**SSAFY 2022 8기 Web PJT 관통 프로젝트**

|  |  |
| --- | --- |
| 구분 | DAY 4 |
| 진행일자 | 2022년 9월 16일 |
| 주제 | Data Visualizor |
| 학습내용 | 1. 요구사항 정리 2. 기술 스택 3. 개발환경 구축 4. 기본 세팅하기 5. 백엔드 구축 6. 프론트엔드 구축 7. 심화 과제 |
| 예상구현기간 | 8H |

임베디드 뿐만 아니라 수 많은 분야에서 data들은 존재하기 마련이다. 이에 따라 data를 요구사항에 맞게 시각화(visualization) 하는 연습이 필요하다. 따라서 해당 프로젝트를 통해

네이버 datalab API를 사용해 키워드 검색어 추이를 가져와 차트로 나타내는 Data Visualizing을 진행한다.

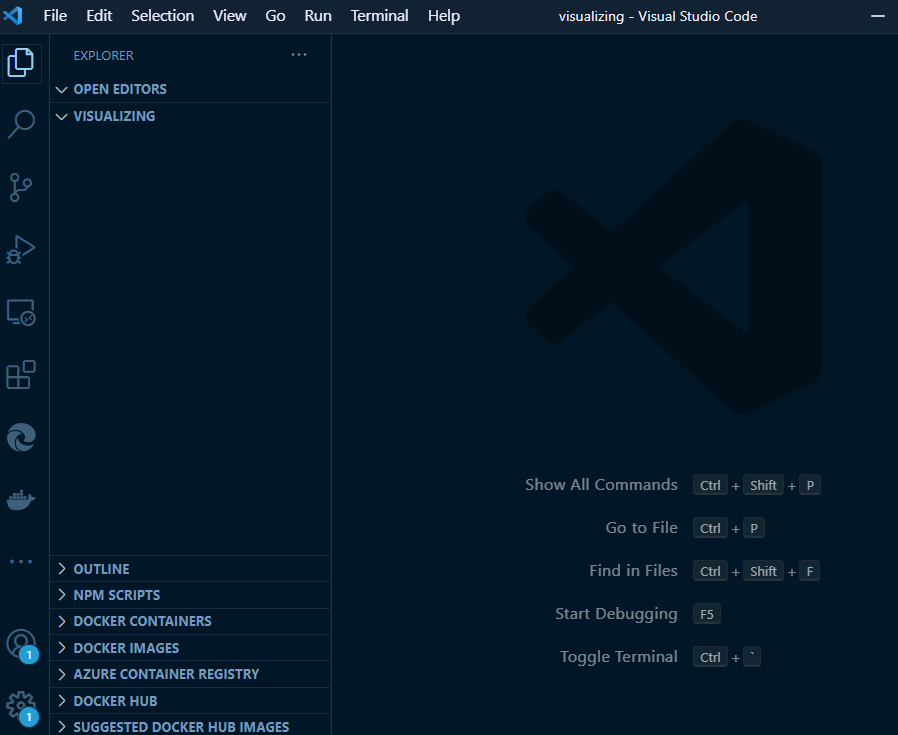


1. **요구사항 정리**

naver datalab api를 활용해서 data를 시각화 한다.

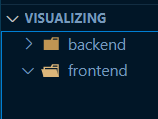
* 1. API 서버 구축하기
     1. express 서버에서 naver datalab api를 가져온다.
     2. api를 axios로 가져와서 리턴해주는 프로젝트를 생성한다.
  2. Vue.js로 visualizing
     1. axios로 express단에 요청을 보낸다
     2. 응답 값을 chart화 시킨다.

1. **기술 Stack**
2. express(Node.js)
3. Vue.js
4. Element-UI
5. **개발 환경 구축**   
   기본적인 웹 개발 Visual Studio Code를 사용한다.

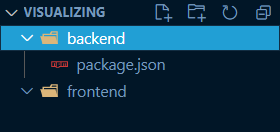
****

1. **기본 세팅하기**
   1. **백엔드 구축**

기본 폴더구조를 세팅해 준다.



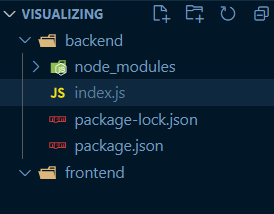
npm init명령어를 통해, 기본 package.json 파일을 생성한다.



이제 프로젝트에 필요한 Package를 설치하자.



|  |  |
| --- | --- |
| Pacakge 이름 | 내용 |
| express | Node.js 라우팅 &서버 역할 |
| axios | 비동기 요청 라이브러리 |
| cors | cross origin issue를 해결하기 위한 라이브러리 |
| dotenv | 환경변수 분리를 위한 라이브러리 |

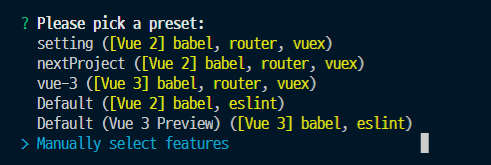
****

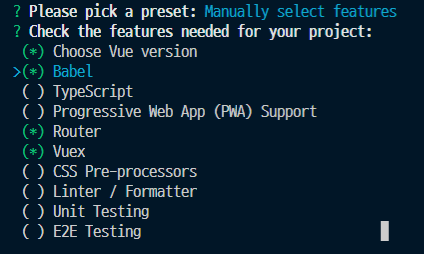
마지막으로 index.js까지 생성해주면 backend에 대한 기본 셋팅이 완료되었다.

* 1. **프론트 엔드 구축**

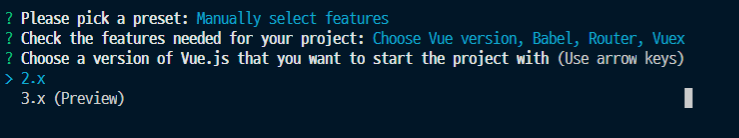
vue-cli를 활용해서 vue 기본 셋팅을 진행해 준다.



