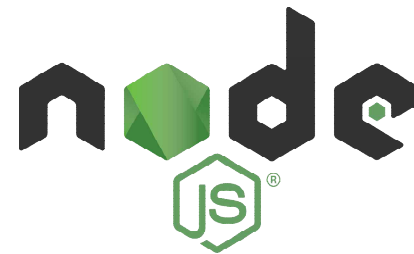


네트워크 프로그래밍

웹(Node.js) 기반의 소켓통신프로그래밍

- 2018-11-13
- 김 태 완



Node.js?

<https://ko.wikipedia.org/wiki/Node.js>



- 구글 크롬의 V8 Engine JavaScript 엔진
- JavaScript Runtime
- 확장성 있는 네트워크 어플리케이션
- Server-Side 플랫폼
- 뛰어난 확장성(Node)
- 가볍고, 효율적
- 라이선스 (MIT)
- 멀티 플랫폼
(Windows, Linux, MacOS , etc)

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

<https://velopert.com/node-js-tutorials>

Node.js?

<https://ko.wikipedia.org/wiki/Node.js>



- Speed (run on google JS Engine)
- Asynchronous I/O
- Non-Blocking I/O
- Data Streaming
(HTTP request / response as a single event)
- Real-Time Application
(client / server side)
- Easy / Fast 코딩
- Open-Source (NPM)

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

<https://velopert.com/node-js-tutorials>

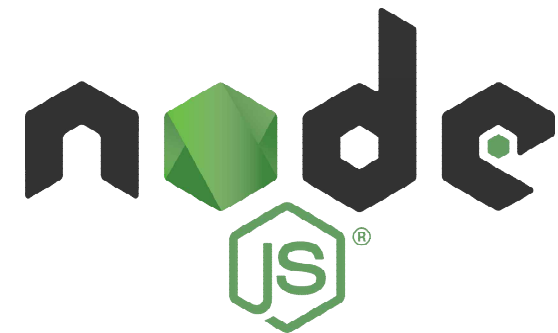
Node.js?

<https://ko.wikipedia.org/wiki/Node.js>



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

<https://velopert.com/node-js-tutorials>



Node.js?

<https://ko.wikipedia.org/wiki/Node.js>

A screenshot of a PayPal advertisement for the Business Debit MasterCard. The background is a dark wood-grain texture. On the right side, there is a photograph of the actual card, which is silver with a blue stripe and the PayPal logo. The card is placed on a grey laptop. The text on the left side of the ad includes the title "PayPal Business Debit MasterCard®", a list of benefits, a "Get a Card Today" button, and two footnotes.

PayPal Business Debit MasterCard®

- Withdraw money from your PayPal balance at any ATM.
- No annual fee.
- Earn unlimited 1% cash back on eligible purchases.[†]
- MasterCard Zero Liability policy for unauthorized purchases.*

Get a Card Today

[†] Additional enrollment for Cash Back is required. Earn Cash Back when you use your card online, over the phone or sign up for a purchase. Cash Back will be earned on net purchases when the receipt is signed or when the purchase is authorized as a credit transaction by phone or online. Purchases made using a PIN are not eligible to earn Cash Back.

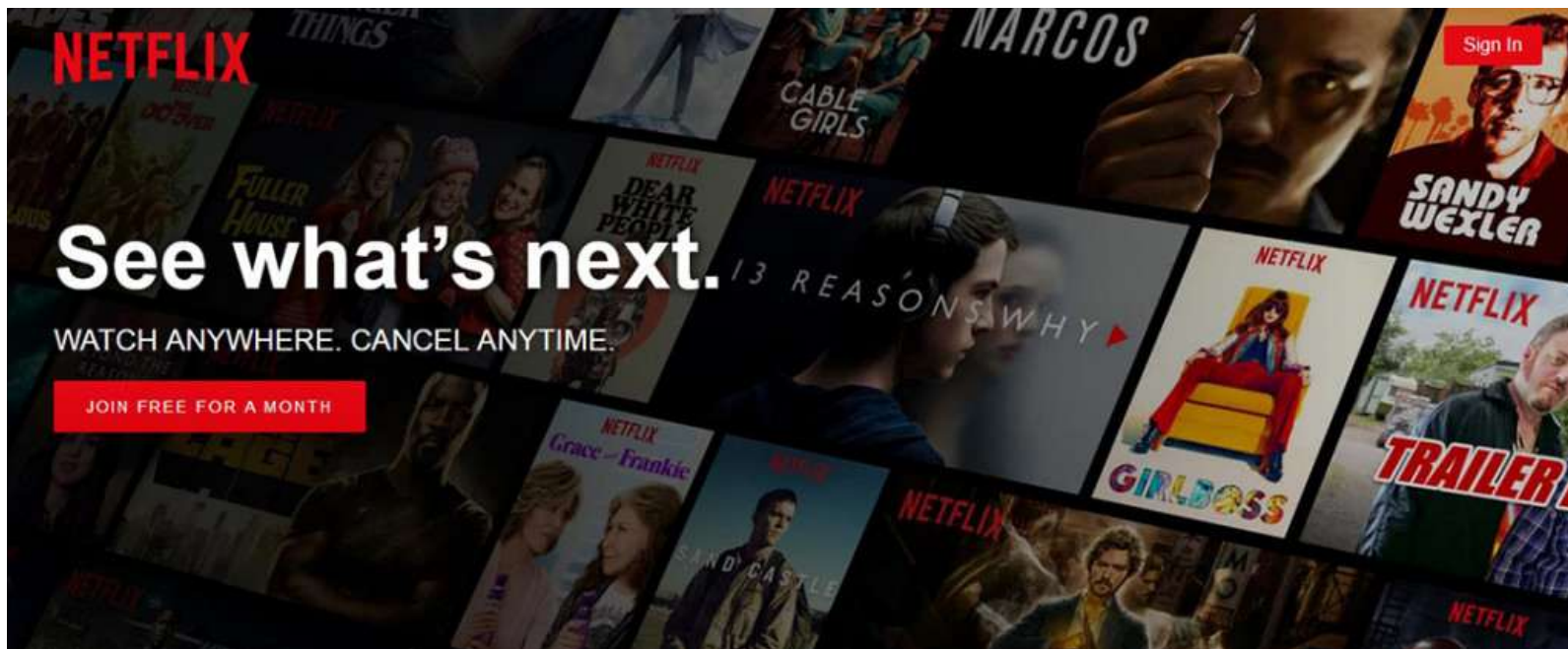
* Conditions and exceptions apply, see **Cardholder Agreement**.

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



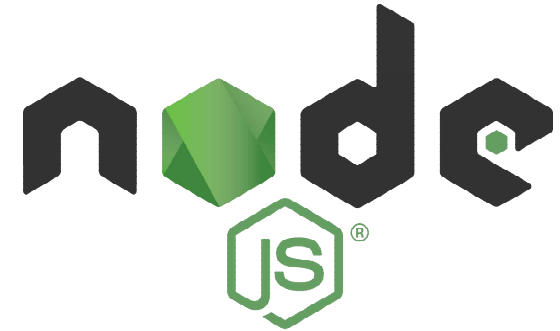
Node.js?

<https://ko.wikipedia.org/wiki/Node.js>



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

2018-11-19



Node.js?

<https://ko.wikipedia.org/wiki/Node.js>

Uber 드라이버 파트너 라이더 비즈니스 더 보기 로그인 가입

원하는 대로 이동하세요.

드라이버 파트너


원하는 시간에 자유롭게 운행하세요.
Uber 앱이 주변의 운행 기회를 찾아 드립니다.
[자세히 알아보기](#)

드라이버 파트너로 가입하기 →

라이더

버튼 하나로 원하는 곳까지 간편하게 이동하세요.
[자세히 알아보기](#)

라이더로 가입하기 →



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

2018-11-19



Node.js?

<https://ko.wikipedia.org/wiki/Node.js>

A screenshot of the LinkedIn homepage with a sign-up form overlay. The form is titled "최고의 분야 전문가되기" (Become the best in your field) and "LinkedIn에서 가능합니다. 무료로 가입하세요." (Possible on LinkedIn. Join for free). It contains fields for "성" (Last name), "이름" (First name), "이메일 주소" (Email address), and "비밀번호(6자 이상)" (Password, 6+ characters). Below the fields is a checkbox for terms and conditions, and a blue "회원 가입" (Join) button. The background shows a grid of diverse professional portraits. At the top of the page are fields for "이메일 주소" and "비밀번호" with a "로그인" (Login) button. At the bottom are fields for "동료 찾기" (Find colleagues) with "성" and "이름" fields, and a "검색" (Search) button.

