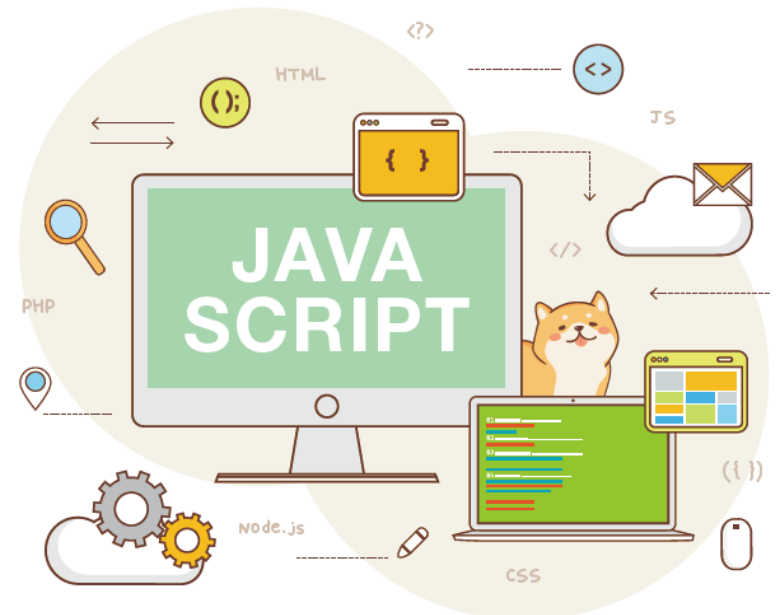




스마트헬스 케어 - 10주차

학습 내용

1. HTTP 통신(웹 요청과 응답)
2. AJAX
3. Node-Red 설치
4. Node-Red 기본 노드

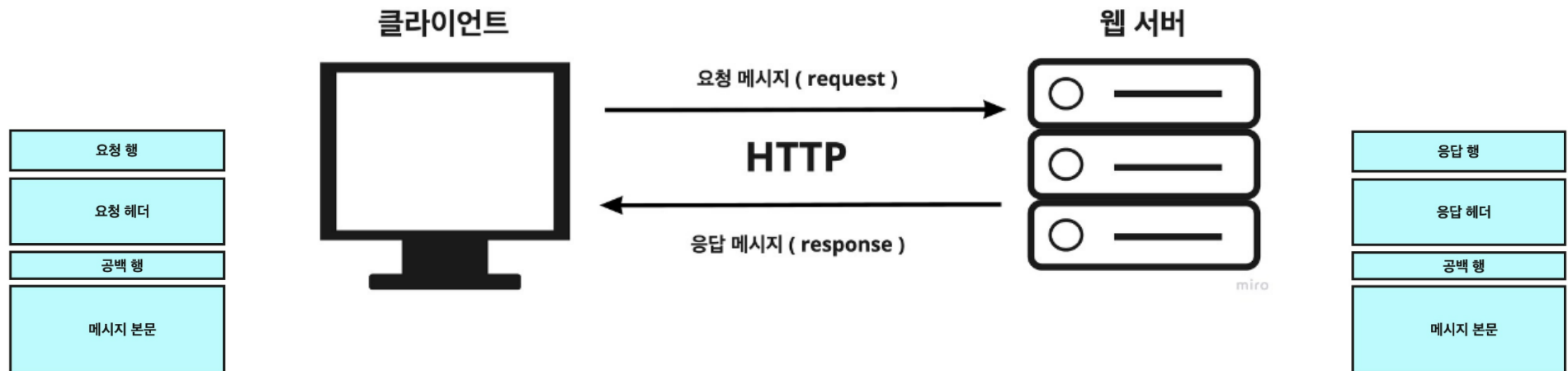


이론 및 예제 실습

1_HTTP 통신 특성

요청(Request)과 응답(Response) 방식으로 통신

- 비연결 지향 : 클라이언트 쪽에서 Request를 보내고 Server쪽에서 Response를 받으면 이어졌던 연결이 끊어짐
- Stateless 통신 : 클라이언트 상태를 저장하지 않음.
- 동기 통신 : 요청 후 응답을 받아야 다음 동작이 이루어짐

