

실습

Mini-project

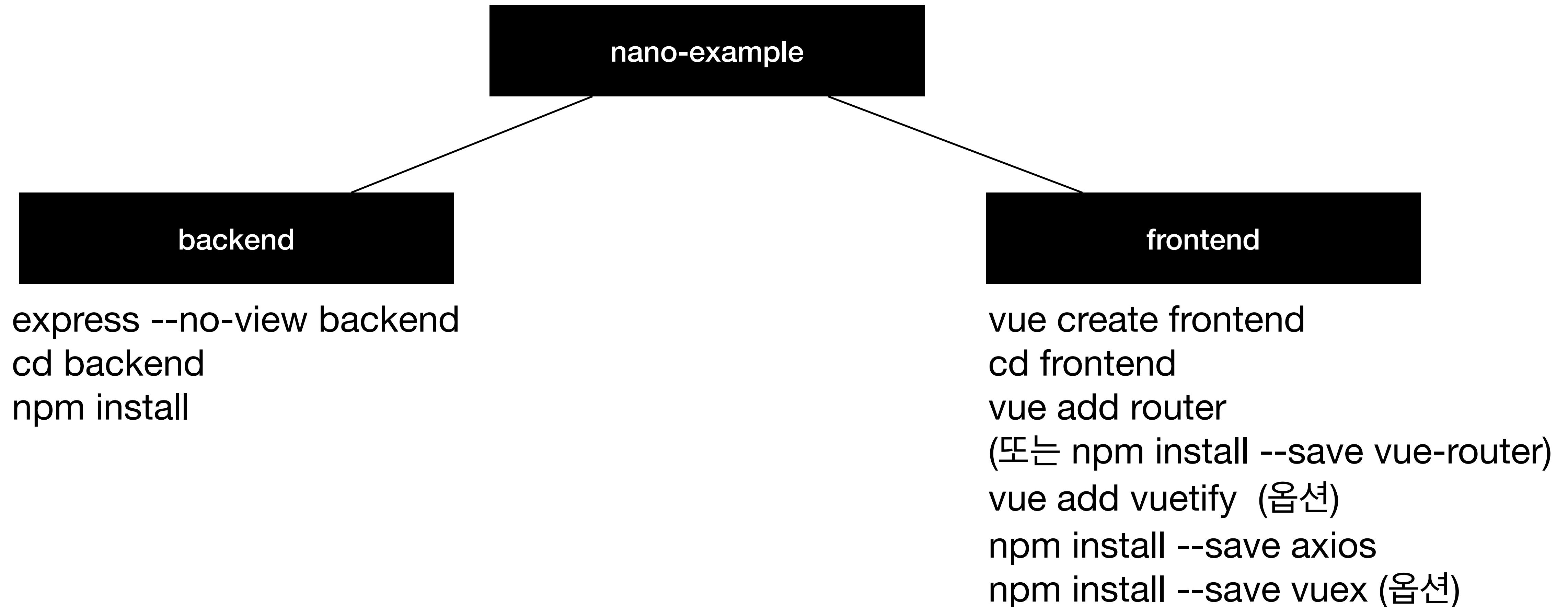
구현 방법

- Node와 Vue 프로젝트를 각각 별개의 프로젝트로 구현하고 운영도 따로 하는 방법
 - Node와 Vue가 다른 서버에서 구동되는것 가능
- 서버와 Vue 프로젝트를 각각 생성하지만 Vue의 배포 파일을 Node 서버의 public 폴더로 지정하는 방법
 - Node 서버만 켜도 동작함

Back-end

- Node.js 와 Express, sqlite를 이용해 구현.
- Express Generator를 이용하면 기본 구조가 잡혀있는 Back-end를 만들 수 있음.
 - `npm install -g express-generator` : 설치 이후 다음 명령어로 앱 코드 생성
 - `express --view=pug myapp` : view template engine을 pug를 쓰는 myapp
 - `express --no-view myapp` : view template engine이 없는 myapp

프로젝트 구조



frontend 준비

- frontend/ 폴더에 vue.config.js 파일을 생성

```
1  module.exports={
2    ...
3    devServer:{
4      // /api로 시작되는 요청 주소 정의
5      proxy:{
6        '/api':{
7          target: 'http://localhost:3000/api',
8          changeOrigin:true,
9          pathRewrite:{
10             '^/api':''
11          }
12        }
13      },
14
15      // 배포 위치
16      // npm run build를 입력해 배포한다.
17      outputDir: '../backend/public',
18
19      transpileDependencies: [
20        'vuetify'
21      ]
22  }
```