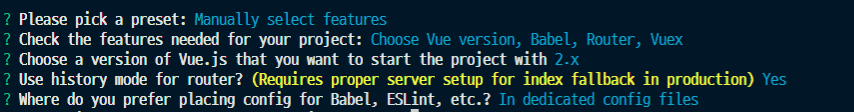
****



기초 preset을 choose vue version, babel, router, vuex를 체크해준다.



vue의 버전은 2.x 버전대를 사용한다.

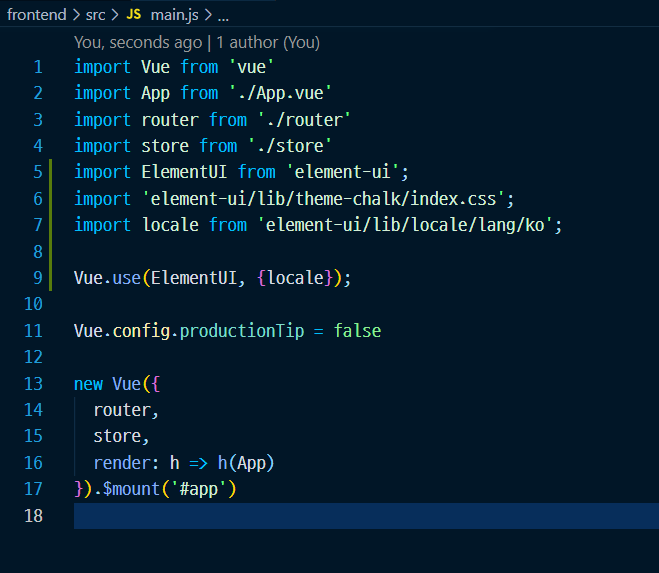


위와같이 세팅하면 자동으로 vue-cli가 기초 vue에 대한 세팅을 진행해준다.

이제 필요한 라이브러리 들을 다운받아 준다.

|  |  |
| --- | --- |
| Pacakge 이름 | 내용 |
| element-ui | vue 프론트엔드 프레임 워크 |
| axios | 비동기 요청 라이브러리 |
| chart.js/vue-chartjs | chart 라이브러리 |
| moment | date parsing 라이브러리 |

main.js를 수정해준다.

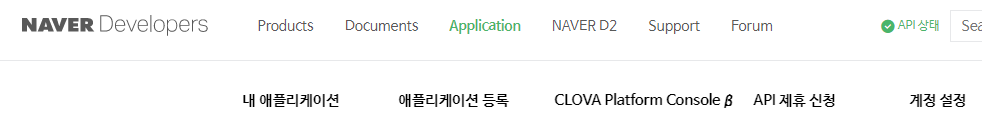


여기까지 진행하면 backend와 frontend의 기초 셋팅은 마무리된다.

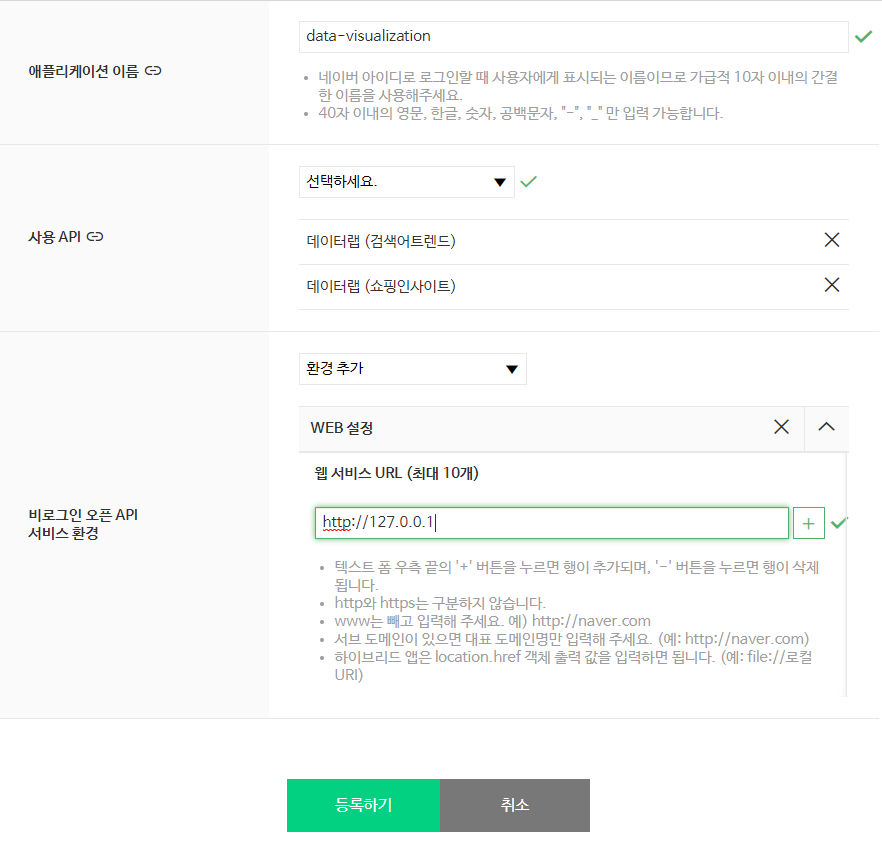
* 1. **API 사용하기**

네이버의 datalab API를 활용한다.

<https://developers.naver.com/main/> 해당 사이트에 접속해서 로그인 후



애플리케이션 등록을 진행해준다.

