Node.js

What is node.js

특징

- 비동기 입출력 방식 (Non-blocking) I/O
- 이벤트 기반 입출력 방식 (Event Loop)
- Single Thread
- 모듈

개발도구

- 편집기 VSCode
  - Extentions
    - ES6 Code Snippets
    - ESLint
    - Prettier - Code formatter
    - Live Server
- 크롬 브라우저
- 노드

Install

brew 설치

```
$ /usr/bin/ruby -e "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

node 설치

```
$ brew install node
```

Vue CLI 설치

```
$ npm install -g @vue/cli
or
$ yarn global add @vue/cli
or
$ yarn dlx @vue/clie

$ vue --version # 확인
$ npm list -g --depth=0  # npm 설치 리스트 확인
```

Vue CLI 삭제

```
$ npm uninstall -g vue-cli
```

vue 프로젝트 생성

```
$ vue create <프로젝트 이름>

$ cd <프로젝트 이름>
$ npm run serve
```

vuetify 패키지 추가

```
$ vue add vuetify
```

vue-router 설치

```
$ vue add router
```

vuex 설치

```
$ vue add vuex
```

axios 설치

```
$ vue add axios
```

Install Node.js on Ubuntu

- Install Node.js on Ubuntu

```
# to install node.js we need CURL
$ sudo apt install -y curl

# install node.js using apt package manager
$ sudo apt update & sudo apt upgrade -y
$ sudo apt install nodejs
$ nodejs -v
$ sudo apt install npm
$ npm -v

# using PPA
$ curl -sL https://deb.nodesource.com/setup_18.x -o nodesource_setup.sh
# version 18.x
$ sudo bash nodesource_setup.sh
$ sudo apt-get install nodejs
$ sudo apt-get install build-essential
```

- nvm

```
$ curl -o-
https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh | bash
$ source ~/.bashrc
$ nvm list-remote
$ nvm install <version>
$ nvm list
$ nvm use <version>
```

- npm update

```
$ npm install -g npm@latest
$ npm -v
```

Modules

- https/http2 모듈
- express, express-session
- multer
- morgan
- static
- cookie/session
- sequlize
- mongoose
- passport
- jwt token
- test; unit → test coverage → integration test → load test →
- web socket; ws, socket.io,
- commander, inquirer
- redis, memcache
- cross-env
- pm2
- sanitize-html, csurf
- winston
- helmet, hpp
- connect-redis
- template engines; pug, nunjucks

typescript

1. 디렉토리 생성 (mkdir)
2. 프로젝트 생성 (npm init)
3. 타입스크립트 모듈 설치
4. tsconfig.json 생성
5. 타입 추가 (npm i @types/node)
6. ts-node 설치 (npm i -g ts-node)

- [Node.js] Node.js에서 Typescript 사용하기 / ts-node, @types/node

가상환경

- nodeenv
- nvm-windows NVM for Windows @github.com
  - C:\Users\%username%\AppData\Roaming\nvm\setting.txt; make sure ANSI to UTF-8

```
> nvm ls  # 설치된 Node.js 확인
> nvm install <version> [arch]  # Node.js 설치
> nvm use <version>  # Node.js 버전 변경
```

- node 버전 관리기 nodenv 를 설치해 봅시다.
- nodeenv를 활용한 프로젝트별 node 가상 환경 관리
- NVM으로 Windows 환경에서 Node 버전 관리하기

JavaScript Basics

- 변수 선언
  - var
  - let
  - const
- Arrow Function
- Array
  - sort()
  - filter()
  - map()
  - reduce()
- Template Literals; back tick `
- Spread Operator; 배열에서 iteration 형태의 데이터 요소를 하나하나로 모두 분해해서 사용
- Object Destructuring; { , } = object
- Array Destructuring; [,] = array
- Default Function Parameter; same as c++, define default value in argument
- Rest Parameter; ...args
- Promise; 비동기 처리에 사용 new Promise((resolve, reject) ⇒ {});
- Async/Await; 비동기
- Regular Expression
  - exec()
  - test()
  - match()
  - search()
  - replace()
  - split()

Start