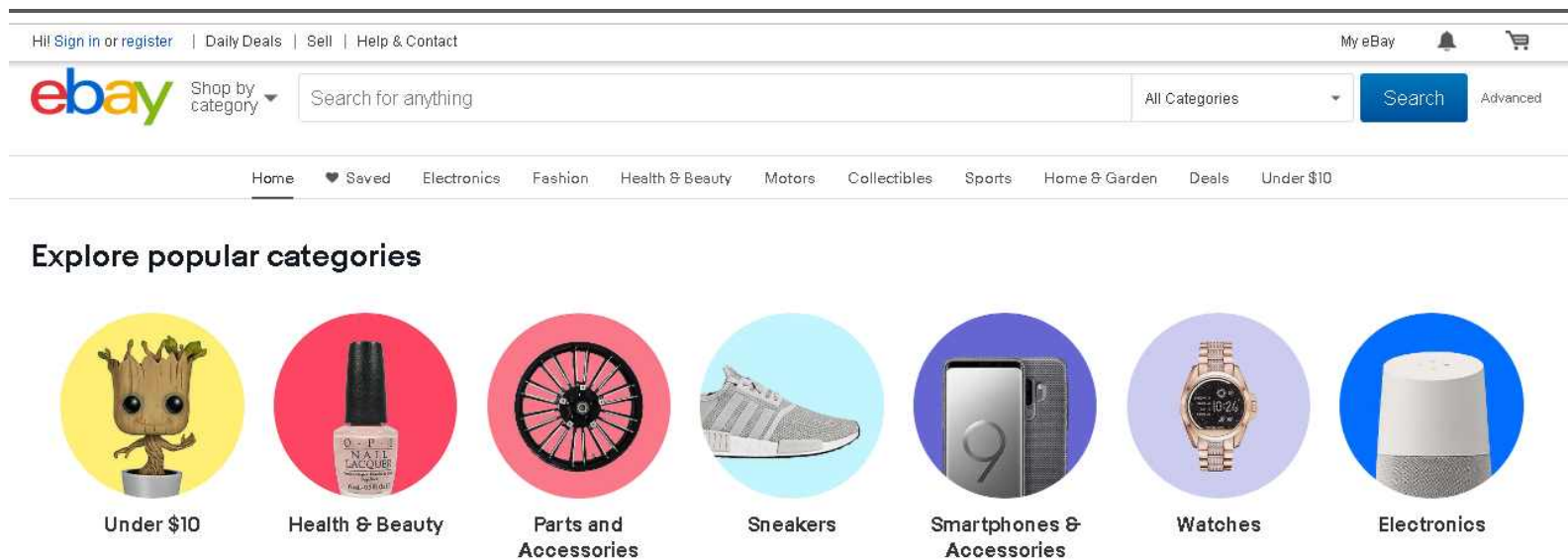
<https://nodejs.org/ko/>

2018-11-19



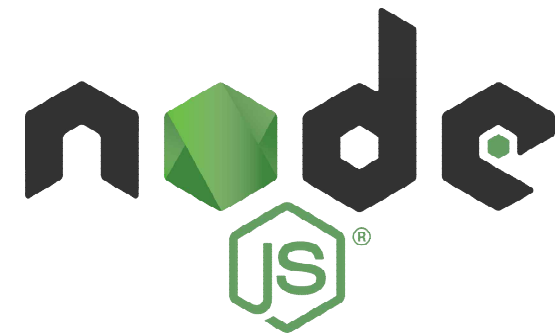
Node.js?

<https://ko.wikipedia.org/wiki/Node.js>



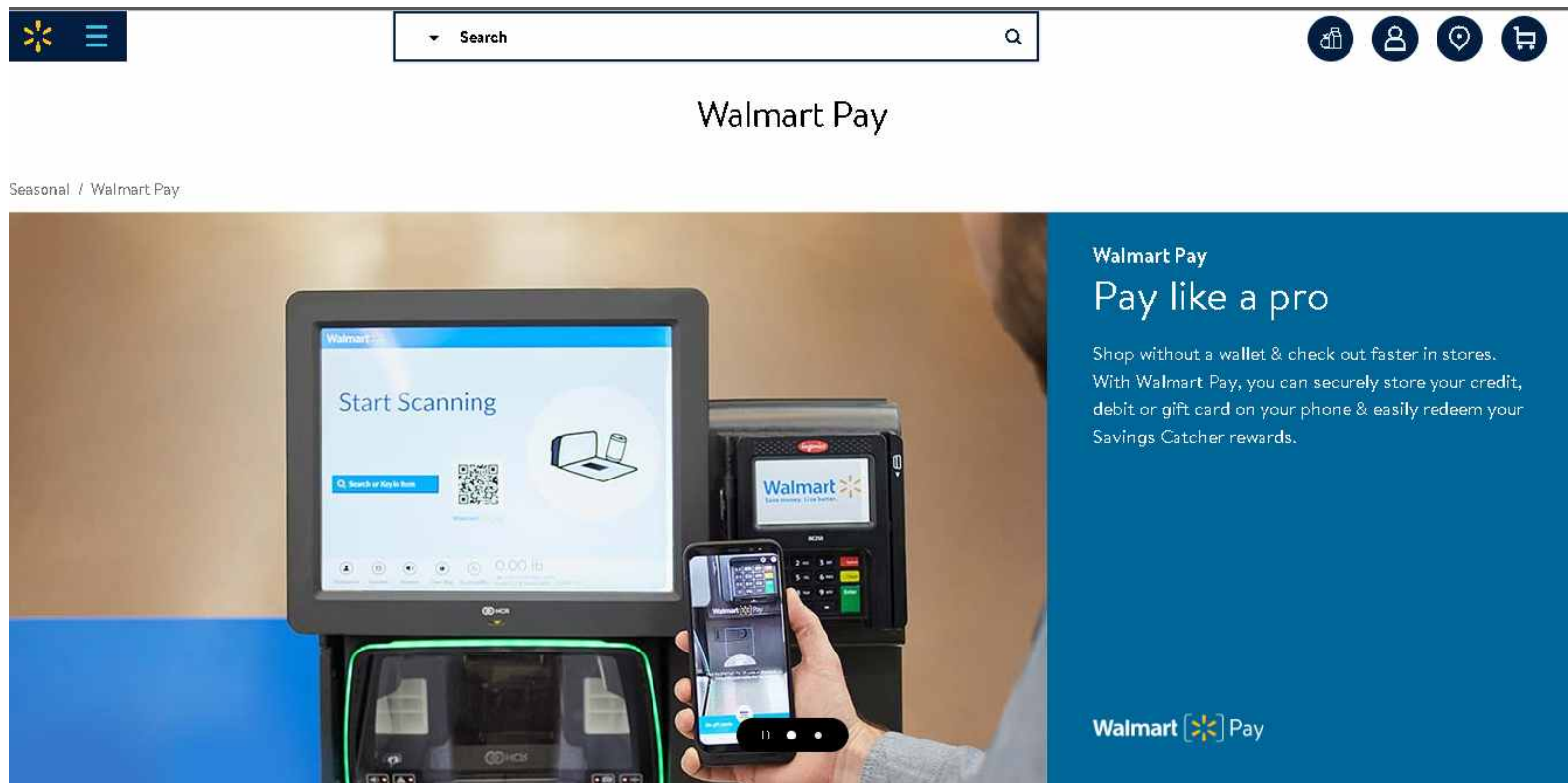
<https://nodejs.org/ko/>

2018-11-19



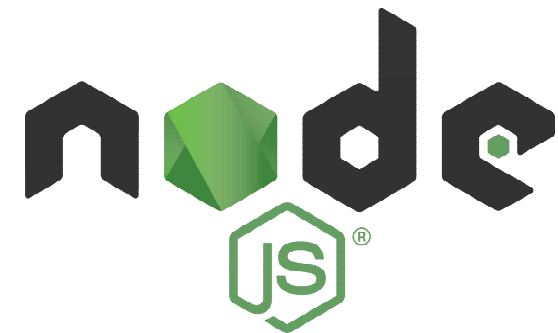
Node.js?