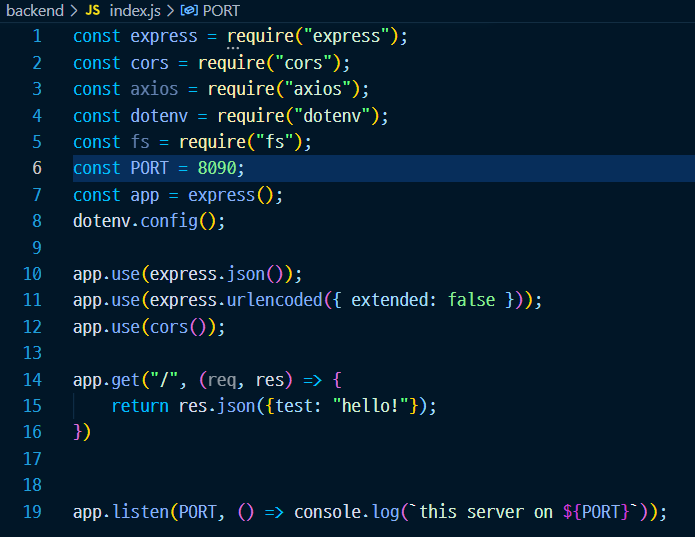


위와같이 사용 API를 설정해주고 환경에 WEB 환경

<http://127.0.0.1> 을 추가해준다.

개발을 위한 기본 구축은 마무리되었다.

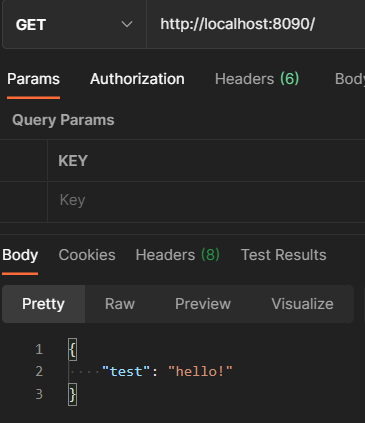
1. backend 구축하기



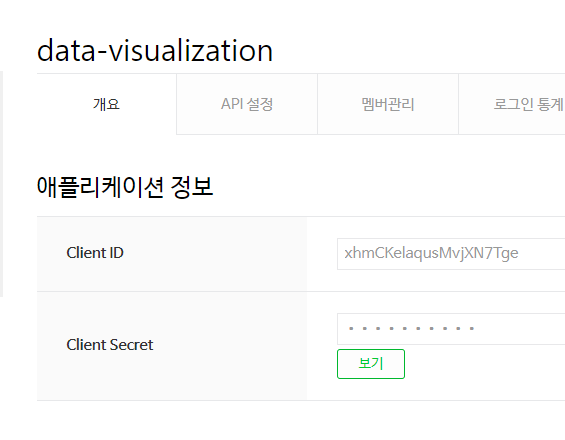
express를 위한 셋팅을 구축해준다.

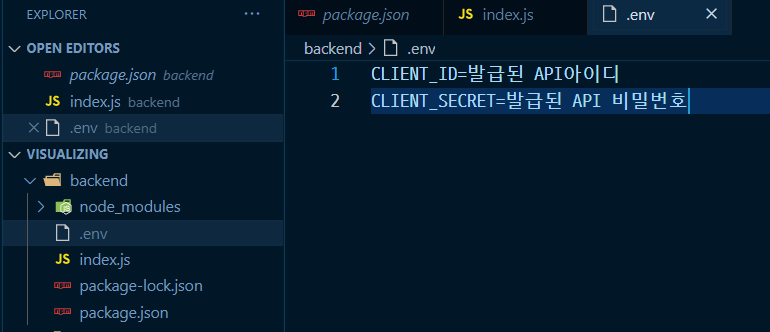


서버를 기동시켜 준 뒤 postman을 활용해서 서버의 동작 여부를 확인한다.

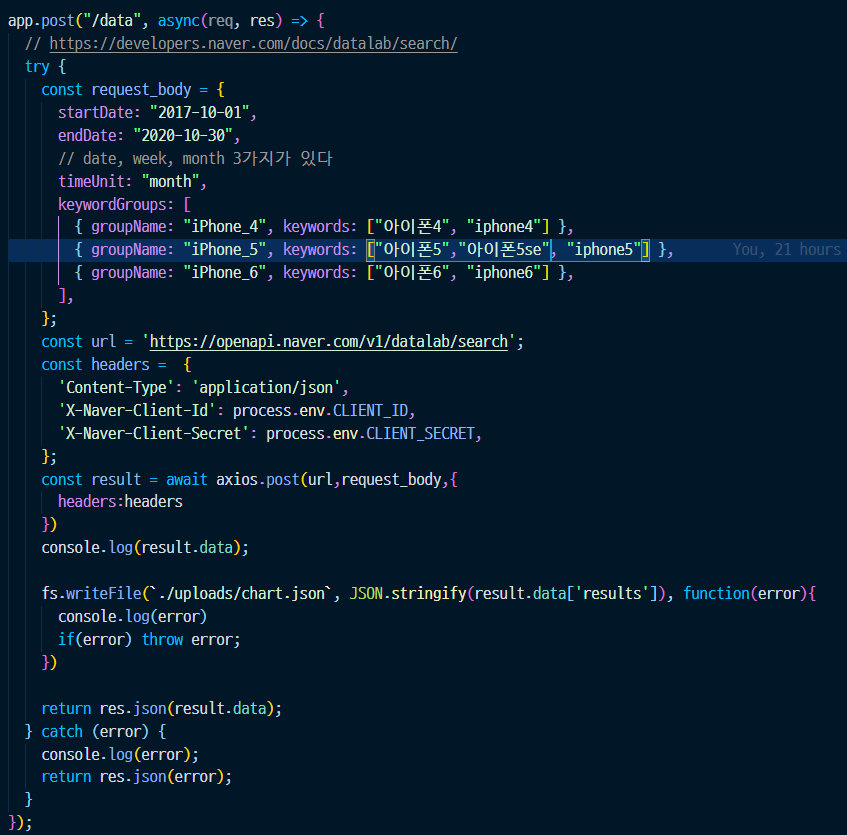


.env 파일을 생성해서 발급된 API ID와 비밀번호를 기입한다.