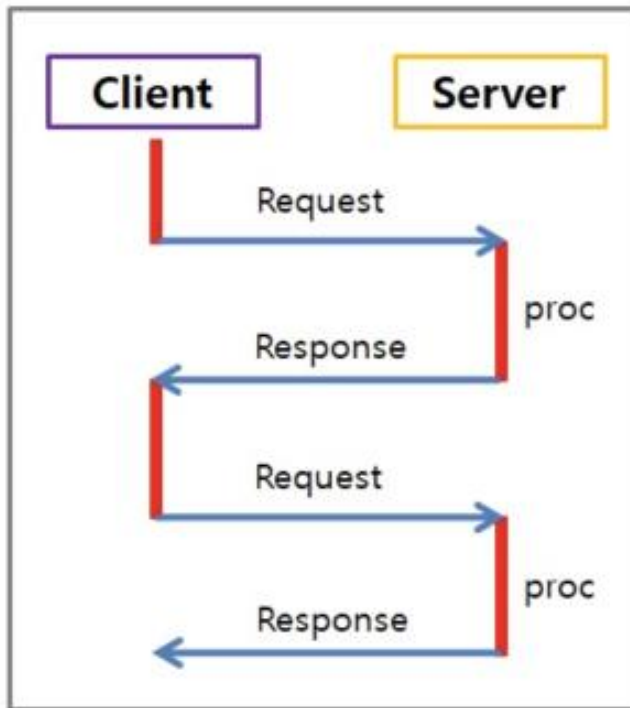


```
GET / HTTP/1.1
Host: www.naver.com:443
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/ap
exchange;q=0.9
sec-ch-ua: "Chromium";v="92", " Not A;Brand";v="99", "Google Chrome";v="92"
sec-ch-ua-mobile: ?0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/92.0.4515.159 Safari/537.36
```

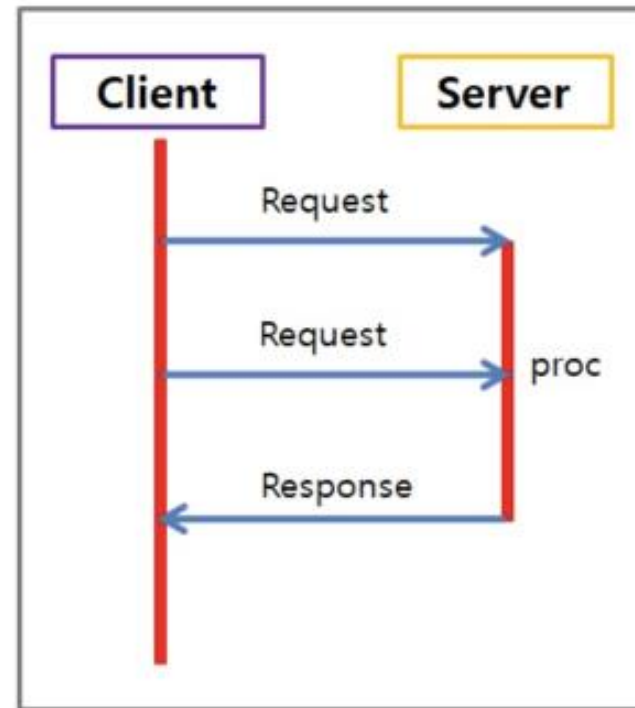
```
HTTP/1.1 200
cache-control: no-cache, no-store, must-revalidate
content-encoding: gzip
content-type: text/html; charset=UTF-8
date: Wed, 25 Aug 2021 02:42:02 GMT
p3p: CP="CAO DSP CURA ADMa TA1a PSAa OUR LAW STP PHY ONL UNI PUR FIN COM NAV INT
DEM STA PRE"
pragma: no-cache
referrer-policy: unsafe-url
server: NWS
strict-transport-security: max-age=63072000; includeSubdomains
x-frame-options: DENY
x-xss-protection: 1; mode=block
```

1_동기 통신과 비동기 통신

- 동기 통신 : 요청 후 응답을 받아야 다음 동작이 이루어짐
- 비동기 통신 : 요청 후 응답을 기다리지 않고 바로 다음 동작이 이루어지며
응답이 완료되면 이벤트를 받아서 처리



<동기식>



<비동기식>

2_REST API 개념

1. REST(Representational State Transfer) : 네트워크상에서 Client와 Server 사이의 통신 방식으로 HTTP URL (Uniform Resource Identifier)을 통해 자원(Resource)을 명시하고, HTTP Method (POST, GET, PUT, DELETE)를 통해 해당 자원에 대한 CRUD 작업을 적용하는 것