← vuetify 설치했을 때만

Apis

- backend 의 api 계획

url	method	response
/api/books	get	[{ bookId:string, title:string, author:string, cover:string}]
/api/books/:bookId	get	{ bookId:string, title:string, subtitle:string, author:string, translator:string, cover:string price:int, publisher:string, page:int, ISBN:string, info:string}

backend

- routes 폴더에 books.js 파일을 추가

```
1  const express = require('express');  
2  const router = express.Router();  
3  
4  module.exports = router;
```


backend

- app.js 파일에 다음 내용 추가



```
1  var express = require('express');
2  var path = require('path');
3  var cookieParser = require('cookie-parser');
4  var logger = require('morgan');
5
6  var indexRouter = require('./routes/index');
7  var usersRouter = require('./routes/users');
8  var booksRouter = require('./routes/books');
9
10 var app = express();
11
12 app.use(logger('dev'));
13 app.use(express.json());
14 app.use(express.urlencoded({ extended: false }));
15 app.use(cookieParser());
16 app.use(express.static(path.join(__dirname, 'public')));
17
18 app.use('/', indexRouter);
19 app.use('/users', usersRouter);
20 app.use('/api/books', booksRouter);
21
22 module.exports = app;
```



backend

- 프로젝트에 model 폴더 추가 후 그안에 books.js 파일 추가
 - naver 책 정보 참고

```
1 module.exports=[
2   {
3     "bookId":"20578083", "title":"완전한 행복", "subtitle":""," "author":"정유정",
4     "cover":"https://bookthumb-phinf.pstatic.net/cover/205/780/20578083.jpg?update=20210720",
5     "price":15800, "publisher":"은행나무", "page":524, "ISBN":"9791167370280",
6     "info":"다시, 정유정!\n자기애의 늪에 빠진 삶은 얼마나 위태로운가,\n압도적 서사 위 정교하고 서늘한 공포 \n우리가 기다린 바로 그, 정유정!"
7   },
8   {
9     "bookId":"20777131", "title":"달려구트 꿈 백화점 2", "subtitle":"단골손님을 찾습니다", "author":"이미예",
10    "cover":"https://bookthumb-phinf.pstatic.net/cover/207/771/20777131.jpg?update=20210720",
11    "price":13800, "publisher":"팩토리나인", "page":308, "ISBN":"9791165343729",
12    "info":"100만 독자를 사로잡은 《달려구트 꿈 백화점》, 그 두 번째 이야기"
13  },
14  {
15    "bookId":"19007535", "title":"미드나잇 라이브러리", "subtitle":"Midnight Library",
16    "author":"매트 헤이그", "translator":"노진선",
17    "cover":"https://bookthumb-phinf.pstatic.net/cover/190/075/19007535.jpg?update=20210727",
18    "price":15800, "publisher":"인플루엔셜", "page":408, "ISBN":"9791191056556",
19    "info":"밤 12시, 죽기 바로 전에만 열리는 마법의 도서관에서 인생의 두 번째 기회를 드립니다"
20  },
21  {
22    "bookId":"20657805", "title":"오늘 밤, 시계에서 이 사랑이 사라진다면",
23    "subtitle":"매일 기억을 잃는 너와 두 번 다시 돌아갈 수 없는 사랑을 했다",
24    "author":"이치조 미사키", "translator":"권영주",
25    "cover":"https://bookthumb-phinf.pstatic.net/cover/206/578/20657805.jpg?update=20210724",
26    "price":14000, "publisher":"모모", "page":375, "ISBN":"9791191043297",
27    "info":"밤에 자고 일어나면 기억이 리셋되는 '선행성 기억상실증'을 앓는 소녀 히노 마오리와 무미건조한 인생을 살고 있는 평범한 고등학생 가미야 도류
28  }
29 ]
```

https://drive.google.com/file/d/1RpUm6_YatbsGdz8AhTfRJZEA55tc-D5v/view?usp=sharing

