1. json-server 개요



- ❖json-server란?
 - json 파일을 이용해 간단한 REST API Server를 생성해주는 도구
 - 주로 백엔드를 Mockup 하기 위한 도구. 테스트 서버로 사용
- ❖설치 방법
 - 두가지 방법
 - 전역 설치
 - npm install -g json-server
 - 로컬 설치
 - npm install json-server -D
 - 실행 방법
 - db.json을 생성해 샘플 데이터를 추가
 - npx json-server --watch db.json

2. db.json 생성



❖ db.json

```
"contacts":
{ "name": "나연", "tel": "010-2222-2221", "address": "서울", "id": 1001 },
{ "name": "정연", "tel": "010-2222-2221", "address": "서울", "id": 1002 },
{ "name": "모모", "tel": "010-2222-2221", "address": "서울", "id": 1003 },
{ "name": "사나", "tel": "010-2222-2221", "address": "서울", "id": 1004 },
  "name": "지효", "tel": "010-2222-2221", "address": "서울", "id": 1005 },
  "name": "미나". "tel": "010-2222-2221". "address": "서울". "id": 1006 }.
  "name": "다현", "tel": "010-2222-2221", "address": "서울", "id": 1007 },
  "name": "채영", "tel": "010-2222-2221", "address": "서울", "id": 1008 },
  "name": "쯔위". "tel": "010-2222-2221". "address": "서울". "id": 1009 }.
  "name": "박초롱", "tel": "010-2222-2221", "address": "서울","id": 1010 },
  "name": "윤보미", "tel": "010-2222-2221", "address": "서울","id": 1011 },
  "name": "정은지", "tel": "010-2222-2221", "address": "서울","id": 1012 },
  "name": "손나은", "tel": "010-2222-2221", "address": "서울", "id": 1013 },
  "name": "김남주", "tel": "010-2222-2221", "address": "서울","id": 1014 },
  "name": "오하영", "tel": "010-2222-2221", "address": "서울","id": 1015 },
  "name": "솔라", "tel": "010-2222-2221", "address": "서울", "id": 1016 },
  "name": "문별", "tel": "010-2222-2221", "address": "서울", "id": 1017 },
  "name": "휘인", "tel": "010-2222-2221", "address": "서울", "id": 1018 },
  "name": "화사", "tel": "010-2222-2221", "address": "서울", "id": 1019 },
  "name": "유주", "tel": "010-2222-2221", "address": "서울", "id": 1020 }
```

3. 서버 기동



❖ 기동 방법

- npx json-server --watch db.json
 - 기본 포트 :3000번
 - http://localhost:3000
- db.json을 생성할 때의 contacts와 같은 수준의 속성이 엔티티로 간주됨
 - http://localhost:3000/contacts
- 기본 접근 라우트
 - 전체 조회 : GET /contacts
 - 한건 조회 : GET /contacts/1001
 - 한건 추가: POST /contacts
 - 한건 수정: PUT /contacts/1001
 - 한건 삭제: DELETE /contacts/1001

4. 데이터 조회

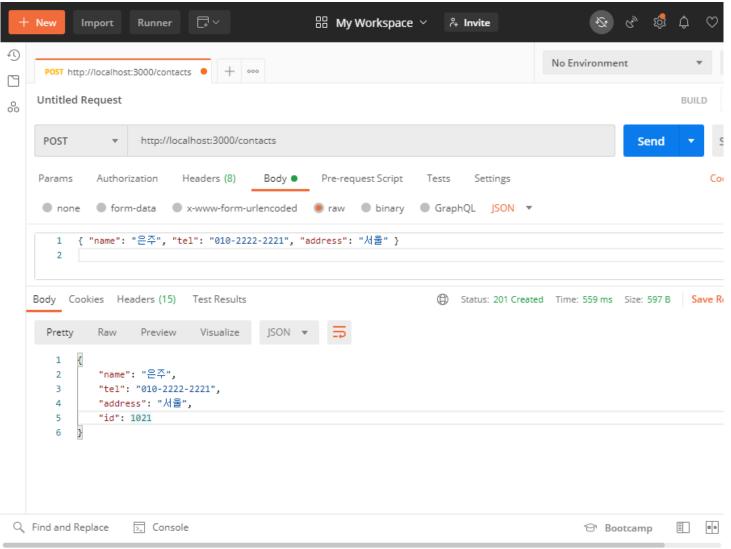


- ❖ 목록 조회
 - GET /contacts
- ❖ 필터링 방법
 - GET /contacts?id=1001&id=1005 //id가 1001또는 1005인 아이템 조회
 - GET /contacts? name=유주 //이름이 '유주'인 아이템 조회
 - GET /contacts?age_gte=20 //나이가 20세 이상인 아이템 조회
 - 연산자: _get, _lte, _gt, _lt, _ne
 - Full Text Search: GET /contacts?q=나
 - LIKE: /contacts? name_like=^나 //정규식을 사용: ^나 (시작하는), 연\$ (끝나는)
- ❖ 페이징
 - GET /contacts?_page=1&_limit=3 //한페이지에 3개씩 1번째 페이지
 - 기본값:_page=1,_limit=10
- ❖ 정렬
 - GET /contacts?_sort=id&_order=desc //id 로 내림차순 정렬
 - GET /contacts?_sort=name,tel&_order=asc,desc

5. 데이터 추가



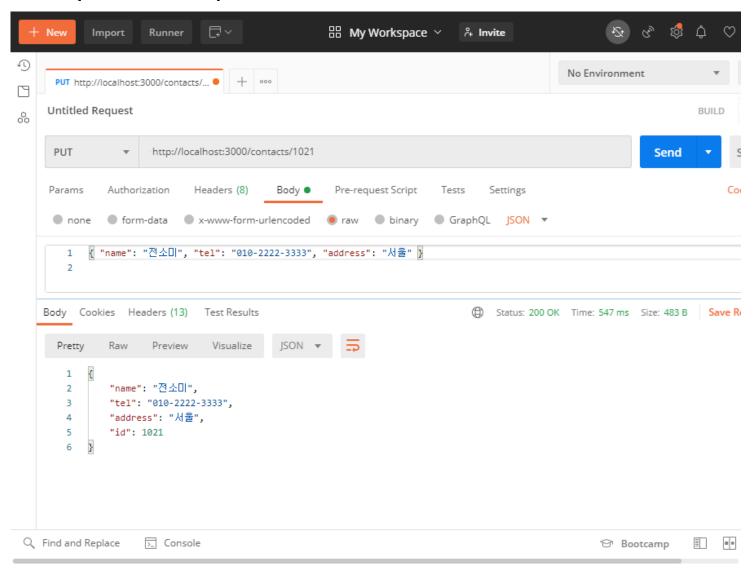
❖ POST /contacts



6. 데이터 수정



PUT /contacts/1021



7. 데이터 삭제



❖ DELETE /contacts/1021