<https://ko.wikipedia.org/wiki/Node.js>



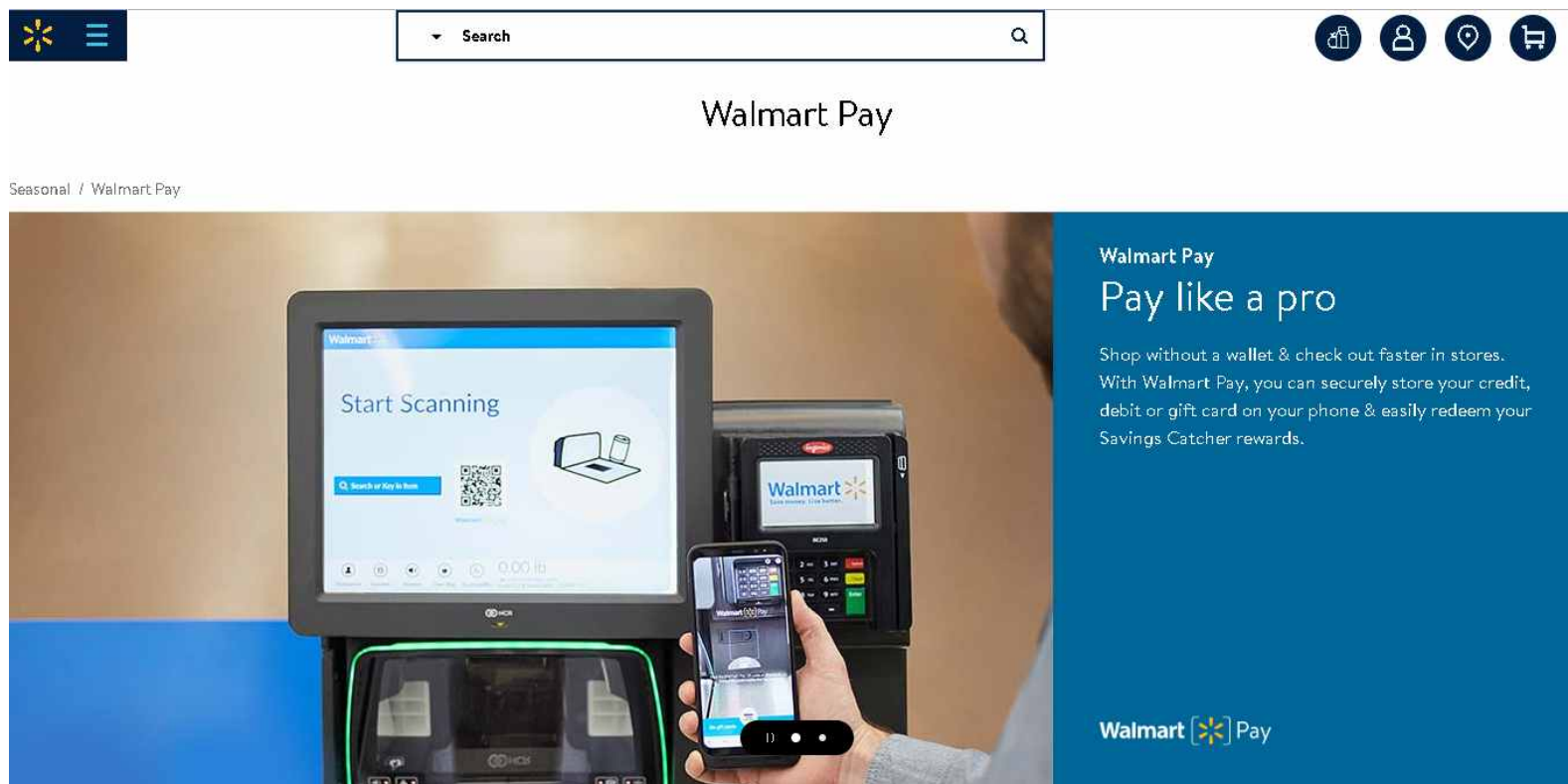
<https://nodejs.org/ko/>

2018-11-19



Node.js?

<https://ko.wikipedia.org/wiki/Node.js>

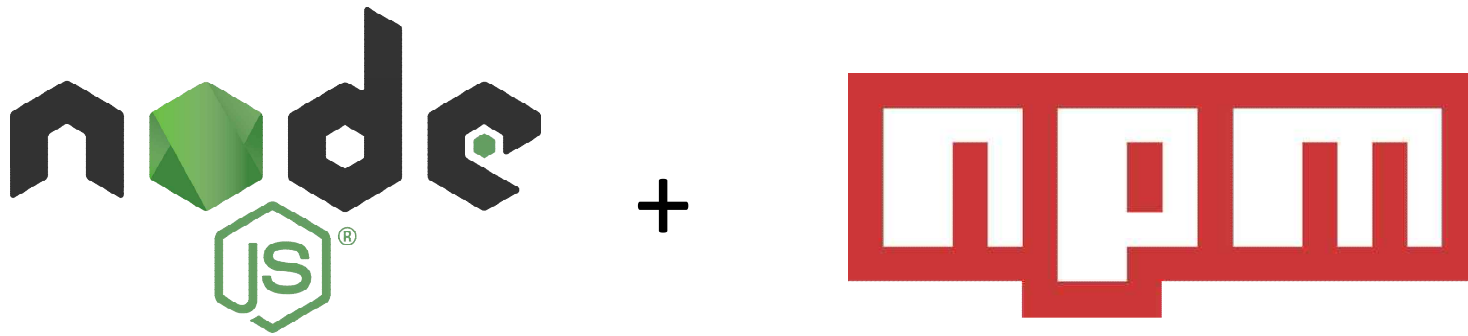


<https://nodejs.org/ko/>

2018-11-19

Node.js Open Source Library

<https://www.npmjs.com/>



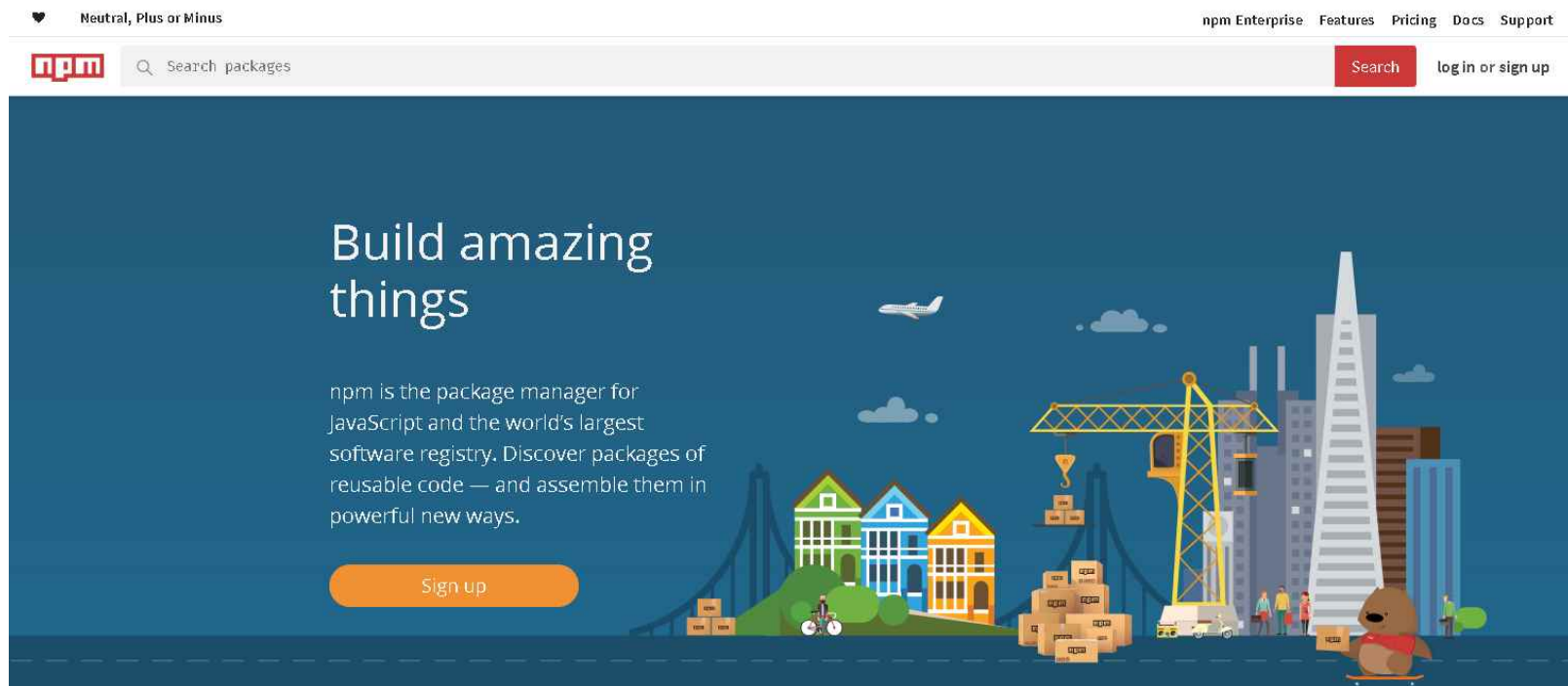
Open Source Library?

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



Node.js Open Source Library

<https://www.npmjs.com/>



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



♥ Nifty Pun Master

npm Enterprise Features Pricing Docs Support

npm

socket

Search

log in or sign up

4655 packages found

123...233»

Sort Packages

Optimal

Popularity

Quality

Maintenance

Who's Hiring?

Triplebyte, MuleSoft, Callisto and lots of other companies are hiring javascript developers.

See all 19 companies

socket

Socket is a connect clone for simple socket based applications

mountainmansoftware published 0.0.1 • 6 years ago

socket.io

Node.js realtime framework server

realtime framework websocket tcp events socket io

darrachequesne published 2.1.1 • 6 months ago

ws

Simple to use, blazing fast and thoroughly tested websocket client and server for Node.js

HyBi Push RFC-6455 WebSocket WebSockets real-time

lpinca published 6.1.0 • a month ago

socket.io-client

[Build Status](https://secure.travis-ci.org/socketio/socket.io-client.svg?branch=master)[http://travis-ci.org/socketio/socket.io-client]

[Dependency Status](https://david-dm.org/socketio/socket.io-client.svg)[https://david-dm.org/socketio/socket.io-

realtime framework websocket tcp events client

darrachequesne published 2.1.1 • 6 months ago

sockjs-client

SockJS-client is a browser JavaScript library that provides a WebSocket-like object.

[Search](#)[log in or sign up](#)

Share your code. npm Orgs help your team discover, share, and reuse code. [Create a free org »](#)

WS

6.1.0 • [Public](#) • Published a month ago

[Readme](#)[1 Dependencies](#)[4,450 Dependents](#)[99 Versions](#)

ws: a Node.js WebSocket library

npm v6.1.0 build passing build passing coverage 100%

ws is a simple to use, blazing fast, and thoroughly tested WebSocket client and server implementation.

Passes the quite extensive Autobahn test suite: [server](#), [client](#).

Note: This module does not work in the browser. The client in the docs is a reference to a backend with the role of a client in the WebSocket communication. Browser clients must use the native `WebSocket` object. To make the same code work seamlessly on Node.js and the browser, you can use one of the many wrappers available on npm, like [isomorphic-ws](#).

