

# Node.js 모듈을 활용한 철도패턴 데이터 접근을 위한 웹 어플리케이션 개발

오현진\*, 장용형\*, 오엽덕<sup>o</sup>

\*한국교통대학교 소프트웨어전공,

<sup>o</sup>한국교통대학교 소프트웨어전공

e-mail: 1826074@a.ut.ac.kr\*, jhk4538jj@naver.com\*, rdoh@ut.ac.kr<sup>o</sup>

## Development of Web application for railway pattern data approach using Node.js modules

HyeonJin Oh\*, Zhang Yong Heng\*, Ryumduck Oh<sup>o</sup>

\*Dept. of Software, Korea National University of Transportation,

<sup>o</sup>Dept. of Software, Korea National University of Transportation

### ● 요약 ●

본 논문에서는 node.js에서 제공하는 oracledb, express, ejs 모듈을 이용해 데이터베이스에 저장되어 있는 철도 패턴 데이터를 라우터를 통해 정해진 경로로 전달하여 ejs 파일로 작성된 페이지에 접근하여 출력하는 웹 어플리케이션을 구현하고자 한다. 웹 어플리케이션의 사용으로 철도 데이터를 필요로 하는 기업이나 사용자가 보다 쉽고 빠르게 데이터를 확인하고, 이를 원하는 방향으로 이용할 수 있다.

**키워드:** node.js, express, oracledb, ejs(embedded javascript),  
라우터(router), 데이터베이스(database)

## I. Introduction

데이터베이스는 다양한 산업군에서 이용되고 있으며, 각종 정보와 자료를 효율적으로 관리, 운용할 수 있어 그 중요성이 점차 커지는 추세이다.

현재 철도 데이터를 필요로 하는 기업이나 사용자가 점차 늘어나고 있고 해당 데이터를 제공하는 기업들도 생겨나고 있다. 하지만 현재 제공되는 데이터들은 csv 파일이나 json 파일 형식으로 제공되고 있어 데이터 가공이나 서비스 제공에 불편을 초래하기도 한다.

이에 본 논문에서는 Node.js의 기능 중 하나인 ejs 파일 기능을 이용하여 oracle 데이터베이스를 연동하고 저장된 데이터들을 웹으로 송출하는 웹 어플리케이션을 구현하여 클라이언트에게 쉽고 간편하게 데이터들을 제공한다.

연계 방안도 연구되고 있다.[4]

현재 국내에서 철도 데이터에 접근하기 위해서는 공공데이터 포털에서 제공하는 csv 파일이나 json 파일 등의 파일 형식을 다운로드하여 데이터를 가공하는 과정이 필요하다.

이로인해 기업이나 전문가들은 원하는 데이터들의 범위를 한정하여 데이터를 가공하는 번거로운 과정을 거치며, 일반 사용자들도 철도 데이터에 대한 접근에 거부감을 느끼게 만드는 요인으로 작용할 가능성이 있다.

따라서 본 논문에서는 데이터베이스에 저장된 데이터들을 기업, 전문가들뿐만 아니라 일반 사용자들도 쉽게 데이터베이스에 저장된 데이터를 눈으로 보고 확인할 수 있게 Node.js의 express, oracledb, ejs 모듈을 이용하여 사용자가 데이터를 불러오기 위한 버튼을 클릭하면 자동으로 OracleDB에 접근해 데이터를 서버로 전송하고, 해당 데이터를 웹 페이지에 보기 편한 형태로 출력해주는 웹 어플리케이션을 구현한다.

## II. Related Work

### 1. domestic trends

국내외에서는 데이터를 기반으로 한 철도 안전관리 구현 방안 및 모니터링에 대한 연구가 활발히 진행되고 있으며[1][2][3], 철도 이용객들에게 편의성을 제공하기 위한 공공데이터 포털 시스템과의

### III. The Proposed Scheme

#### 1. System Environment

웹 어플리케이션의 개발 환경은 Table 1.과 같다.

Table 1. web application development environment

개발 언어	Javascript / ejs
라이브러리	Node.js
DBMS	OracleDB
사용 모듈	express, oracledb, ejs

DBMS로는 대량의 정보처리에 좋은 성능을 보이고, 속도가 빠르며, 동시 다중 접속을 지원하고 서버/클라이언트의 환경을 제공하는 OracleDB를 이용하였다.

#### 2. System Architecture

웹 어플리케이션의 전체 구조는 아래 Fig. 1.과 같이 설계하였다.