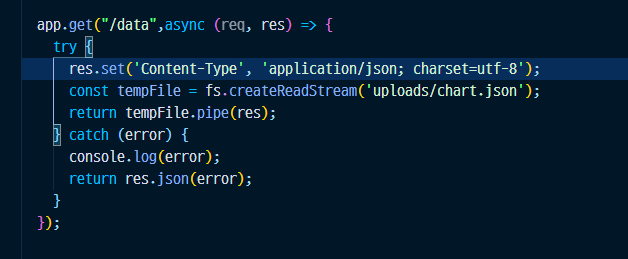


이제 data를 받아오는 app.post를 작성해 본다.



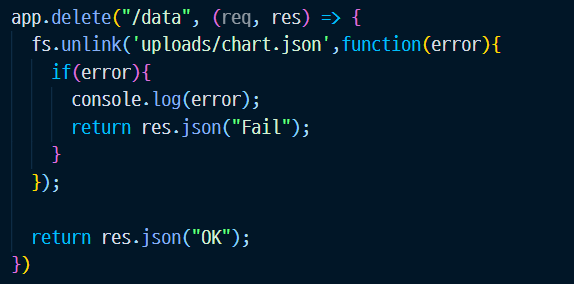
요청에 성공시 chart.json 을생성한다. 여기서 주의 할 점은 chart.json은 단순 데이터를 저장하고 보여주는 서버에 대한 부하를 덜어주기위한 캐싱의 목적으로만 사용하고 있다. 가져온 데이터의 내용을 변경하거나 추가하면 안되는 것에 유의한다.

이제 get을 통해 파일을 읽어오는 로직을 작성한다.

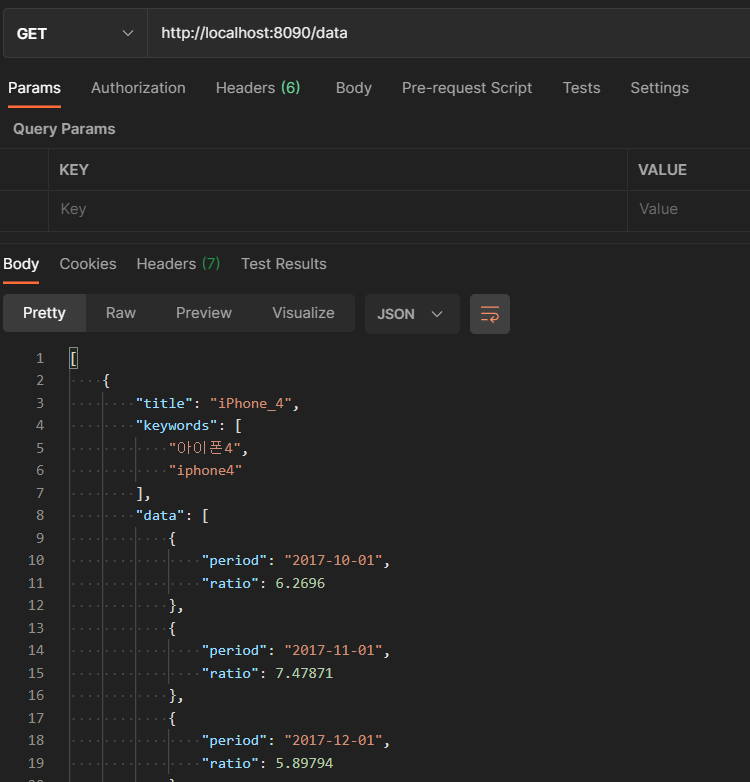


post 요청시에 만들어진 chart.json을 단순 리턴해주는 로직이다.

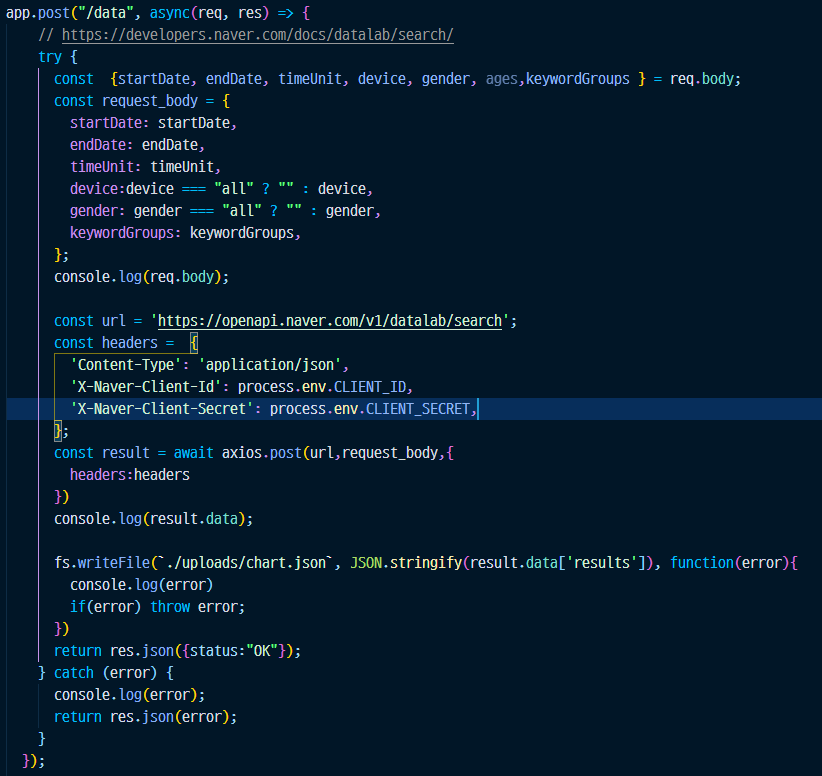
delete 요청 또한 만들어 준다.