```
$ node <file>
```

Node.js 내장 모듈과 객체

- Console
  - console.log(내용, ...args)
  - console.error(...)
  - console.table(테이블형 데이터)
  - console.time(레이블) / console.timeEnd(레이블)
  - console.dir(오브젝트, 옵션)
- Timers
  - setTimeout(콜백함수, 밀리초)
  - setInterval(콜백함수, 밀리초)
  - setImmediate(콜백함수)
- Process
  - listeners
    - beforeExit
    - exit
    - disconnect
    - message
  - process.env
  - process.nextTick
  - process.exit()
- OS
- Path
  - path.basename(path[,ext])
  - path.delimiter
  - path.dirname(path)
  - path.extname(path)
  - path.format(pathOjbject)
  - path.isAbsolute(path)
  - path.join([...paths])
  - path.parse(path)
  - path.sep
- URL;
  - WHATWG API
  - url.parse()
- Crypto
- File system
  - fs.readFile(path, [options], callback)
  - fs.readFileSync(path, [options])
  - fs.writeFile(path, data, [options], callback)
  - writeFileSync(path, data [options])
  - fs.watchFile(filename[, options], listener)

json-server

```
$ npm install -g json-server
```

```
$ json-server --watch <file>
```

express

```
$ npm install express
```

- middlewares
  - body-parser
  - compression
  - connect-rid
  - cookie-parser
  - cors
  - csurf
  - errorhandler
  - method-override
  - morgan
  - multer
  - response-time
  - serve-favicon
  - serve-index
  - serve-static
  - express-session
  - connect-timeout
  - vhost

Useful

- winston
- nodemailer
- node-cron
- xlsx
- xlsx-js-style
- ics
- socket.io
- axios
- cheerio
- forever
- pm2

References

- Node Version Manager

book toc

01 | 노드에 대해 알아보고 개발 도구 설치하기

노드란 무엇일까? 노드의 비동기 입출력 방식 노드에서 구현하는 이벤트 기반 입출력 방식 노드를 더 쉽게 사용할 수 있게 하는 모듈 개발 도구 설치하기 설치할 프로그램 목록 미리 확인하기 브라켓 설치하기 크롬 브라우저 설치하기 브라켓 기본 사용 방법 노드 설치하기

Getting Started

02 | 노드 간단하게 살펴보기

첫 번째 노드 프로젝트 만들기 자바스크립트 파일 만들어 실행하기 브라켓의 확장 기능 설치하고 브라켓에서 노드 프로그램 실행하기 노드 셸에서 직접 코드 입력하고 실행하기 콘솔에 로그 뿌리기 프로세스 객체 간단하게 살펴보기 노드에서 모듈 사용하기 더하기 함수를 모듈로 간단히 분리하기 module.exports로 메인 파일에 더하기 함수 호출하기 외장 모듈 사용하기 간단한 내장 모듈 사용하기 시스템 정보를 알려 주는 os 모듈 파일 패스를 다루는 path 모듈

JavaScript on node.js

03 | 노드의 자바스크립트와 친해지기