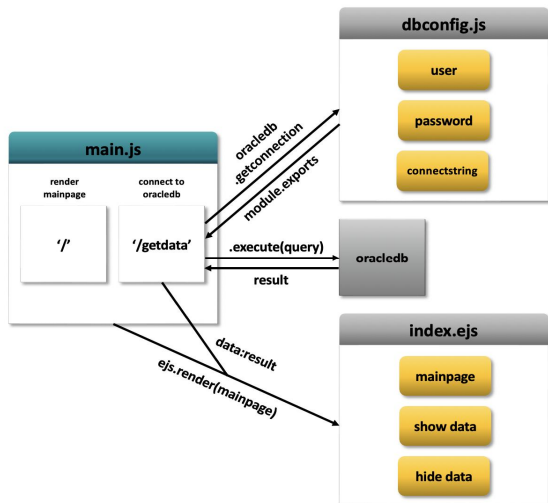


Fig. 1. Whole Structure of Web Application

프로젝트는 두 개의 js파일과 하나의 ejs 파일로 구성되어 있다. dbconfig.js 파일에는 oracle 데이터베이스에 연결하기 위한 userID, password, connectstring 정보를 module.exports를 통해 다른 js 파일에서 변수명으로 접근하여 사용할 수 있도록 하고 있다.

oracle 데이터베이스에 접근하기 위한 정보를 별도의 js 파일로 분리해둔 이유는 계정 정보에 대한 보안성 때문이다.

사용자가 데이터베이스의 데이터에 접근하는 과정은 아래 Fig. 2.와 같다.

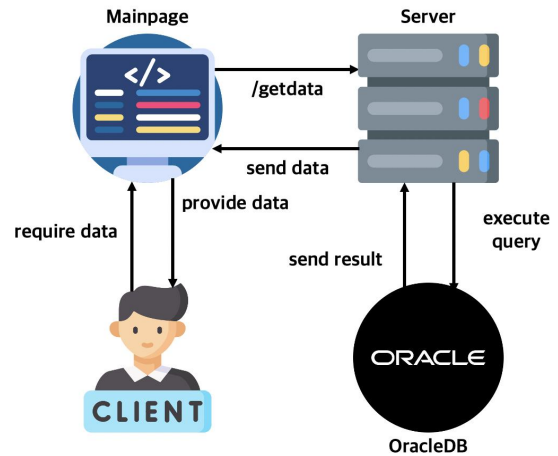


Fig. 2. Data Flow in Web Applications

사용자가 웹 페이지의 버튼을 눌러서 데이터를 요구하면, 페이지는 /getdata 경로로 접근하여 OracleDB와 연결 후 지정된 질의를 수행한다.

수행된 결과 데이터가 서버를 통해 다시 웹 페이지로 전달되어 화면에 출력되면, 사용자가 해당 데이터를 확인하는 방식이다.

#### 3. 프로시쥬어 연동1: main.js

main.js 파일에서는 express, oracledb, ejs, fs 모듈을 이용하고 있으며, index.ejs 파일을 fs 모듈을 이용해서 mainpage 변수에 불러와 저장하였다.

express 모듈의 라우터를 이용하여 '/' 경로에는 간단히 아무 데이터도 출력되지 않는 상태의 mainpage로 이동하도록 한다.

'/getdata' 경로에는 oracledb 모듈과 dbconfig.js 파일의 변수들을 이용하여 oracle 데이터베이스와의 연결을 생성하고, 연결이 성공적으로 수행되었을 경우 생성되어있는 테이블의 데이터를 모두 검색하는 query를 수행한다. 수행한 결과를 mainpage로 전달한다.

#### 4. 프로시쥬어 연동 2: index.ejs

index.ejs 파일은 간단히 데이터베이스의 테이블을 출력할 수 있도록 Fig. 3.와 같이 디자인 해두었으며, 데이터 불러오기 버튼을 누르면 /getdata 경로로 이동하도록 한다.

이동후 데이터베이스로부터 데이터를 받아와 Fig. 4.과 같이 출력할 수 있게 하였고, 데이터 감추기 버튼을 누르면 다시 원래 페이지의 경로인 / 경로로 이동하여 아무런 데이터가 없는 상태의 메인페이지로 돌아가도록 구현한다.

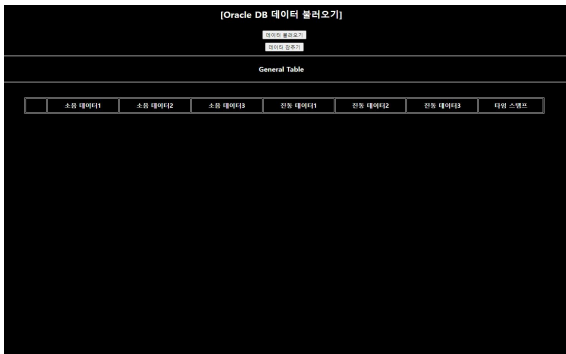


Fig. 3. Composition of the Mainpage

실제 Fig. 4.과 Fig. 5.를 비교해보면, oracledb에 저장되어 있는 데이터와 화면에 출력된 데이터가 일치하는 것을 확인할 수 있다.