backend

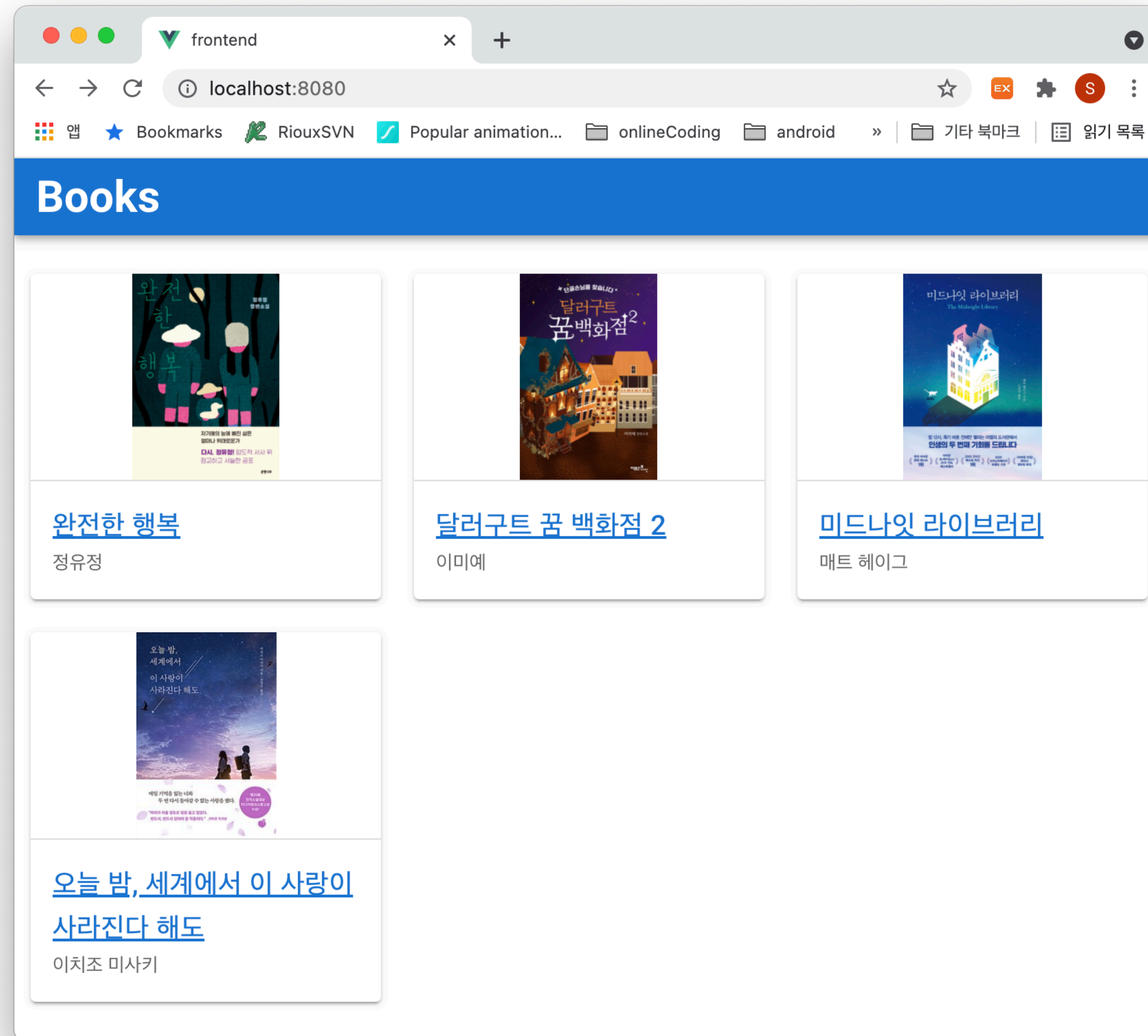
- routers/books.js 파일 수정
- <http://localhost:3000/api/books/>

```
1  const express = require('express');
2  const router = express.Router();
3  const books = require('../model/books');
4
5  module.exports = router;
6
7  router.get('/', (req, res)=>{
8    let briefs = []
9    books.forEach(element => {
10      let {bookId, title, author, cover} = element
11      briefs.push({bookId, title, author, cover})
12    });
13    res.json(briefs)
14  })
15
16
17  router.get('/:bookId', (req, res)=>{
18    const bookId = req.params.bookId
19    let book = books.find(element=> element.bookId==bookId)
20    res.json(book)
21  })
```

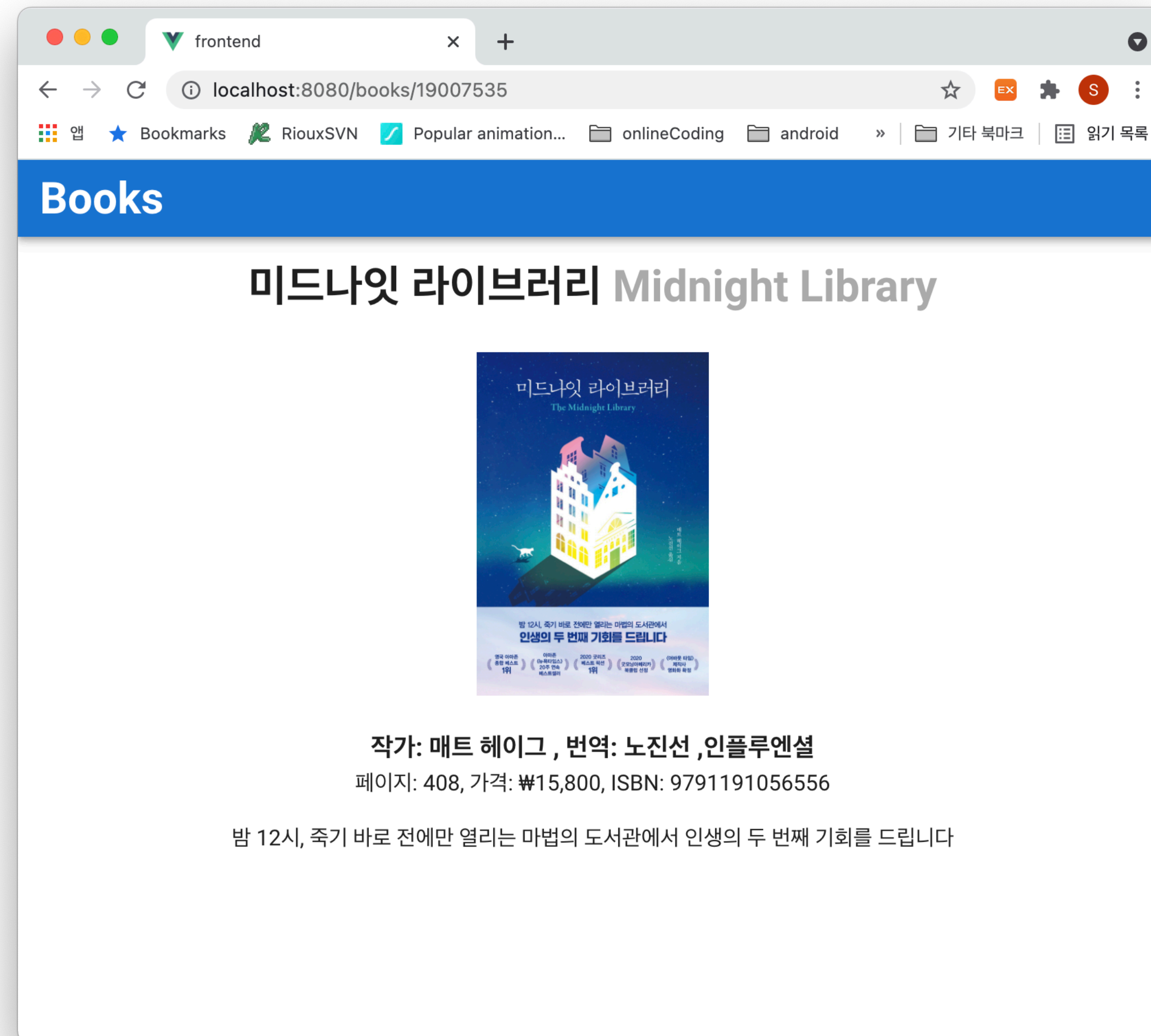
frontend

- “/” : 책 리스트를 받아와 책 리스트 화면을 구성한다.
- “/books/:bookId”: 선택된 책의 상세 정보 화면을 출력한다.

frontend example



frontend example



frontend 배포

- 기능 테스트를 완료했다면
- frontend 폴더에서 npm run build를 실행
- 빌드가 끝났다면 backend/public 폴더를 확인. 파일들이 추가 된 것을 확인한다.
- 이제 vue app은 실행하지 않고 Node 서버만 실행해서 동작을 확인한다.

- <https://drive.google.com/drive/folders/1zG-NRMYfUsu5xhyYUIR0TrDIcBndV25W?usp=sharing>

-