자바스크립트의 객체와 함수 이해하기 변수로 자료형 알아보기 자바스크립트의 함수 배열 이해하기 배열의 모든 요소 하나씩 확인하기 배열에 값 추가 및 삭제하기 splice( ) 메소드로 배열 요소 여러 개를 한꺼번에 추가하거나 삭제하기 slice( ) 메소드로 배열 일부 요소 복사하여 새로운 배열 만들기 콜백 함수 이해하기 함수를 호출했을 때 또 다른 함수를 파라미터로 전달하는 방법 함수 안에서 값을 반환할 때 새로운 함수를 만들어 반환하는 방법 프로토타입 객체 만들기

Node.js basics

04 | 노드의 기본 기능 알아보기

주소 문자열과 요청 파라미터 다루기 주소 문자열을 URL 객체로 변환하기 요청 파라미터 확인하기 이벤트 이해하기 이벤트 보내고 받기 계산기 객체를 모듈로 만들어 보기 파일 다루기 파일을 읽어 들이거나 파일에 쓰기 파일을 직접 열고 닫으면서 읽거나 쓰기 버퍼 객체 사용하는 방법 알아보기 스트림 단위로 파일 읽고 쓰기 http 모듈로 요청받은 파일 내용을 읽고 응답하기 fs 모듈로 새 디렉터리 만들고 삭제하기 로그 파일 남기기

Web Server

simple web server

express

middlewares

routing

cookie & session

file upload

모듈화

뷰 템플릿

- ejs
- pug

사용자 인증

- 패스포트

채팅서버

- socket.io

JSON-RPC

- RESTful

위치기반 서버

- 공간데이터 처리

모바일 서버

- 단말 관리
- 푸시 서버 → firebase

게시판

Deployment on Cloud Services

git

heroku

openshift

aws

05 | 웹 서버 만들기

간단한 웹 서버 만들기 클라이언트가 웹 서버에 요청할 때 발생하는 이벤트 처리하기 클라이언트에서 요청이 있을 때 파일 읽어 응답하기 파일을 스트림으로 읽어 응답 보내기 파일을 버퍼에 담아 두고 일부분만 읽어 응답 보내기 서버에서 다른 웹 사이트의 데이터를 가져와 응답하기 익스프레스로 웹 서버 만들기 새로운 익스프레스 서버 만들기 미들웨어로 클라이언트에 응답 보내기 여러 개의 미들웨어를 등록하여 사용하는 방법 알아보기 익스프레스의 요청 객체와 응답 객체 알아보기 익스프레스에서 요청 객체에 추가한 헤더와 파라미터 알아보기 미들웨어 사용하기 static 미들웨어 body-parser 미들웨어 요청 라우팅하기 라우터 미들웨어 사용하기 URL 파라미터 사용하기 오류 페이지 보여 주기 express-error-handler 미들웨어로 오류 페이지 보내기 토큰과 함께 요청한 정보 처리하기 쿠키와 세션 관리하기 쿠키 처리하기 세션 처리하기 파일

업로드 기능 만들기 multer 미들웨어 설치해서 파일 업로드하기 클라이언트의 요청 처리 함수 추가하기

node.js with Databases

06 | 데이터베이스 사용하기

몽고디비 시작하기 몽고디비란? 몽고디비 사용을 위한 프로그램 설치하기 몽고디비에 데이터를 추가하거나 조회하기 익스프레스에서 몽고디비 사용하기 새로운 프로젝트 만들기 mongodb 모듈을 사용하여 로그인 기능 만들기 사용자가 보내온 아이디와 비밀번호 비교하기 로그인 처리를 요청하는 패스에 라우팅 함수 추가하기 사용자 추가 기능 만들기 데이터베이스 관리 도구 사용하기 몽구스로 데이터베이스 다루기 몽구스 모듈 사용하기 몽구스로 사용자 인증하기 인덱스와 메소드 사용하기 사용자 리스트 조회 기능 추가하기 비밀번호 암호화하여 저장하기 virtual 함수 사용하기 스키마 객체의 virtual( ) 함수 사용법 알아보기 비밀번호 암호화하여 저장하는 코드 적용하기 MySQL 데이터베이스 사용하기 관계형 데이터베이스 간단하게 이해하기 MySQL 설치하기 화면이 있는 관리 도구 HeidiSQL 설치하기 MySQL을 사용하는 사용자 추가 기능 만들기 사용자 추가 요청을 처리하는 함수 만들기 MySQL에 들어 있는 사용자 정보로 로그인하기

07 | 익스프레스 프로젝트를 모듈화하기