- 자원 : 해당 소프트웨어가 관리하는 모든 것 예) 문서, 그림, 데이터, 해당 소프트웨어 자체 등

2. API(Application Programming Interface) : 클라이언트가 리소스를 요청할 수 있도록 서버측에서 제공된 인터페이스(interface)를 의미

3. REST API : REST 기반으로 서비스 API를 구현하는 것

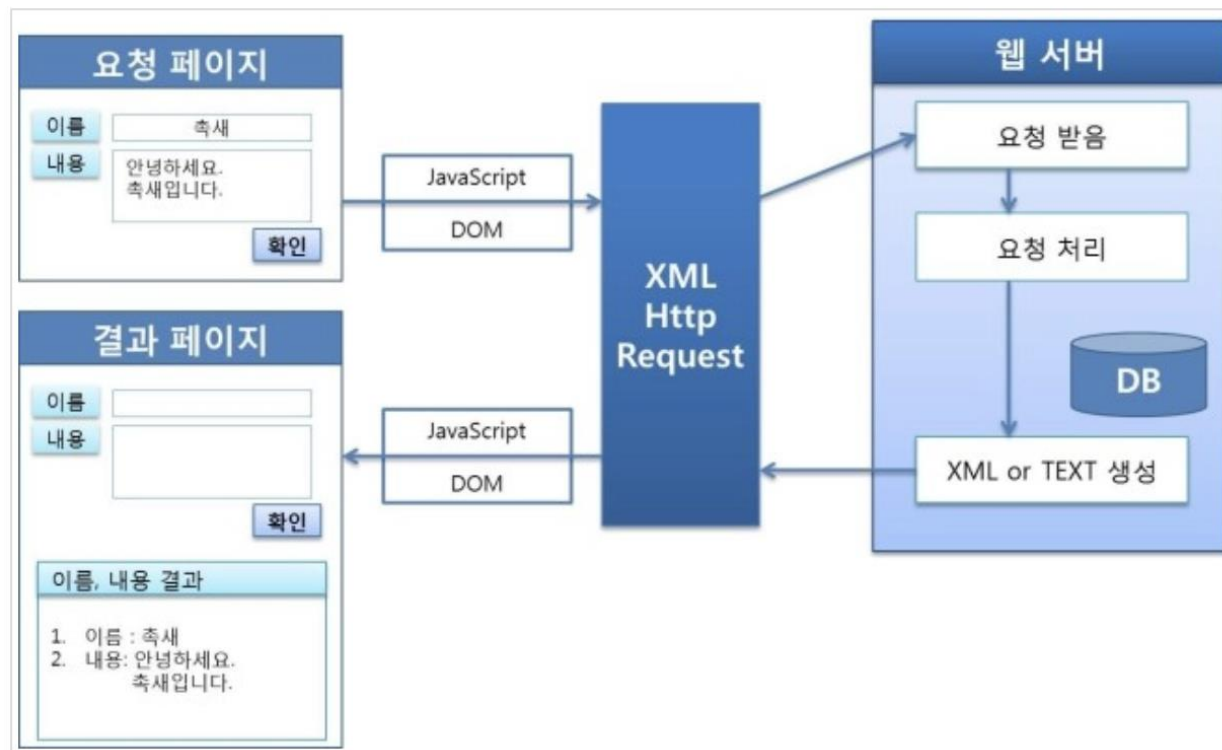
- REST는 HTTP 표준을 기반으로 구현하므로, HTTP를 지원하는 프로그램 언어로 클라이언트, 서버를 구현
- JSON 혹은 XML를 통해 데이터를 주고 받는 것이 일반적이다.