모든 로직은 생성 후에 POSTMAN을 통해 테스트해본다.



이제 요청에 따라 데이터를 맞게 요청 할 수 있게 app.post(‘/data’)를 수정해준다.





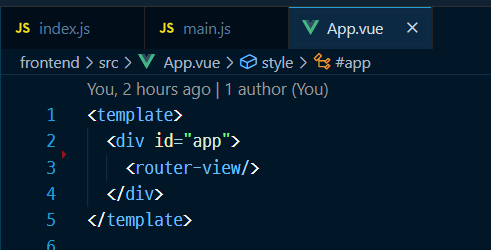
API Document( <https://developers.naver.com/docs/datalab/shopping/> )를 기반으로

backend 를 수정하였다.

백엔드 구축은 이로써 마무리 된다

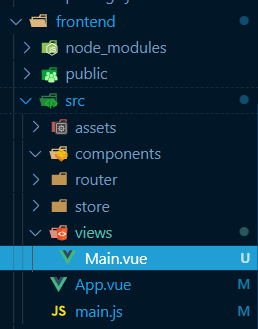
1. **프론트엔드 구축**

먼저 App.vue를 수정해 준다.



기존의 router-link 부분들은 제거해 주었다.

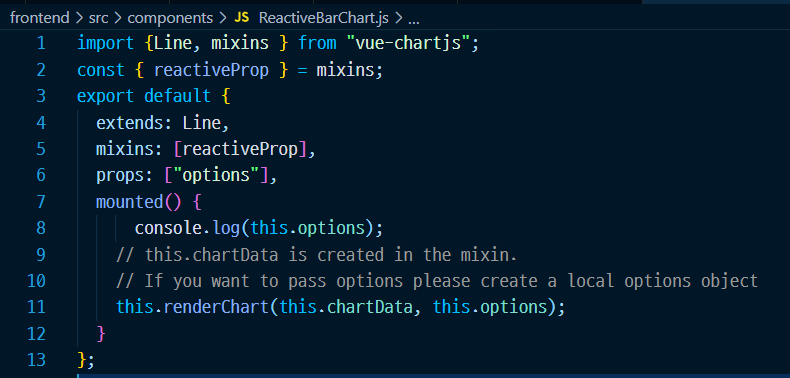
또한 views 폴더에 기존에 있던 Home.vue, About.vue를 삭제해주고 router 단에서도 삭제해준다.



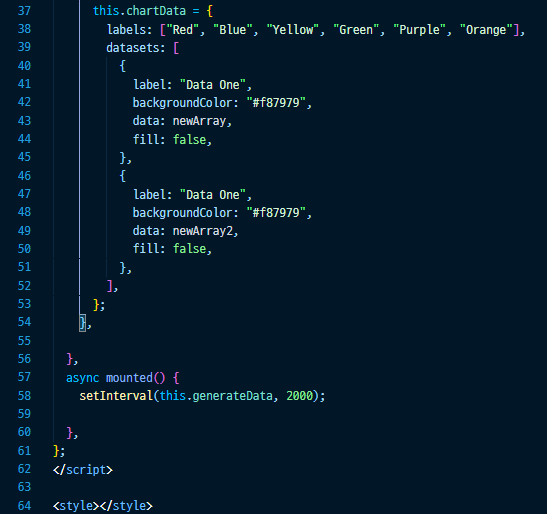
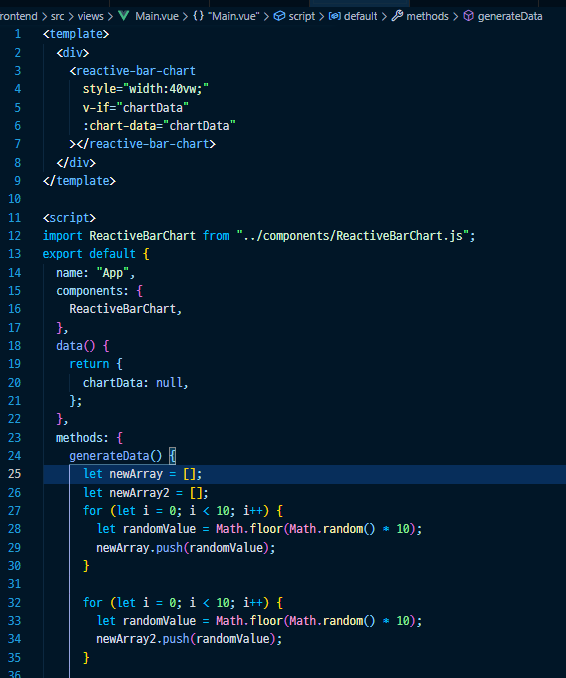
또한 Main.vue를 views폴더에 및 라우터에 추가해준다.