Table of Contents

- [Protocol support](#)
- [Installing](#)
 - [Opt-in for performance and spec compliance](#)
- [API docs](#)

install

```
> npm i ws
```

± weekly downloads

444,351



version

6.1.0

license

MIT

open issues

13

pull requests

1

homepage

[github.com](#)

repository

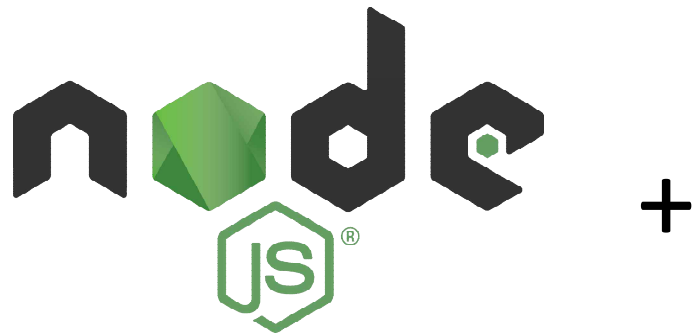
[github](#)

last publish

a month ago

collaborators

Node.js?



+



Desktop Application?

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



Node.js + Electron



Skype



GitHub Desktop



Figma



Flow



Discord



WordPress.com



Beaker Browser



Hyper



Kap



Now Desktop



Visual Studio Code



Kitematic



Insomnia



Sygus



Simplenote



Collectie



Caret



JIBO

<https://github.com>
<https://github.com>

2018-1



GitKraken



Ghost



WebTorrent



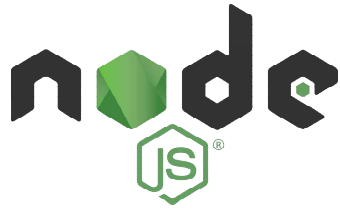
1Clipboard



Slack




Atom



Node.js + Electron

<https://electronjs.org/apps>

 ELECTRON

후원자 앱 문서 블로그 커뮤니티 출시 검색 한국어

Filter apps by name, description, etc...

카테고리들

Productivity 198

Developer Tools 139

Utilities 108

Music 44

Social Networking 40

Photo & Video 40

Games 29

Finance 23

Business 20

Education 16

Graphics & Design 11

Entertainment 6

Books 5


News 4


Lifestyle 4


Medical 2


Travel 1


Health & Fitness 1


 **Dusk Player** Minimal music player for your desktop.


 **RIDE** Remote IDE for Dyalog APL.


 **Superscript** A text editor for comic book writers.


 **P3X Redis UI** Redis UI based on redis-commander and phpRedisAdmin.


 **EasyHand** Helpful tool of making easier to work with huge quantity of files, folders and IDEs in the same...


 **Tagstoo** Software to tag folders and files, with multimedia and ebooks preview.


 **MoviePrint** With MoviePrint you can create screenshots of entire movies in an instant.

 **Altair GraphQL Client** Beautiful feature-rich GraphQL Client for all platforms.

 **Raven Reader** Simple RSS Reader for desktop without any distraction.

 **Genotify** Light-weight, cross-species gene lookup and summarization for desktop.

 **tiny-timer** Simple timer desktop app.

 **Excel Parser Processor** A Cross-Platform Desktop App for processing all rows of excel files.

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

2018-11-19

Node.js 준비물



1. Node.js 설치하기

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



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

다운로드 - Windows (x64)

10.13.0 LTS

안정적, 신뢰도 높음

11.1.0 현재 버전

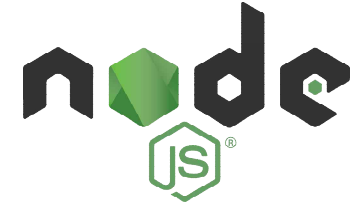
최신 기능

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

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

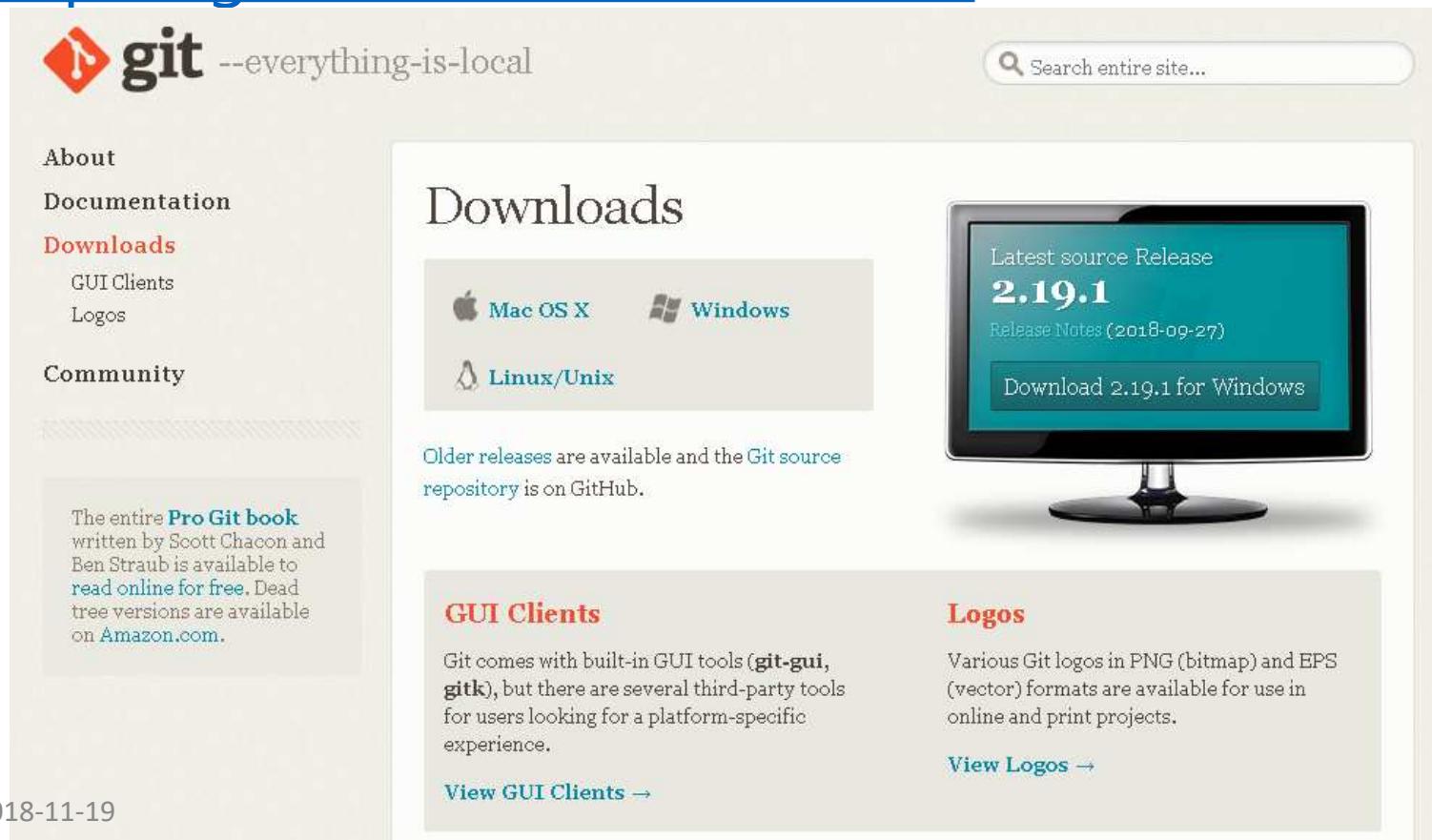
LTS 일정은 [여기서](#) 확인하세요.

Node.js 준비물

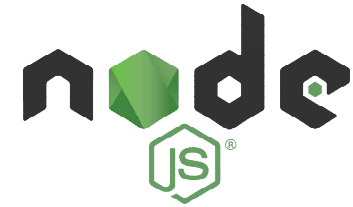


2. git 설치하기

<https://git-scm.com/downloads>

A screenshot of the Git website's Downloads page. The page has a light beige background. At the top left is the Git logo (a red diamond with a white 'G') and the text "git --everything-is-local". To the right is a search bar with the placeholder text "Search entire site...". On the left side, there is a navigation menu with links: "About", "Documentation", "Downloads" (highlighted in red), "GUI Clients", "Logos", and "Community". Below the menu is a box containing text about the "Pro Git book". The main content area is titled "Downloads" in a large, dark font. Below the title, there are three boxes for different operating systems: "Mac OS X" (with an Apple logo), "Windows" (with a Windows logo), and "Linux/Unix" (with a Linux logo). To the right of these boxes is a large image of a computer monitor displaying the text "Latest source Release 2.19.1" and "Release Notes (2018-09-27)", with a button that says "Download 2.19.1 for Windows". Below the monitor image, there is a paragraph stating "Older releases are available and the Git source repository is on GitHub." At the bottom, there are two columns: "GUI Clients" with a paragraph about built-in and third-party tools, and "Logos" with a paragraph about available logo formats. Both columns have a "View" link with an arrow.