4. RESTful : 'REST API'를 제공하는 웹 서비스를 'RESTful'하다고 함

- 일반적으로 REST라는 아키텍처를 구현하는 웹 서비스를 나타내기 위해 사용되는 용어
- 이해하기 쉽고 사용하기 쉬운 REST API를 만드는 것이 목적

2_AJAX

- Ajax는 JavaScript의 라이브러리 중 하나이며 Asynchronous Javascript And Xml의 약자
- JavaScript를 사용한 비동기 통신, 클라이언트와 서버간에 데이터를 주고받는 기술
- 사용이유
 - 1: html 페이지 전체가 아닌 일부분만 갱신할수 있도록 XML HttpRequest객체를 통해 서버에 request를 해서 Json이나 xml형태로 필요한 데이터만 받아 갱신하므로 자원과 시간을 아낄 수 있음
 - 2: REST API를 호출하기 위해



2_AJAX 코드 형식

```
<script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
```

```
$.ajax({
```

호출 Method

```
type : "GET",
```

서버 호출 URL

```
url : "http://test.soxcorp.co.kr:31880/testapi",
```

서버로 전달할 데이터

```
data : { name: "soxcorp", age: 4 },
```

받는 데이터 형식

```
dataType : "text",
```

서버로 부터 데이터를
받으면 호출되는 함수
(이벤트 핸들러)

```
success : function (data) {  
    document.write(data);  
    console.log(JSON.parse(data));  
},
```

서버가 보낸 데이터는
data에 담긴다

통신 에러시 호출되는
함수
(이벤트 핸들러)

```
error : function () {  
    alert("통신실패!!!!");  
}
```

```
});
```


3_Node-Red 설치

1. Node JS 설치

<https://nodejs.org/ko/>

node --version && npm --version



Node.js®는 Chrome V8 JavaScript 엔진으로 빌드된 JavaScript 런타임입니다.

다운로드 - Windows (x64)

18.12.0 LTS

안정적, 신뢰도 높음

19.0.0 현재 버전

최신 기능

[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

[다른 운영 체제](#) | [변경사항](#) | [API 문서](#)

LTS 일정은 [여기서](#) 확인하세요.

2. Node-Red 설치 및 실행

npm install -g --unsafe-perm node-red

```
관리자: C:\Windows\system32\cmd.exe
C:\Users\mailt\Desktop> node --version && npm --version
v16.14.2
8.4.1
C:\Users\mailt\Desktop> node-red
'node-red'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.
C:\Users\mailt\Desktop> npm install -g --unsafe-perm node-red
added 292 packages, and audited 293 packages in 22s
38 packages are looking for funding
  run `npm fund` for details
5 low severity vulnerabilities
To address issues that do not require attention, run:
  npm audit fix
To address all issues (including breaking changes), run:
  npm audit fix --force
Run `npm audit` for details.
C:\Users\mailt\Desktop>
```