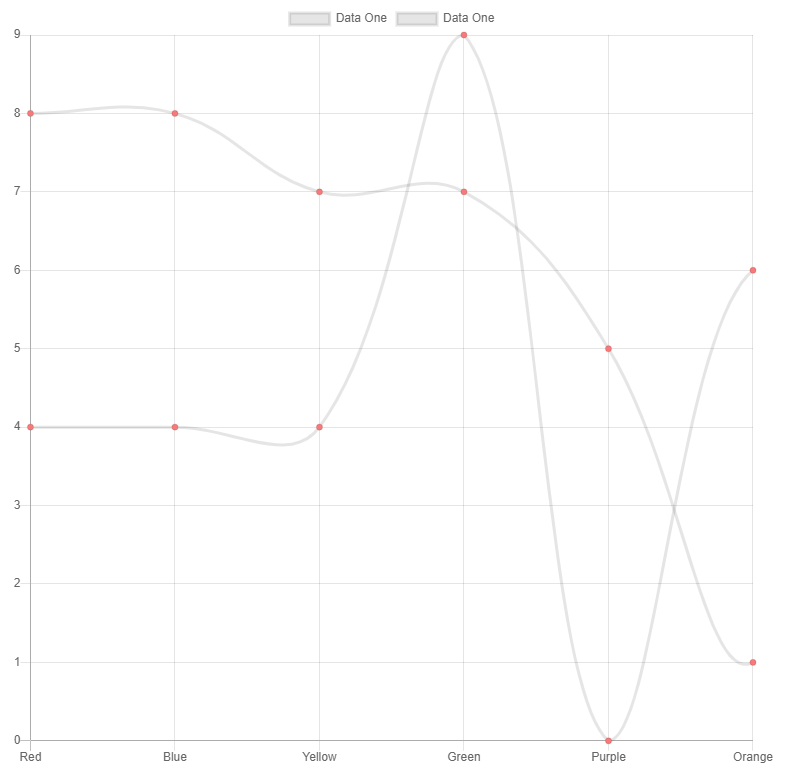


먼저 Chart를 위한 ReactiveBarChart.js를 componets 폴더에 생성해 준다.



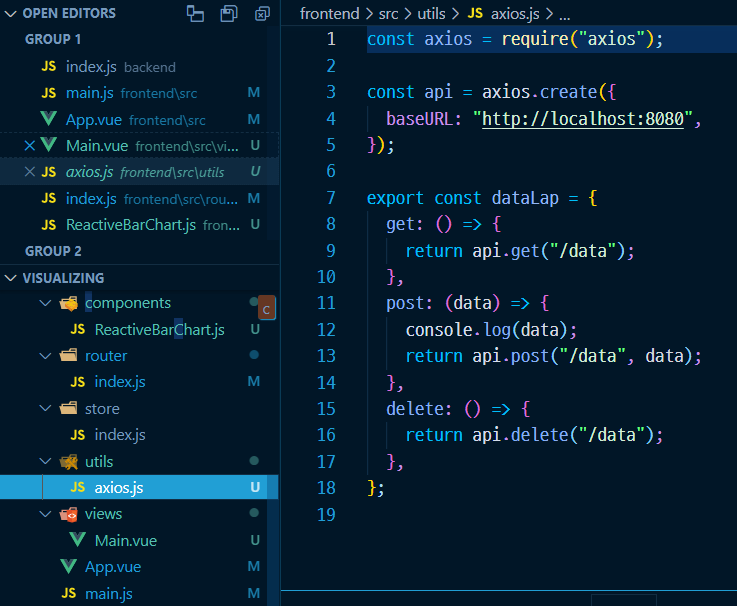
위를 기반으로 Main.vue를 작성해 준다.



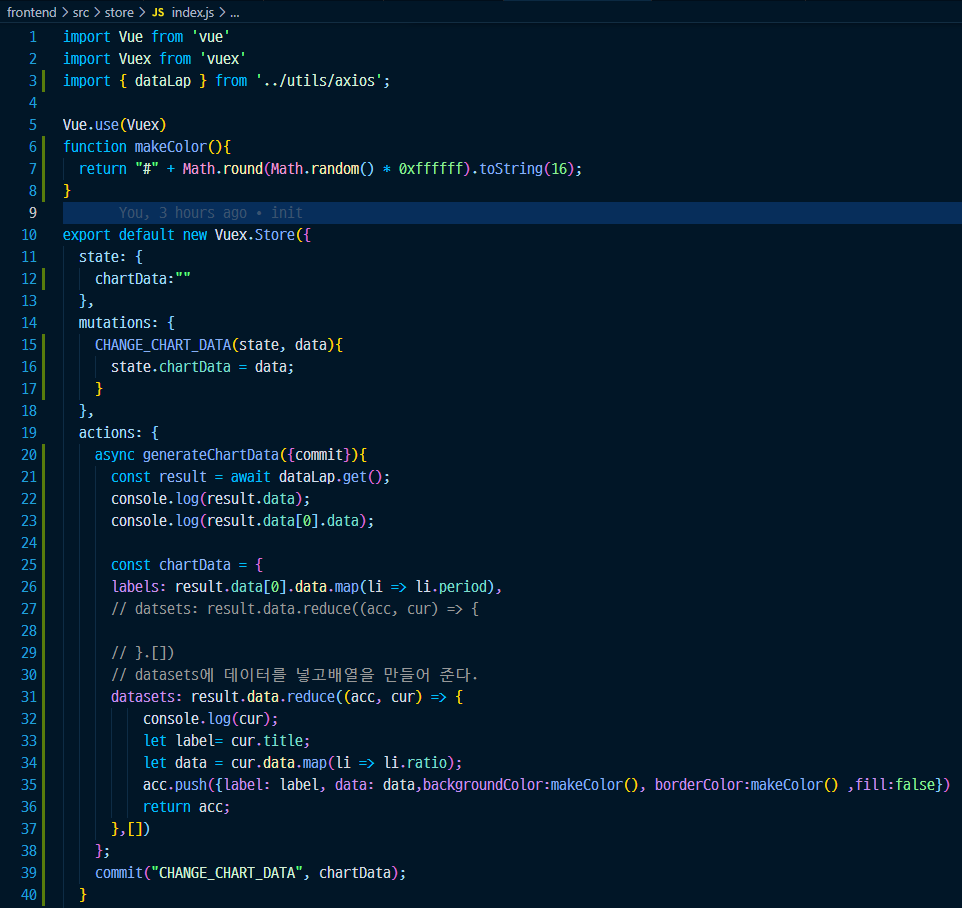


위와 같이 차트가 계속 갱신되면 1차 차트 구현이 완료된 상태이다.