Node.js 준비물



3. Visual Studio Code 설치하기

<https://code.visualstudio.com/>

The image shows the Visual Studio Code website and a screenshot of the code editor. The website header includes links for Visual Studio Code, Docs, Updates, Blog, Community, Extensions, and FAQ. A search bar and a Download button are also present. A banner below the header states: "Version 1.28 is now available! Read about the new features and fixes from September." The main content area features the text "Code editing. Redefined." and "Free. Open source. Runs everywhere." Below this is a "Download for Windows" button with a dropdown arrow, and a link to "Other platforms and Insiders Edition". A disclaimer at the bottom of the website states: "By using VS Code, you agree to its license and privacy statement." The date "2018-11-19" is displayed at the bottom left. The screenshot of the code editor shows the "EXTENSIONS" sidebar on the left with a list of popular extensions: C# 1.22 (356K stars), Python 0.11.2 (211K stars), Debugger for Chrome (148 stars), C/C++ 0.7.0 (143K stars), Go 0.6.39 (99K stars), and ESLint 0.10.0 (88K stars). The main editor area displays a TypeScript file named "app.ts" with the following code:

```
1 import app from './app';
2 import debugModule = require('debug');
3 import http = require('http');
4
5 const debug = debugModule('node-express-typescript:server');
6
7 // Get port from environment and store in Express.
8 const port = normalizePort(process.env.PORT || '3000');
9 app.set('port', port);
10
11 // create
12 const server = app.listen(port);
13
14 server.on('error', (error) => {
15   if (error.syscall !== 'listen') {
16     throw error;
17   }
18   const bind = typeof port === 'string'
19     ? ` ${port} `
20     : ` ${port} `;
21   const err = ` ${bind} address already in use
22     `;
23   console.error(err);
24   process.exit(1);
25 });
26
27 // Normal
28 function normalizePort(val: any): number|string|boolean {
29   let port = parseInt(val, 10);
30   if (isNaN(port)) {
31     return false;
32   }
33   if (port < 0) {
34     return false;
35   }
36   if (port > 65535) {
37     return false;
38   }
39   return port;
40 }
```

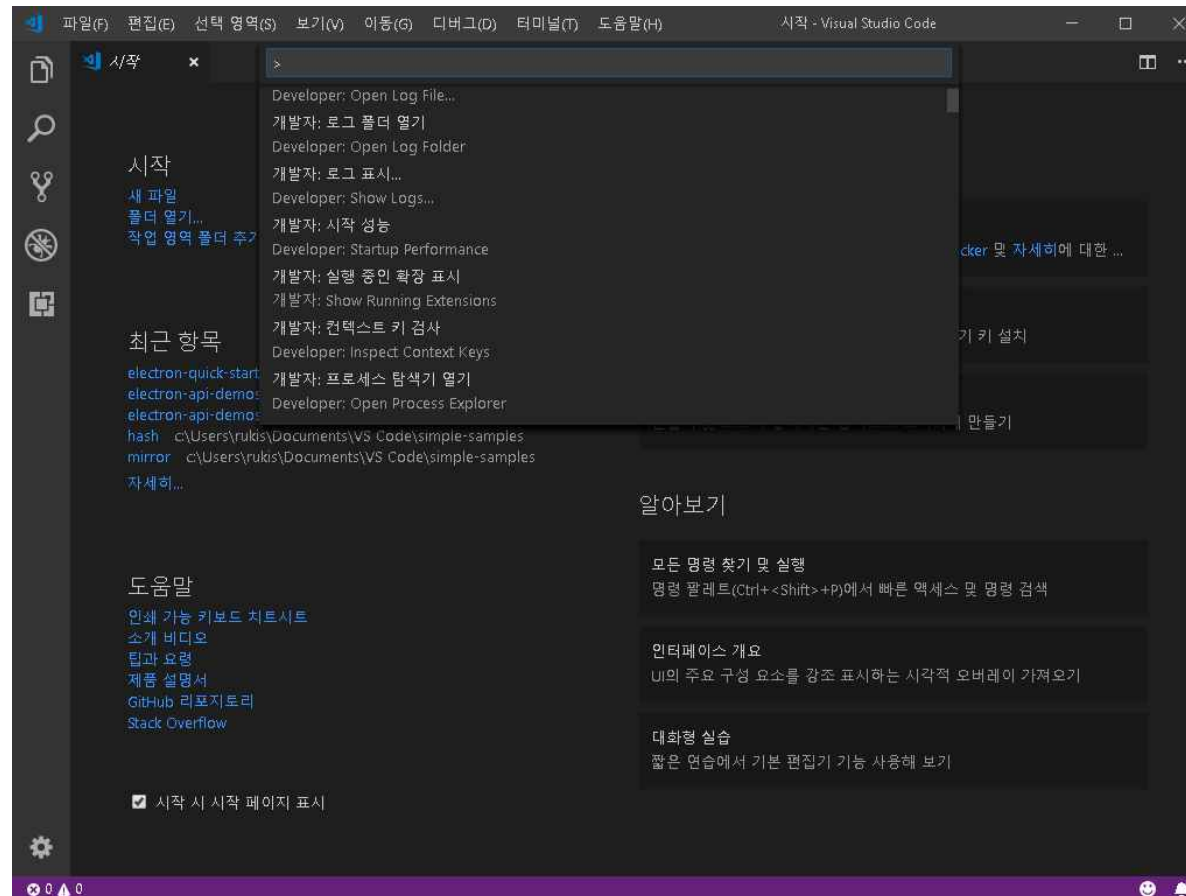
The status bar at the bottom of the code editor shows "Ln 9, Col 21", "Spaces: 2", "UTF-8", "LF", and "TypeScript".

Node.js 시작하기



Visual Studio Code

➤ Ctrl + Shift + p / F1 (명령팔레트)

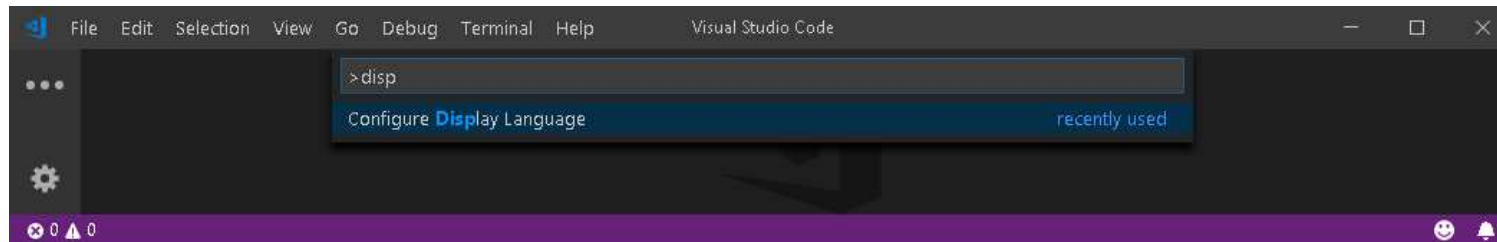


Node.js 시작하기



Visual Studio Code

➤ 한글화 -> F1 -> Configure Display Language



```
{ "locale": "en" }  
1 {  
2 // Defines VS Code's display language.  
3 // See https://go.microsoft.com/fwlink/?LinkId=761051 for a list of supported locales.  
4 "locale": "en" // Changes will not take effect until VS Code has been restarted.  
5 }
```



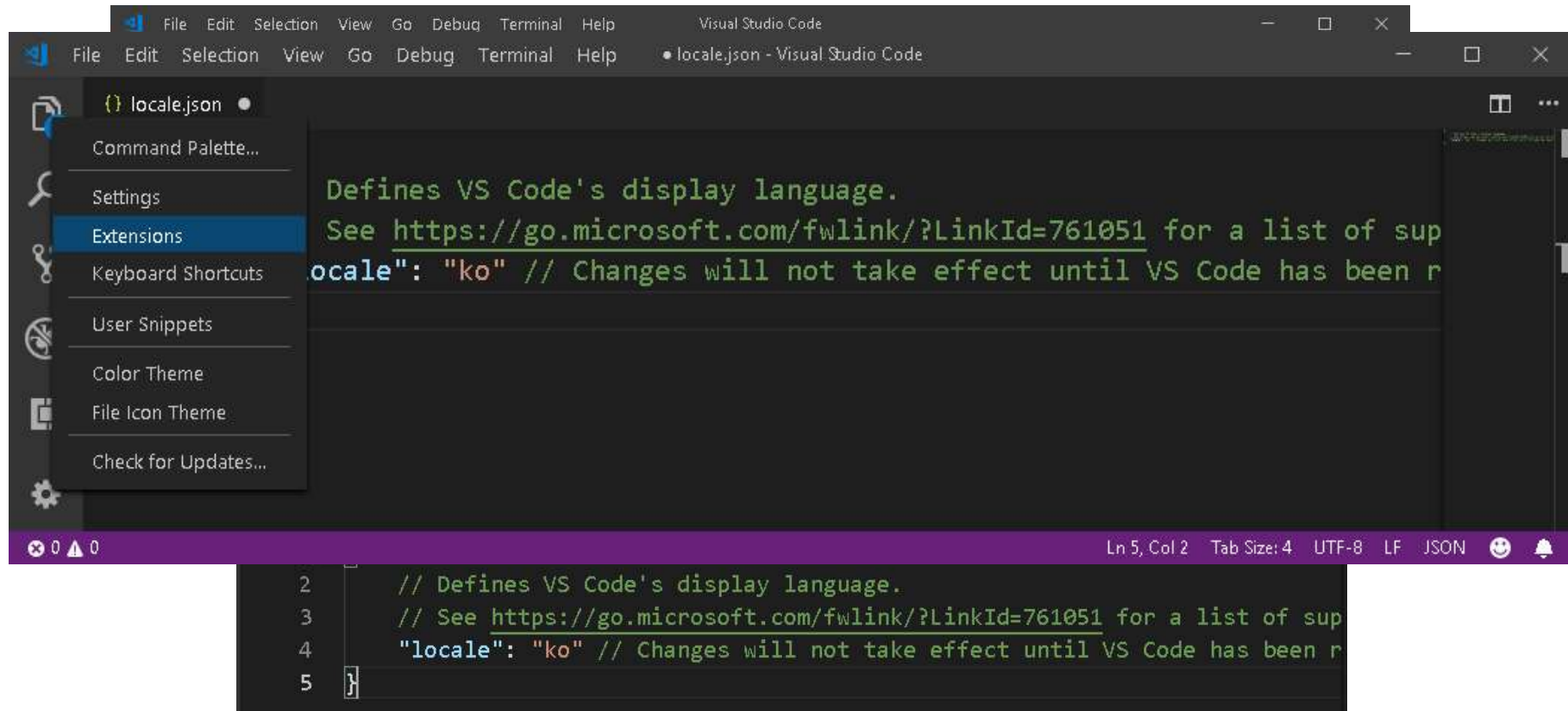
```
{ "locale": "ko" }  
1 {  
2 // Defines VS Code's display language.  
3 // See https://go.microsoft.com/fwlink/?LinkId=761051 for a list of supported locales.  
4 "locale": "ko" // Changes will not take effect until VS Code has been restarted.  
5 }
```