node-red

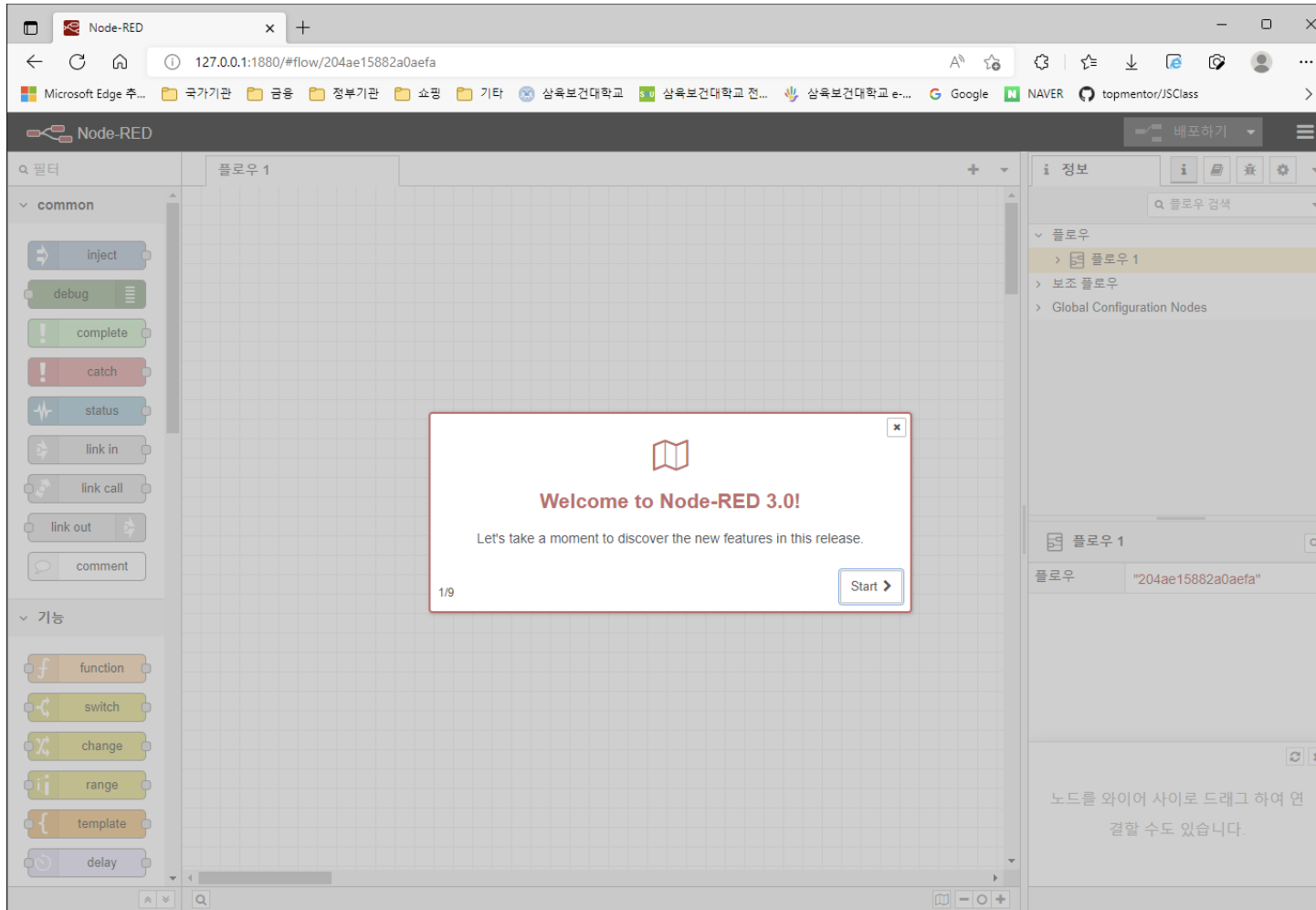
```
node-red
12 Sep 11:08:32 - [info] Server now running at http://127.0.0.1:1880/
12 Sep 11:08:32 - [warn] Encrypted credentials not found
12 Sep 11:08:32 - [info] Starting flows
12 Sep 11:08:32 - [info] Started flows
```



3_Node-Red 설치

3. 접속

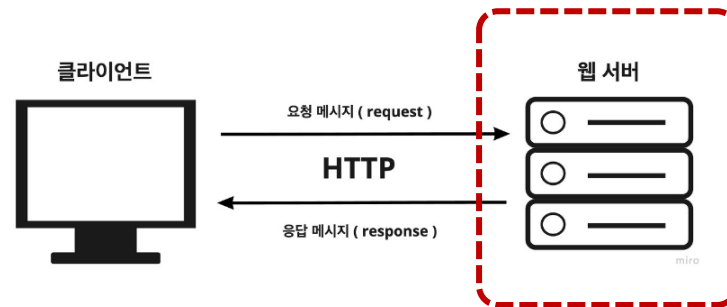
<http://127.0.0.1:1880>



3_Node-Red

Node-Red : IoT 애플리케이션을 쉽게 제작하기 위한 도구

- Node JS 환경상에서 동작
- 웹 환경에서 저작



1

사물인터넷 응용을 제작하는 비주얼 도구

2

간단한 런타임 배포, 시작품 제작에 적합

3

간단한 자동 실행 런타임을 쉽게 작성

4

다양한 연동을 간단하게 그려서 확장함

5

낮은 진입 장벽 : 누구나 쉽게 배우고 사용 가능

6

개방형 표준, 유연성, 공유

3_Node-Red

The screenshot displays the Node-RED web interface with three main panels and several annotations:

- 노드 팔레트 (Node Palette):** Located on the left, it contains two sections: **common** (inject, debug, complete, catch, status, link in, link call, link out, comment) and **기능 (Function)** (function, switch, change, range, template, delay, trigger, exec). The **inject** node is highlighted with a red box.
- 편집창 (Edit Canvas):** The central workspace for building flows, currently showing a grid.
- 정보창 (Info Panel):** Located on the right, it displays information about the selected flow, including the flow ID "204ae15882a0aefa".