이제 backend 요청을 위한 api를 작성해 준다.



위를 기반으로 vuex에 store을 수정한다.



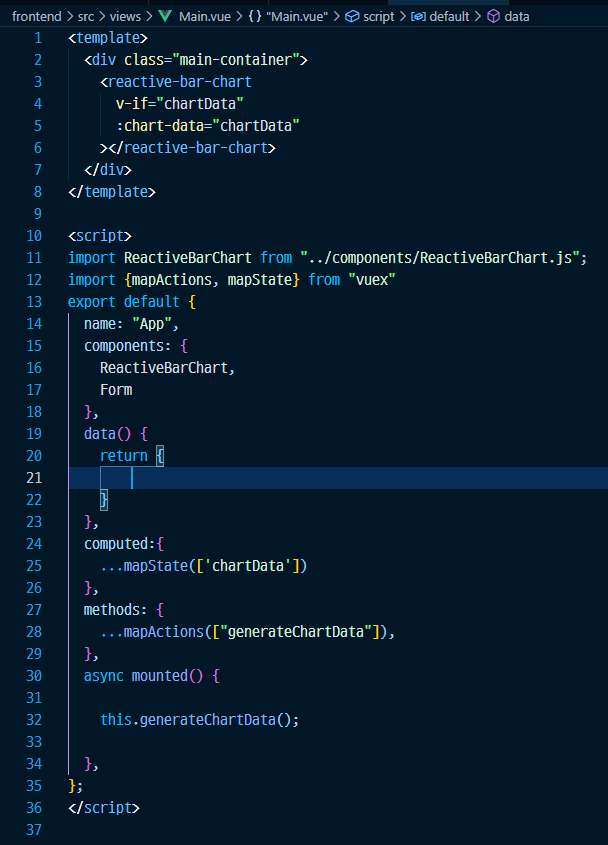
state, mutations, actions를 구성한다

action에서 비동기 요청 수행 -> mutation을 통해 state 변화 순서로 진행된다.

이제 Main.vue를 수정해준다.

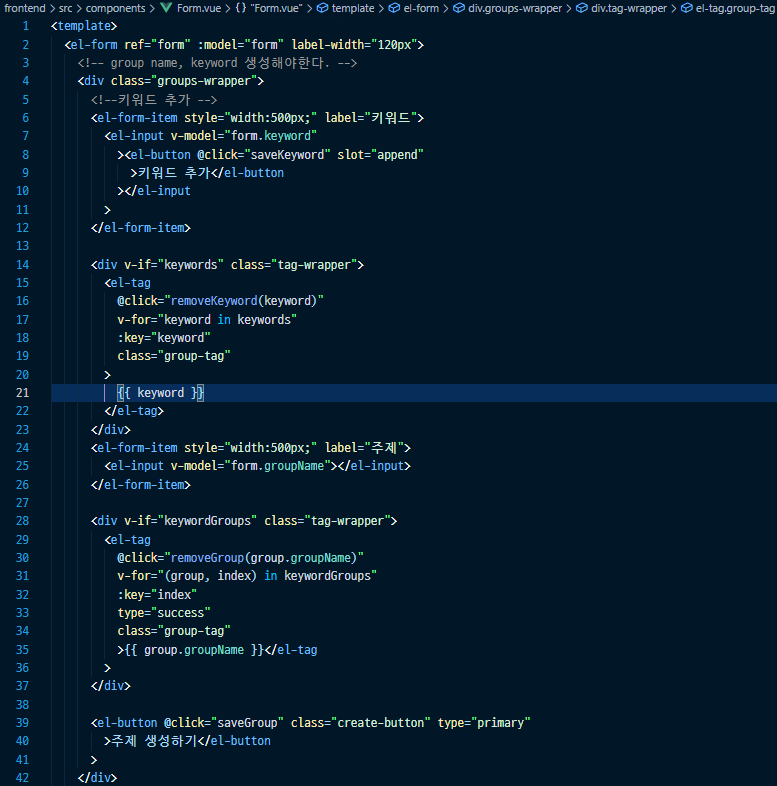


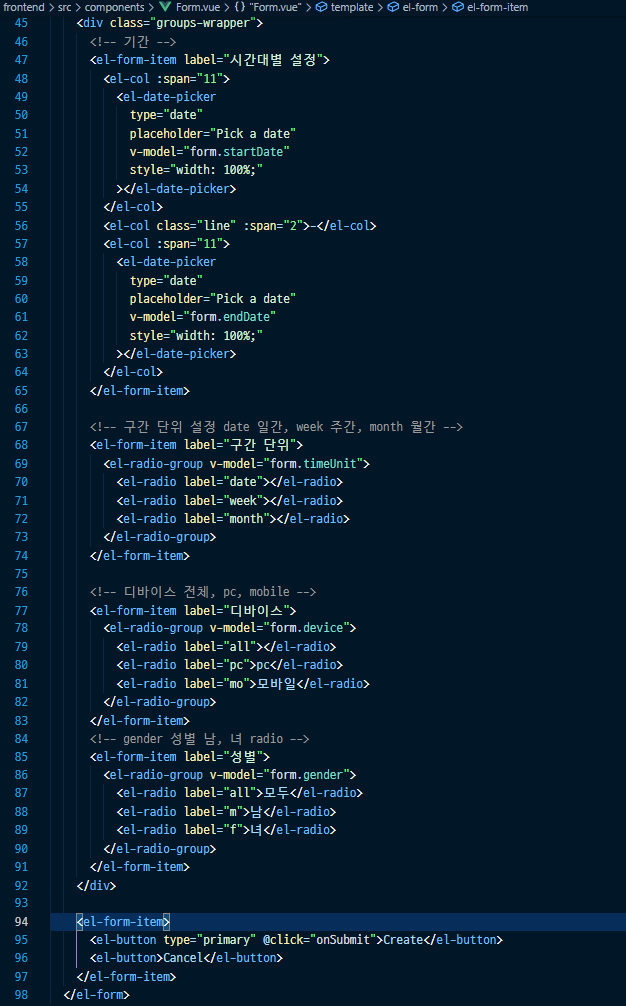
이제 Main.vue를 수정해준다.