Node.js 시작하기

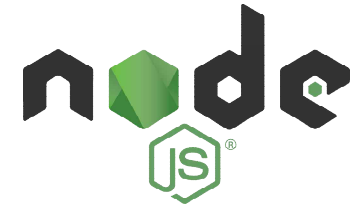


Visual Studio Code

➤ 한글화 -> F1 -> Configure Display Language

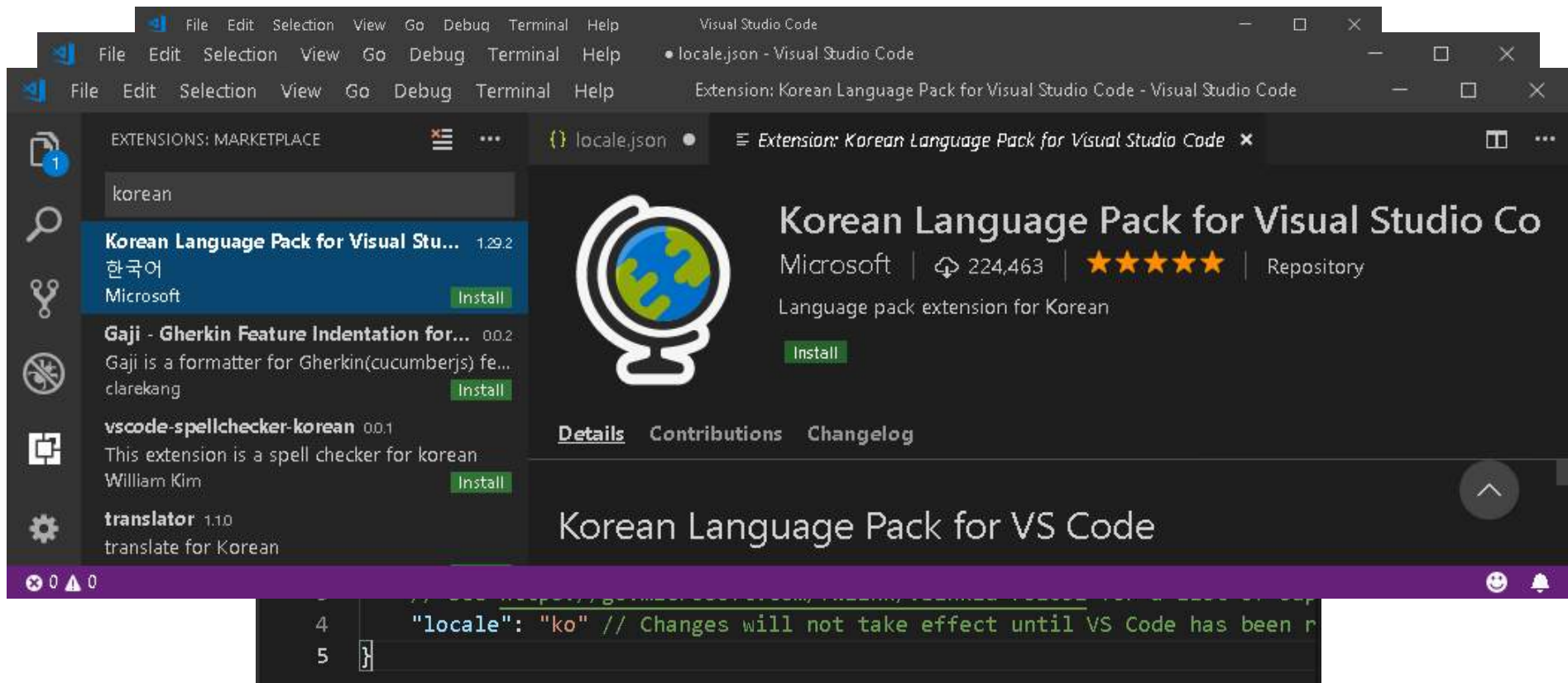


Node.js 시작하기

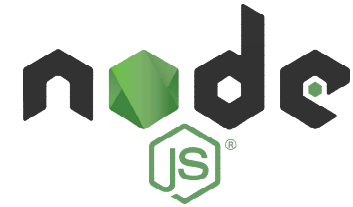


Visual Studio Code

➤ 한글화 -> F1 -> Configure Display Language



Node.js 시작하기



Visual Studio Code

➤ 한글화 -> F1 -> Configure Display Language



Node.js 시작하기



▶ HELLO NODE JS

A screenshot of the Visual Studio Code editor interface. The top menu bar shows options like File (F), Edit (E), Select (S), View (V), Go (G), Debug (D), Terminal (T), and Help (H). The left sidebar contains icons for Explorer, Search, Source Control, and Run and Debug. The Explorer view shows a project named "NETWORKPROGRAMMINGSTUDY" with files "StartNode.js" and "main.js". The main editor area shows the content of "main.js":

```
1 var msg = 'Hello Node.js';  
2 console.log(msg);
```

The bottom panel shows the "Debug Console" with the following output:

```
C:\Program Files\nodejs\node.exe --inspect-brk=5889 StartNode.js\main.js  
Debugger listening on ws://127.0.0.1:5889/a6dd7a74-082d-4e80-89eb-cb1ae86d3598  
For help, see: https://nodejs.org/en/docs/inspector  
Hello Node.js
```

The status bar at the bottom indicates the current file is "main.js:2", the encoding is "UTF-8", and the line ending is "CRLF".

Node.js 프로젝트 생성



➤ Express 사용하기

➤ 파일 -> 폴더 열기 -> 선택

➤ `npm install -g express-generator`

```
문제  출력  디버그 콘솔  터미널
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\rukis\Documents\NetworkProgrammingStudy> npm install -g express-generator
```

➤ `express -e ./`

```
문제  출력  디버그 콘솔  터미널
PS C:\Users\rukis\Documents\NetworkProgrammingStudy> express -e ./프로젝트명 혹은 폴더명
```


Node.js 프로젝트 생성



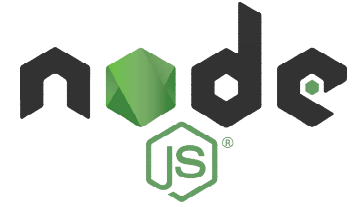
➤ Express 프로젝트 생성

```
12  /* Get port from environment and store in Express.
13  */
14
15  var port = normalizePort(process.env.PORT || '3000');
16  app.set('port', port);
17
18  /**
19  * Create HTTP server.
20  */
21
22  var server = http.createServer(app);
23
24  /**
25  * Listen on provided port, on all network interfaces.
```

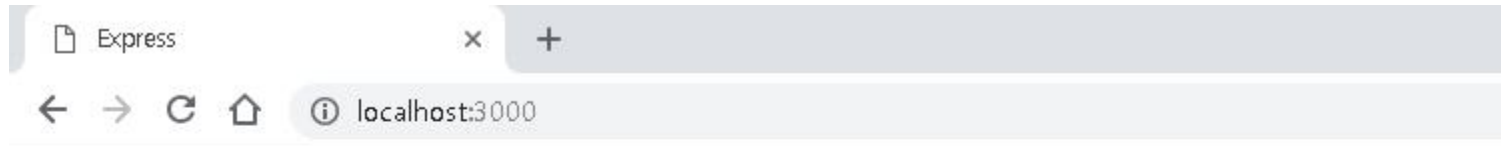
➤ F1 -> npm install or 터미널 -> npm install

```
>npm install
npm: Install Dependencies
최근에 사용한 항목
```