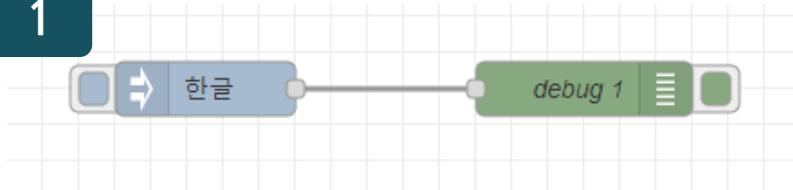
Annotations and interactions:

- A red dashed arrow points from the **inject** node in the palette to the text **항목 드래그-드랍** (Drag-and-drop item).
- A red box highlights the **노드 팔레트** and the **정보창** panels.
- A tooltip at the bottom right states: **alt-⌘를 사용하여 노드 팔레트를 관리 할 수 있습니다.** (You can manage the node palette using alt-⌘).

4_Node-Red Node 편집 방법

1. 노드 배치 + 노드 연결
2. 노드 속성 편집
3. '배포하기' 클릭
4. 디버그 창에서 확인
 - * API 확인 시에는 브라우저에서 확인

1



2

A screenshot of the Node-RED node configuration panel for the '한글' node. It shows three properties: 'msg. payload' set to 'Hello World', 'msg. topic' set to an empty string, and 'msg. nowtime' set to 'timestamp'. Each property has a small 'x' icon to its right.

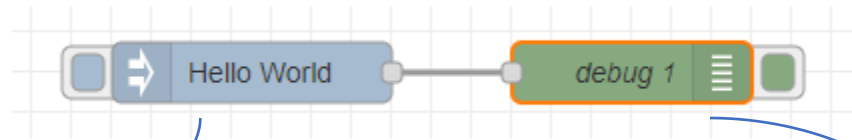
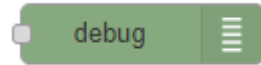
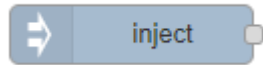
3

A screenshot of the Node-RED interface. At the top, there is a red button labeled '배포하기' (Deploy). Below it, the '디버그' (Debug) console is open, showing two messages. The first message is an object with properties: '_msgid', 'payload', 'topic', and 'nowtime'. The second message is a string '한글'. The console also shows the timestamp '2022. 9. 17. 오전 10:30:09' and 'node: debug 1'.

4

4_Node-Red Node

시작 노드와 출력 노드 : 노드 간 주요 데이터 전달은 msg.payload에 실어서 보내고 받는다



msg. payload =

msg. topic =

msg. nowtime =

속성

대상

출력대상 ☒ 디버그 창
☐ 시스템 콘솔
☐ 노드 상태(32자)

이름

속성

이름

msg. payload =

msg. topic =

string
number
boolean
JSON
buffer
timestamp
expression
env variable
msg.

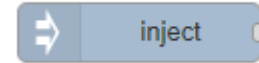
디버그

모든 플로우
all

2022. 9. 17. 오전 10:39:47 node: debug 1
msg : string[11]
"Hello World"

4_Node-Red Node

1. inject Node : 플로우 실행 (클릭 실행, 자동 (반복) 실행)



☐ Node-RED시작의 0.1 초 후, 아래를 시행

반복 지정한 시간간격

시간각격 1 초

☐ Node-RED시작의 0.1 초 후, 아래를 시행

반복 지정한 일시

시각 12:00

요일 ☒ 월요일 ☒ 화요일 ☒ 수요일
☒ 목요일 ☒ 금요일 ☒ 토요일
☒ 일요일

☐ Node-RED시작의 0.1 초 후, 아래를 시행

반복 지정한 시간간격, 일시

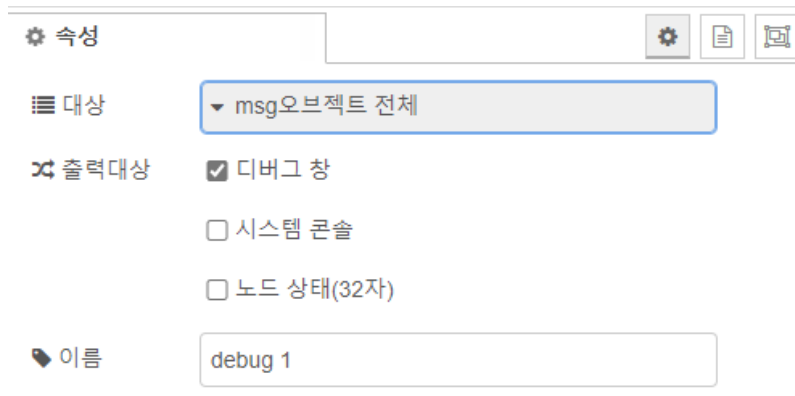
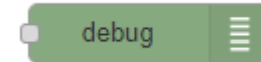
시간각격 1 분