모듈화 방법 자세히 살펴보기 다양한 방법으로 모듈 만들기 exports에 객체 지정하기 module.exports를 사용해서 객체를 그대로 할당하기 module.exports에 함수만 할당하기 exports와 module.exports를 함께 사용하기 require( ) 메소드의 동작 방식 이해하기 모듈을 분리할 때 사용하는 전형적인 코드 패턴 함수를 할당하는 코드 패턴 인스턴스 객체를 할당하는 코드 패턴 프로토타입 객체를 할당하는 코드 패턴 사용자 정보 관련 기능을 모듈화하기 스키마 파일을 별도의 모듈 파일로 분리하기 사용자 처리 함수를 별도의 모듈 파일로 분리해 보기 설정 파일 만들기 설정 파일 분리하기 설정 파일에 데이터베이스 스키마 정보 넣기 설정 파일에 라우팅 정보 넣기 UI 라이브러리로 웹 문서 예쁘게 꾸미기 Semantic UI 라이브러리로 웹 문서 꾸미기 Card 컴포넌트 추가하기 □table□ 태그로 로그인 입력 상자와 버튼 추가하기 □style□ 태그로 전체 화면 모양 만들기 로그인 카드를 모바일 화면에 맞도록 CSS 조정하기 반응형 웹으로 웹 문서를 구별해서 보여 주기 사용자 리스트 웹 문서 수정하기 사용자 조회에 응답하는 웹 문서 꾸미기 사용자 추가용 웹 문서 꾸미기

08 | 뷰 템플릿 적용하기

ejs 뷰 템플릿 사용하기 뷰 템플릿으로 로그인 웹 문서 만들기 뷰 템플릿으로 사용자 리스트 웹 문서 만들기 뷰 템플릿으로 사용자 추가 웹 문서 만들기 pug 뷰 템플릿 사용하기 pug로 HTML 문서 만들기 pug 템플릿으로 로그인 웹 문서 만들기 pug 템플릿으로 사용자 리스트 웹 문서 만들기 pug 템플릿으로 사용자 추가 웹 문서 만들기

09 | 패스포트로 사용자 인증하기

패스포트로 로그인하기 패스포트의 기본 사용 방법 살펴보기 플래시 메시지와 커스텀 콜백 이해하기 스트래티지 설정과 검증 콜백 로컬 인증하기 로컬 인증을 위해 데이터베이스 스키마와 패스포트 설정하기 로그인과 회원가입 화면을 만들기 위한 라우팅 함수 등록하기 로그인과 회원가입 화면을 만들기 위한 뷰 템플릿 만들기 로그인과 회원가입 기능 실행하여 확인하기 패스포트 관련 코드를 모듈화하기 페이스북으로 로그인하기

10 | 채팅 서버 만들기

socket.io 사용하기 socket.io를 사용하기 위해 모듈 설치하기 app.js 메인 파일에 기본 코드 하나씩 추가하기 사용자가 웹 브라우저에서 볼 웹 문서 만들기 서버에 보낸 메시지를 그대로 받기 일대일 채팅하기 그룹 채팅하기 방 만들기 그룹 채팅에서 메시지 보내기 채팅 웹 문서 예쁘게 꾸미기

11 | JSON-PRC 서버 만들기

JSON-PRC를 웹 서버에 적용하기 JSON-RPC 모듈 설치하여 사용하기 echo 함수 만들어 실행하기 echo 함수의 오류 테스트하기 계산기 모듈 추가하여 실행하기 1단계: 핸들러 모듈 파일 만들기 2단계: 핸들러 모듈 파일 등록하기 3단계: 클라이언트 웹 문서에서 호출하기 데이터베이스에서 사용자 리스트 조회하기 데이터 부분을 암호화하기

12 | 위치 기반 서비스 서버 만들기

커피숍 위치 저장하기 커피숍의 위치 정보 다루기 커피숍 스키마 만들기 커피숍 정보를 추가하고 커피숍 리스트 조회하기 가장 가까운 커피숍 찾기 영역 안의 커피숍 찾기 반경 안의 커피숍 찾기 지도에 커피숍의 위치 표시하기 지도에 내 위치 보여 주기 가장 가까운 커피숍을 찾아 지도 위에 보여 주기 일정 범위 안의 커피숍을 찾아 지도 위에 보여 주기

13 | 모바일 서버 만들기

모바일 단말에서 웹 서버로 요청하기 안드로이드 앱 개발 도구 설치하기 실제 모바일 단말에 연결하기 모바일용 서버 프로젝트 만들기 안드로이드 앱 프로젝트 만들기 모바일 단말 관리 기능 만들기 데이터베이스 스키마 추가하기 라우팅 함수 추가하기 단말 정보 추가를 요청하는 앱 만들기 웹 브라우저에서 단말 리스트 조회하기 모바일 단말로 푸시 메시지 전송하기 단말의 등록 ID를 확인하여 모바일 서버로 전송하기 웹 브라우저에서 푸시 메시지 전송하기

14 | 게시판 만들기

스키마를 추가하고 페이지 단위 조회 방식 이해하기 페이지 단위로 조회하기 글쓰기와 글 조회 기능 만들기 페이지 단위로 글 목록 조회하기 1단계 ? 라우팅 함수 만들기 2단계 ? config.js 파일에 라우팅 함수 등록하기 3단계 ? 응답 웹 문서를 구성할 뷰 템플릿 만들기 4단계 ? 글 목록 조회를 요청하는 웹 페이지 만들기

15 | 클라우드에 올리기

Git 클라이언트 설치하기 헤로쿠 클라우드에 올리기 헤로쿠 사이트에 회원가입하고 앱 만들기 Heroku CLI 설치하기 배포하기 위한 설치 및 준비하기 소스 업로드하기 오픈시프트 클라우드에 올리기 아마존 클라우드에 올리기 배포에 필요한 기능 알아보기 비정상으로 종료되었을 때 자동으로 다시 시작하기 클러스터링 사용하기 레디스 사용하기 레디스로 subscribe, publish 하기 채팅 서버에서 레디스를 사용하는 샘플