

Express 요청데이터 분석

사용자의 요청데이터

- 사용자의 요청데이터를 추출하는 방법은 2가지가 존재한다.
 1. HTTP POST 요청의 본문(body)를 통해
 2. HTTP POST/GET 요청의 쿼리스트링을 통해
- 쿼리스트링이란?
: URL경로의 끝 부분에 물음표를 붙이고 이 뒤에 키-값형태로 이루어진 스트링
Ex) http://localhost/index?name=gildong&id=gil&password=1234

예제1) 사용자의 요청데이터 분석

1. express_req_data 디렉터리 생성
\$ mkdir express_req_data
2. 생성한 디렉터리(express_req_data)로 이동 후 npm init 진행
\$ cd express_req_data
\$ npm init
3. 클라이언트의 요청데이터를 분석하기 위한 여러 모듈 및 패키지 다운로드
\$ npm install express body-parser --save
4. main.js에 아래와 같이 입력
\$ vi main.js

```
"use strict";

const port = 3000;
const express = require("express");
const app = express();

// 요청해독을 초기화 (default : utf-8)
app.use(express.urlencoded({ extended: false }));
// 요청해독을 json으로 진행
app.use(express.json());

// 요청 데이터를 분석하는 미들웨어
app.use(function(req, res, next){
    console.log(req.body);
    console.log(req.query);
    next();
});

// POST 라우트
app.post("/", function(req, res){
    res.send("HTTP POST Successful");
});

// GET 라우트
app.get("/", function(req, res){
    res.send("HTTP GET Successful");
});

// 서버실행
app.listen(port, function(){
    console.log("Server port : ", port);
});
```

✓ 클라이언트 실행화면

: POST (linux에서 제공하는 curl명령어를 사용하였음)

```
wlgns12www@nodejs-test:~/nodejs/9_3_request$ curl --data \  
> "id=hongildong&password=1234" \ → 요청본문(body)  
> http://localhost:3000?name=hongildong → 요청 쿼리스트링  
HTTP POST Successful → HTTP POST 응답
```

: GET (쿼리스트링을 통해서만 요청데이터 전송가능)

← → ↻ 🏠 | 🛡️ ⓘ localhost:3000?id=hongildong&password=1234&name=hongildong
HTTP GET Successful → HTTP GET 응답
→ 요청 쿼리스트링

✓ 서버 실행화면

```
wlgns12www@nodejs-test:~/nodejs/9_3_request$ node main.js  
server port : 3000  
{ id: 'hongildong', password: '1234' } → POST 요청본문  
{ name: 'hongildong' } → POST 요청 쿼리스트링  
{ } → GET 요청본문  
{ id: 'hongildong', password: '1234', name: 'hongildong' } → GET 요청 쿼리스트링
```

참고문헌

- 조나단 웨슬러, 김성준 (2020.01.31). NODEJS로 프로그래밍 시작하기. 144-149.