시각 00:00 ~ 01:00

요일 ☒ 월요일 ☒ 화요일 ☒ 수요일
☒ 목요일 ☒ 금요일 ☒ 토요일
☒ 일요일

4_Node-Red Node

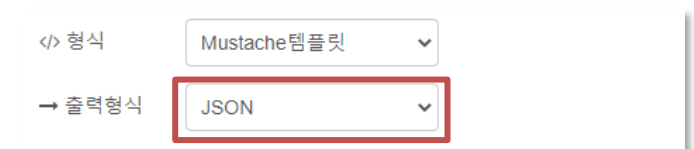
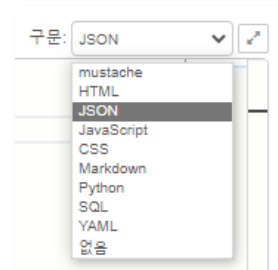
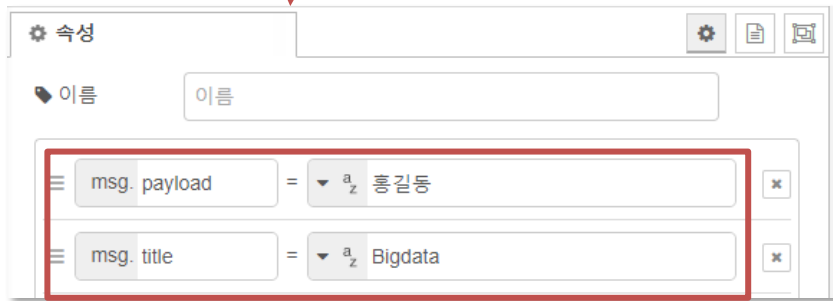
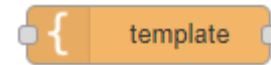
2. debug Node : 출력 노드 - 콘솔에 출력



4_Node-Red Node

3. template Node : 원하는 출력을 static 데이터로 출력

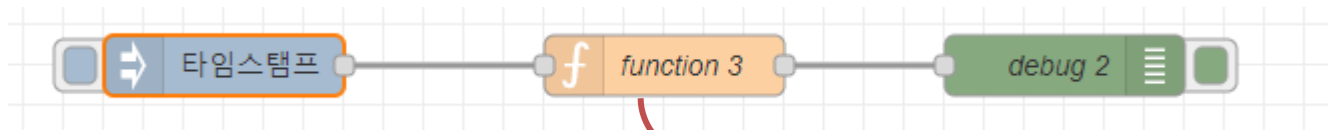
- 출력 데이터는 msg.payload에 실어서 다음 노드에 보냄
- 출력 형식 지정 가능 : 일반 텍스트, json 형식



4_Node-Red Node

4. function Node : 중간에 데이터를 조작할 때

- 이전 노드에서 전달 받은 데이터는 `msg.payload`에서 받음
- 다음 노드로 데이터 전달은 `msg.payload`에 실어서 리턴
 - * 처리 결과를 `msg.payload`에 담아야 함
- 항상 msg를 리턴해야 한다.



function의 노드 수정

삭제 취소 완료

속성

이름 function 1

Setup On Start 코드 On Stop

```
1 var timestr = new Date(msg.payload).toLocaleString("ko-KR");
2
3 msg.payload = timestr;
4
5 return msg;
```

응용 예제

실습-1

실습 문제 : 다음의 REST API를 호출하는 Ajax 프로그램을 만드시오

<http://make-random.com/MakeRandom/api/userInfo.get>

실습-2

실습 문제 : inject 노드에서 2개의 숫자를 받고 function 노드에서 두 수를 더한 결과를 debug 노드에서 출력하도록 노드를 구성하시오.



inject의 노드 수정

삭제 취소 완료

속성

이름 이름

msg. payload = {"num1":100,"num2":200}

msg. topic = a_z