Fig. 4. Mainpage with Data output

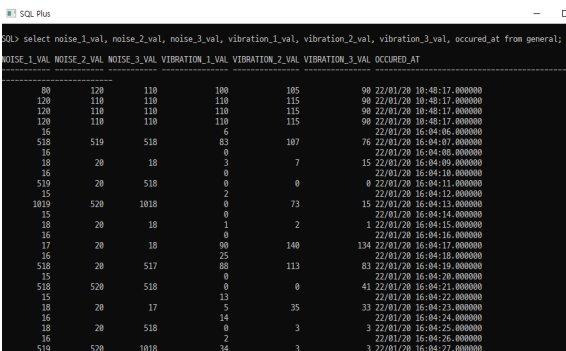


Fig. 5. Data Stored in Real Database

ejs 파일은 일반 html 파일과 작성 방법은 거의 비슷하지만, <% %> 안에 자바스크립트 코드를 작성하고, /getdata 경로로 이동해 메인페이지에 데이터를 받아둔 상태라면, 지정된 데이터들을 모두 표의 형태로 출력해서 클라이언트가 볼 수 있도록 한다.

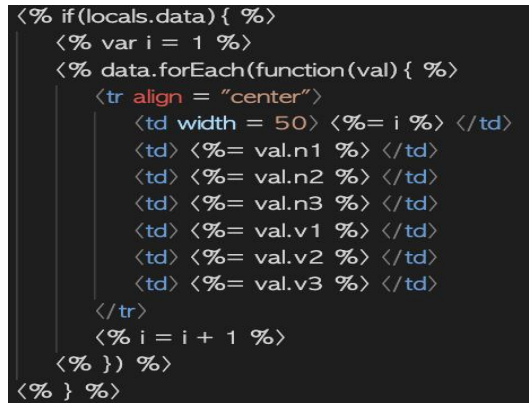


Fig. 6. Example of Writing Javascript Code in ejs file

#### IV. Conclusions

본 논문에서는 Node.js의 express, oracledb, ejs 모듈을 활용하여 데이터가 필요한 기업, 전문가 또는 일반 사용자들이 개발한 웹애플리케이션을 통해 번거로운 데이터 가공 과정을 거치지 않고 페이지에 접속하여 '데이터 불러오기' 버튼을 누르기만 하면 버튼에 연결된 이벤트에 의해 데이터를 다시 페이지로 불러와 한눈에 확인할 수 있다.

/getdata를 통해 경로를 이동하고, OracleDB를 연동, 데이터 검색을 수행할 수 있다.

본 논문에서는 아직 데이터를 전체적으로 가져와 출력하는 기능만 구현하였으나, 연구를 심층적으로 하여 특정 데이터만을 페이지에 출력하는 기능, 접근 권한에 따른 데이터 접근 범위 이원화, 데이터값의 크기에 따른 주목 여부 표시 등의 기능을 더할 수 있다.

앞서 언급한 기능들을 추가하고 페이지를 정리하는 과정을 거친다면, 아직 빅데이터를 활용한 안전관리 및 서비스 이용자에게 대한 정보 제공이 미흡한 산업 분야에 본 웹애플리케이션 형태로 정보를 제공하여 다양한 기업 및 전문가, 그리고 사용자들에게 간편하면서도 보기 편한 데이터의 제공을 도울 수 있을 것으로 보인다.

#### ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2020R1A2C1101867).

#### REFERENCES

[1] THE KOREA TRANSPORT INSTITUTE, "Big Data-based railroad safety management Implementation plan research", 2018.

- [2] Abbas Abduvaytov, Rakhimov Mukhammad Abdu Kayumbek, Heung Seok Jeon, Ryumduck Oh, "The Real Time Railway Monitoring System suitable for Multi-View Object based on Sensor Stream Data Tracking", ICISCT2020, Uzbekistan, 2020
- [3] Rakhimov Mukhammad Abdu Kayumbek, Jeong Won Kang, Ryumduck Oh, "Railway Monitoring System Simulation on the Sensors Stream Data", ICISCT2021, Uzbekistan, 2021
- [4] Sun-won Kim, Yeong-cheol Song, Jong-kyu Lee, Chi-hyung Ahn, "Linkage Methodology of Government's Open Data Portal System for the Railway Passenger's Convenience", 2015