремо.