[01-19] 2st Project

1) product.js

```
import db from "../../config/database.js";
```

```
export const showRooms = (req, res) => {
  let query = "SELECT * FROM room";
  db.query(query,(err,result) => {
     if(err) {
        console.log(err);
        res.send(err);
     } else {
        res.json(result);
     }
  })
}
```

- 현재, get 으로 받아오기 때문에 let query 가 간단함
- 만약, post 로 받아올 경우 아래의 예시를 참고
    - where 조건이 붙고 let data 를 작성해서 db.query에 파라미터를 두개를 부여하게 됨
    - db.query(query, data

```
export const showDetails = (req, res) => {
  let query = "SELECT * FROM room where RoomNum=?";
  // let query = "SELECT * FROM room";
  let data = [req.body.RoomNum]
  db.query(query, data, (err,result) => {
     if(err) {
        console.log(err);
        res.send(err);
     } else {
        res.json(result);
     }
  })
}
```

2) Router.js

```
import Express from "express"; //express 사용할거야.
import {joinId, PostOrders, putPassword , PostUserId, showDetails, showRooms, showUser, admin }
from "./controllers/product.js";

const router = Express.Router(); //express의 router 사용하기위함.

router.get('/rooms', showRooms);
export default router;
```

3) rooms.vue

3-1) 우리는 방목록의 상세보기를 눌렀을 경우 실행될 메소드에 대한 정의가 필요함

- JS 에서의 소스코드
  - nextPage를 클릭시 router.push 로 정보를 받아와 화면에 보낼건데
  - name : roomDetail 이라는 애로 (얘를 지칭하면 component 로 연결된다는것?)
  - dataObj 는 내가 어떤 이름으로 해도 상관없음
  - JSON.stringify 로 보내서 parse 해야 잘 도착함
    - parse는 데이터를 받을 roomDetaild에서 작성됨
  - this.room 을 하면 상단의 room으로 인지됨
  - 해당 파라미터는 index로 v-for에 의해 순서대로

```
export default defineComponent({
 data() {
  return {
    room: [],
  };
 },
 methods :{
  init() { //db 호출
    axios.get('http://localhost:3000/rooms')
        .then(res => {
          console.log(res.data)
          this.room = res.data; //room에 응답값 박아
        })
        .catch(function (error) {
          console.log(error);
        })
  },
  nextPage(index){  //for문 순번 받아옴
    console.log(index)
    console.log(this.room[index])
    router.push({
      name: "roomDetail",
      state: {  //고정이름
        dataObj : JSON.stringify(this.room[index]),  //dataObj : 변수(변동가능)
      }
    })
  }
```

- HTML 에서의 소스코드
  - v-for 에 의해 순서대로 출력되어 해당하는 index로 연결되도록
  - 파라미터에 index를 써주는 것

```
<a href="#" class="primary-btn" @click="nextPage(index)">상세 보기</a>
```

4) index.js

```
{
  path: "/roomDetail",
  name: 'roomDetail',
  component:roomDetail
},
```

5) product.js

```
export const showRooms = (req, res) => {
  let query = "SELECT * FROM room";
  db.query(query,(err,result) => {
     if(err) {
        console.log(err);
        res.send(err);
     } else {
        console.log(result);  //이 부분을 추가
        res.json(result);
     }
  })
}
```

- console.log(result); 를 추가함으로 인해
- 해당 결과값인 DB의 내역(room)과 관련된 내역이 console 창에도 확인되어
- 현재 어디까지 값을 받아오는지 확인이 됨

```
C:₩Users₩student00₩Desktop₩MH₩study history₩2st Project₩git₩Project02₩TM₩backend>node index.js
Server running at http://localhost:3000
[
  {
    RoomNum: 1001,
    Rating: 'Superior',
    Bed: '더블 1구',
    Bath: 1,
    Terrace: 1,
    Spa: 1,
    Pool: 0,
    Rooftop: 0,
    View: 'CITY',
    BasicPersonnel: 2,
    MaxPersonnel: 4,
    AddPrice: 50000,
    Price: 400000,
    Size: 46,
    Checkin: '16:00',
    Checkout: '11:00',
    RoomImg: 'common/room/room-1.jpg',
    Explain: '소중한 사람들과 함께하는 시간은 특별합니다. 혼자 떠난 여행에서 고요히 내면을 응시하는 시간을 가질 수 있다
면, 함께하는 여행에서는 외면의 확장과 배가되는 기쁨을 느낄 수 있습니다. 같은 차 안에 앉아 같은 전경을 바라보며 감탄하고
 이것도 먹어 보라며 서로의 접시에 산해진미를 내려놓고, 미처 살피지 못한 요소를 발견해 눈앞에 가져다 줍니다. 지금 여기에
함께 있다는 느낌, 동행한 이의 웃는 얼굴을 더 많이 볼 수 있는 순간, 무엇으로도 대체할 수 없을 함께하는 여행의 묘미일 것입
니다. 따로 또 같이 즐기는 여행의 풍요로움, Nonamed에서 만나실 수 있습니다.'
  },
```