Node.js 시작하기



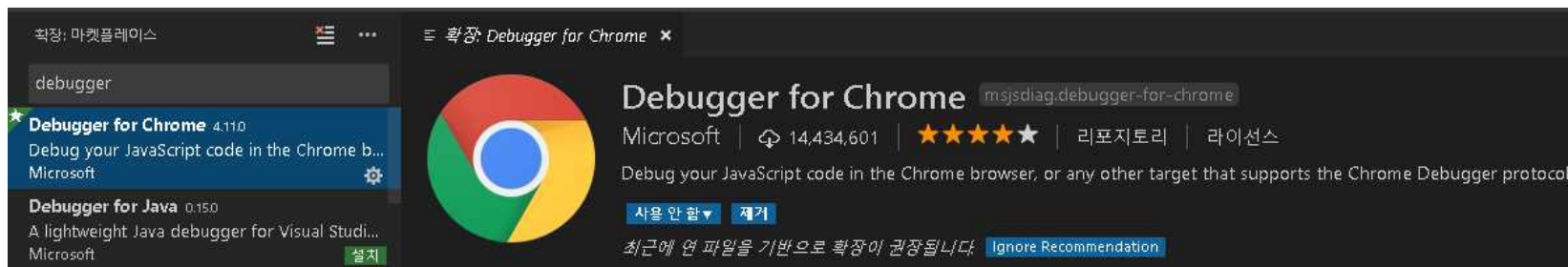
- Express 프로젝트 생성
- 빌드 -> F5, 웹 접속 (localhost:3000)



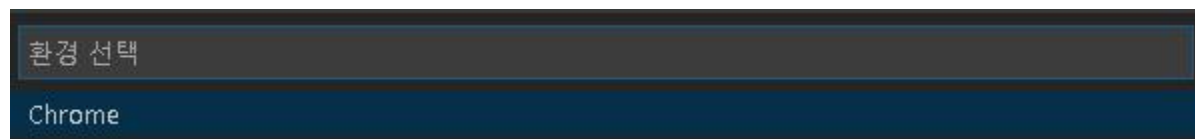
Node.js 시작하기



- Express 프로젝트 (Debugging)
- Debugger for Chrome 확장 설치



- 빌드 > F5 > Chrome



Node.js 시작하기



- Express 프로젝트 (Debugging)
- .vscode/launch.json 파일 자동생성

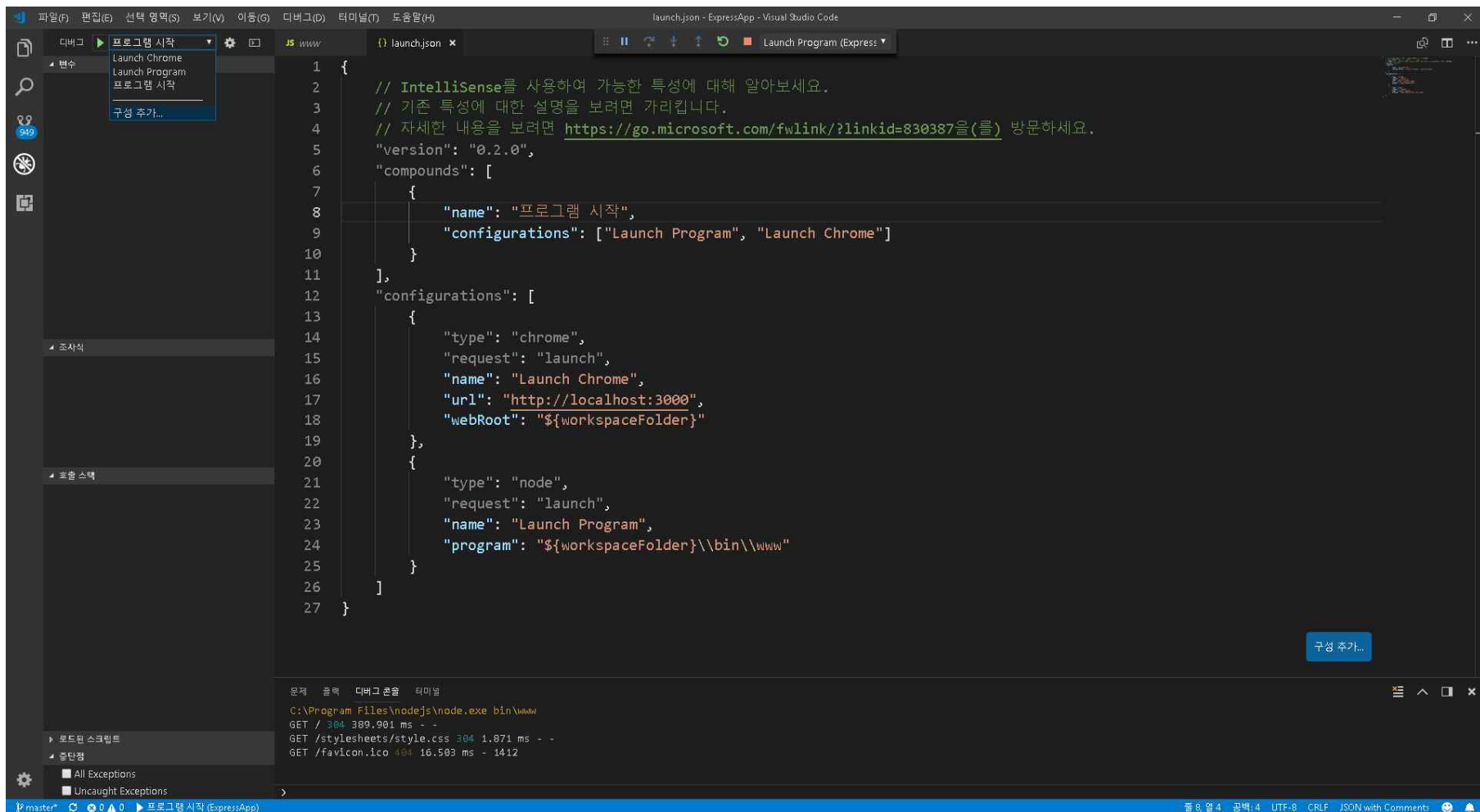
A screenshot of the Visual Studio Code editor interface. The left sidebar shows the Explorer view with a file tree for a project named "EXPRESSAPP". The file tree includes folders like ".vscode", "bin", "node_modules", "public", "routes", "views" and files like "index.js", "users.js", "app.js", "package-lock.json", and "package.json". The main editor area shows the "launch.json" file, which is a JSON configuration for debugging. The file contains comments in Korean and a configuration for launching Chrome against localhost. The JSON content is as follows:

```
{
  // IntelliSense를 사용하여 가능한 특성에 대해 알아보세요.
  // 기존 특성에 대한 설명을 보려면 가리킵니다.
  // 자세한 내용을 보려면 https://go.microsoft.com/fwlink?linkID=2147297
  "version": "0.2.0",
  "configurations": [
    {
      "type": "chrome",
      "request": "launch",
      "name": "Launch Chrome against localhost",
      "url": "http://localhost:8080",
      "webRoot": "${workspaceFolder}"
    }
  ]
}
```

Node.js 시작하기



➤ Express 프로젝트 (Debugging)



Node.js + Express + Html



➤ Jade vs Ejs

<http://jade-lang.com/>

<https://ejs.co/>

➤ Jade 의 경우 기능이 더 뛰어나다.

➤ Ejs 의 경우 html 코드를 그대로 사용하여 이질감이 없다.

➤ html 파일 사용하기

➤ views/index.ejs => views/index.html 파일명 수정

➤ app.js 코드 수정

```
// view engine setup
app.set('views', path.join(__dirname, 'views'));
/// ***** html 사용을 위한 ejs module 추가
App.engine('html', require('ejs').renderFile);
App.set('view engine', 'html');
```

Node.js + Express + Html



➤ routes/index.js

<https://www.w3schools.com/html/>

<https://ejs.co/>

```
res.render('index', { title: 'test' });
```

➤ views/index.html

```
<html> <head>
<title><%=title%></title> </head>
<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body> </html>
```

Node.js 데이터 입력



➤ routes/index.js

<https://www.w3schools.com/html/>

```
// POST
router.post('/', function (req, res) {
  var returnVal = { result: '' };
  if (req.body.msg) {
    console.log("client message : " + req.body.msg);
    returnVal.result= 'success';
  }
  else {
    returnVal.result= 'failure';
  }
  res.json(returnVal);
});
```


Node.js 데이터 입력



➤ javacripts/test.js

<https://www.w3schools.com/html/>
<https://api.jquery.com/>

```
function test()
{
    var data = { msg:
        $("#name").val()
    };
    console.log(data);
    $("#result").text(data.msg);
    $.ajax({
        type: 'POST',
        url: '/',
        data: data,
        contentType:
            'application/json',
        async: true,
        success: function (data) {
```

```
        var res = data;
        if (res.result == 'success') {
            console.log("success!!");
        } else if (res.result ==
            'failure') {
            console.log("failure!!");
        } else {
            alert(JSON.stringify(res));
        }
        $("#result").text(res.result)
    });
}
```

Node.js 데이터 입력



➤ views/index.html

<https://www.w3schools.com/html/>

```
<head>
<title><%=title%></title>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.3.1.min.js">
<script type="text/javascript" src="/javascripts/test.js">/>
</head>
```

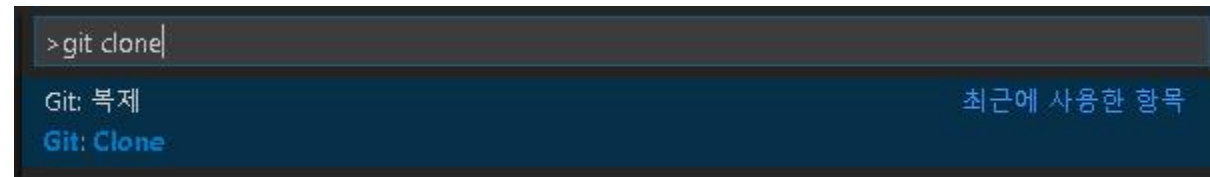
```
<body>
  User Name: <input type="text" id="name" name="name" value="test">
  <br/>
  <button onclick="test()">input</button>
  <br/>
  result: <label id="result"></label>
</body>
```