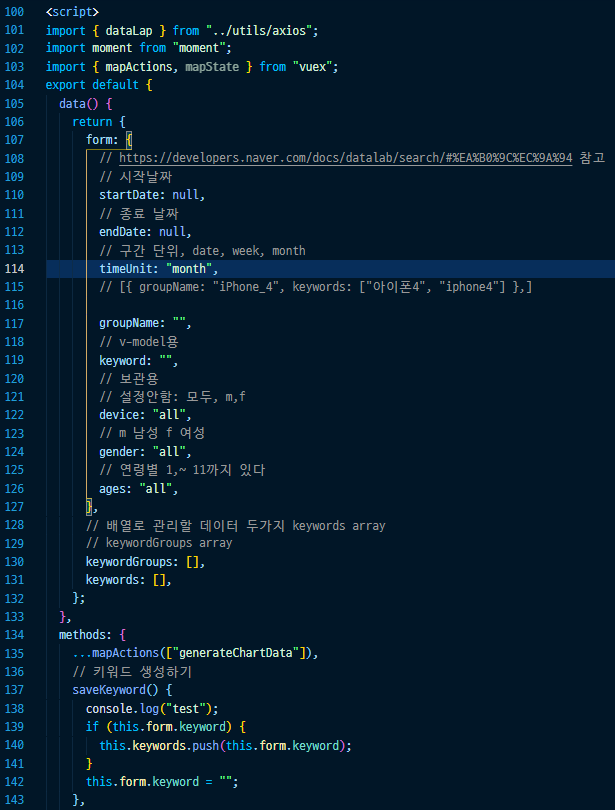


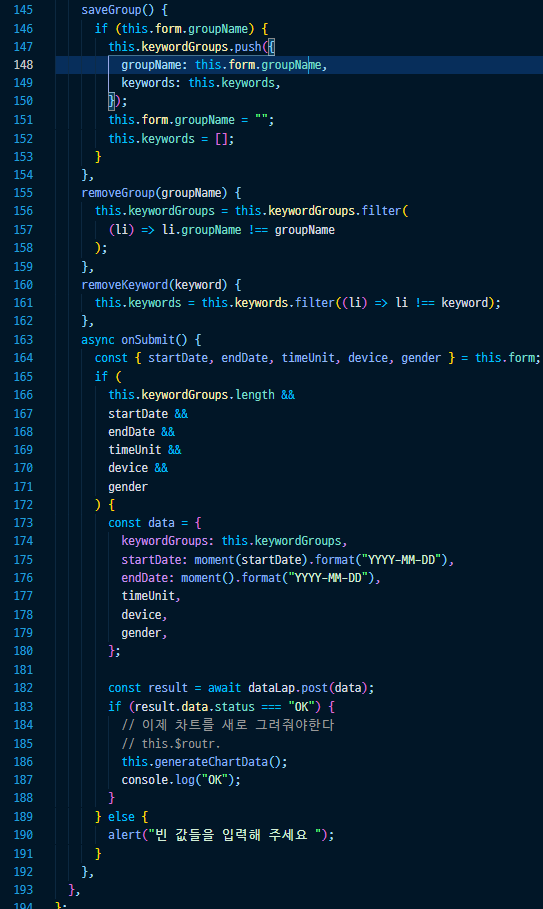
마지막으로 Form을 추가해준다.

해당 Form태그는 backend API에 맞게 데이터를 전송하기 위한 용도이다.

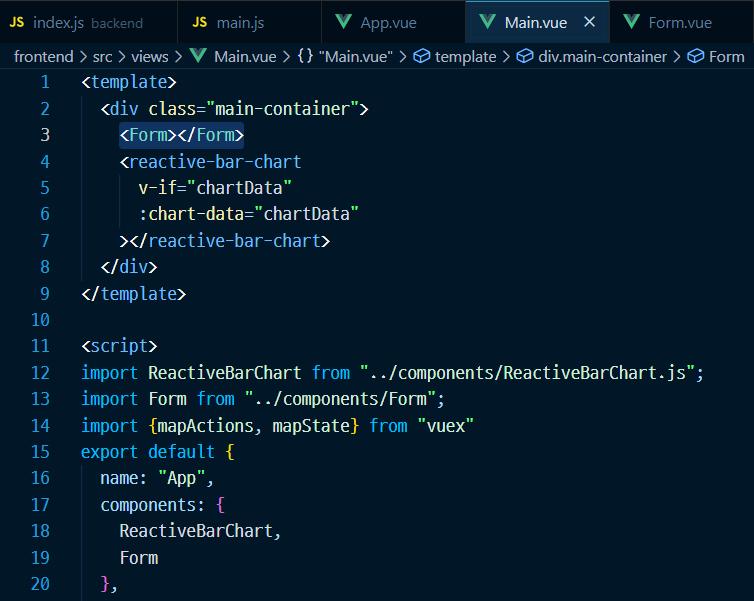








마지막으로 Main.vue에 방금 만든 Form.vue를 추가해준다.





**심화 과제**

검색조건에 ages 추가하기

****



API를 분석해서 나이를 추가해서 검색조건을 추가해보자(배열로 값을 전송해야한다)