Node.js 로그인 처리



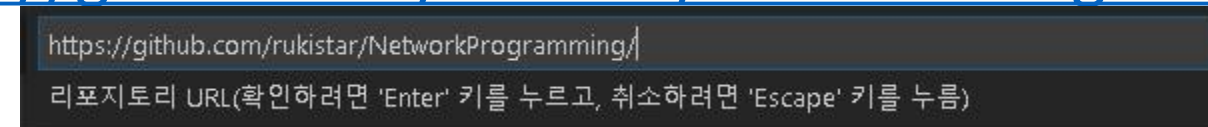
➤ F1 -> Git Clone

<https://www.w3schools.com/html/>

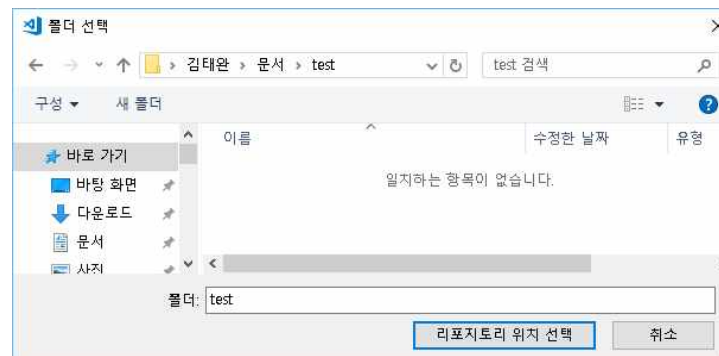


➤ 주소 입력 :

<https://github.com/rukistar/NetworkProgramming/>



➤ 폴더 지정



Node.js 로그인 처리



➤ 파일 -> 폴더 열기

<https://github.com/rukistar/NetworkProgramming/>

The image shows the VS Code interface. On the left, the '파일(F)' menu is open, with '폴더 열기...' (Open Folder...) selected. The main editor area shows the 'LoginExpress' folder in the Explorer view. The folder structure is as follows:

이름	수정한 날짜	유형
.vscode	2018-11-18 오후...	파일 폴더
bin	2018-11-18 오후...	파일 폴더
public	2018-11-18 오후...	파일 폴더
routes	2018-11-18 오후...	파일 폴더
views	2018-11-18 오후...	파일 폴더

The '폴더: LoginExpress' field is visible at the bottom of the Explorer view, and the '폴더 선택' (Select Folder) button is highlighted.

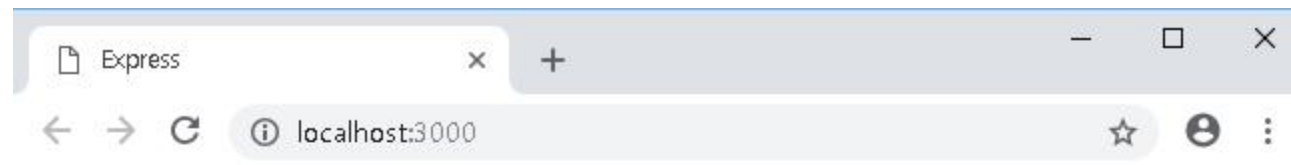
Node.js 로그인 처리



➤ index.html

/LoginExpress

<https://www.w3schools.com/w3css>



Express

Welcome to Express

[로그인 페이지](#)

Node.js 로그인 처리



➤ login.html

/LoginExpress

<https://www.w3schools.com/w3css>

A screenshot of a web browser window. The title bar shows a tab labeled "로그인" (Login). The address bar shows "localhost:3000/Login". The page content features a login form with a background image of a snowy landscape under a green aurora borealis. The form includes a "Username" label, a text input field with placeholder text "Enter Username", a "Password" label, a text input field with placeholder text "Enter Password", a green "Login" button, and a "Remember me" checkbox which is checked.

로그인

← → ↻ ⓘ localhost:3000/Login ☆ ⓘ ⋮

Username

Enter Username

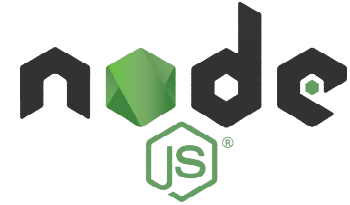
Password

Enter Password

Login

☒ Remember me

Node.js 소켓 프로그래밍



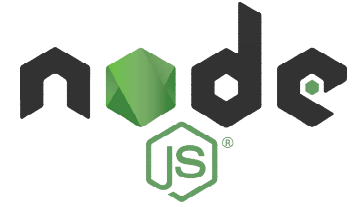
➤ Websocket vs Socket.io

<https://www.npmjs.com/package/ws>

➤ Websocket

- http Web은 http프로토콜을 이용 요청 / 응답 동작
- Web에서 TCP/IP Socket 처럼 실시간 통신 할 수 없음
- 그래서 등장 WebSocket(ws) 프로토콜
- Websocket을 이용하면 웹브라우저
- 실시간 데이터 주고 받을 수 있음
- 최근 대부분 브라우저에서 Websocket지원
- InternetExplorer 의 경우 version 10부터 지원

Node.js 소켓 프로그래밍



➤ Websocket vs Socket.io

<https://www.npmjs.com/package/ws>

➤ Websocket

- `npm install -g express-generator`
- `express -e ./`
- `npm install`
- `npm install ws`

Node.js 소켓 프로그래밍



➤ Websocket vs Socket.io

<https://socket.io>

➤ Socket.io

- Node.js 기반 OpenSource Library
- WebSocket처럼 실시간 데이터 처리 가능
- 멀티 디바이스 (Web, android, iOS, Windows, 등)
- WebSocket이 지원되지 않는 브라우저도 가능

Node.js 소켓 프로그래밍



➤ Socket.io

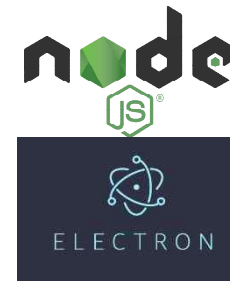
<https://socket.io>

➤ Server

➤ npm install socket.io

➤ Client

➤ npm install socket.io-client



Node.js 응용프로그래밍

➤ Electron

- <https://electronjs.org/docs/tutorial/first-app>
- <https://github.com/electron/electron-quick-start>
- <https://github.com/electron/electron-api-demos>

Node.js 응용프로그래밍

➤ Electron – Socket.io, Chat

- Git clone <https://github.com/electron/electron-quick-start>
- npm install
- npm install jquery
- npm install socket.io
- npm install socket.io-client
- /socket/server.js
- index.html 수정

Node.js 응용프로그래밍

➤ Electron – Socket.io, Chat

➤ /socket/server.js

```
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);
var port = process.env.PORT || 3000;
app.get('/', function(req, res){
  res.sendFile(__dirname + '/index.html');
});
io.on('connection', function(socket){
  socket.on('chat message', function(msg){
    io.emit('chat message', msg);
  });
});
http.listen(port, function(){
  console.log('listening on *:' + port);
});
```

Node.js 응용프로그래밍

➤ Electron – Socket.io, Chat

➤ index.html 수정

```
<body>
<ul id="messages"></ul>
<form action="">
<input id="m" autocomplete="off" /><button>Send</button>
</form>
<script>
window.$ = window.jQuery = require('jquery');
require('./renderer.js')
</script>
</body>
```


Node.js 응용프로그래밍

➤ Electron – Socket.io, Chat

➤ ./renderer.js

```
$(function () {  
  var socket = require('socket.io-client')  
  ('http://localhost:3000');  
  // var socket = io();  
  $('form').submit(function () {  
    socket.emit('chat message', $('#m').val());  
    $('#m').val('');  
    return false;  
  });  
  socket.on('chat message', function (msg) {  
    $('#messages').append($('- ').text(msg));  
    window.scrollTo(0, document.body.scrollHeight);  
  });  
});

```

팀 프로젝트

- 소켓 통신
- 다중 디바이스 (옵션*)
(Electron / Android / Arduino / Raspberry PI 등등...)
- 아이디어 / 기획 / 기능 정의
- 시스템 설계
- 화면 구성 (사용자 인터페이스)
- 구현

참조

- Html
 - <https://www.w3schools.com/html/default.asp>
- Javascript
 - <https://www.w3schools.com/js/default.asp> (javascript)
 - <https://www.w3schools.com/jquery/default.asp> (jquery)
- CSS
 - <https://www.w3schools.com/css/default.asp> (css)
 - <https://www.w3schools.com/w3css/default.asp> (w3.css library)
- Node.js
 - <https://nodejs.org/dist/latest-v10.x/docs/api/> (node.js document)
 - <https://velopert.com/category/dev-log/tech-log/nodejs> (blog)
 - <http://bcho.tistory.com/tag/node.js> (blog)

참조

- Socket.io
 - <https://socket.io/>
- Electron
 - <https://electronjs.org/docs> (electron document)
 - <https://electronjs.org/apps> (electron 참조)
 - <https://electronjs.org/docs/tutorial/first-app> (electron tutorial)
- npm
 - <https://www.npmjs.com/> (node.js library)