1. state

Cosntructor(props){

super(props);

}

this.state = {

}

상태 관리를 위해 초기화(Render 되기전에 초기화)

**서버 요청 데이터 받기**

fetch // 기본방법

axios // 라이브러리

req : 서버주소 // http 메소드

(공식문서 참조 <https://developer.mozilla.org/ko/docs/Web/API/Fetch_API/Using_Fetch>)

[HTTP 기반 API](https://developer.mozilla.org/ko/docs/Web/HTTP/Overview#http_%EA%B8%B0%EB%B0%98_api)

HTTP 기반으로 가장 일반적으로 사용된 API는 [user agent](https://developer.mozilla.org/ko/docs/Glossary/User_agent)와 서버간에 데이터를 교환하는데 사용될 수 있는 [XMLHttpRequest](https://developer.mozilla.org/ko/docs/Web/API/XMLHttpRequest) API 입니다. 최신 [Fetch API](https://developer.mozilla.org/ko/docs/Web/API/Fetch_API)는 보다 강력하고 유연한 기능을 제공합니다.

//상태

useState[resData, setResData] = useState(null);

//useEffect를 사용하지 않으면 fetch가 렌더링 될 때 마다 실행

//useEffect - 컴포넌트 실행시만 fetch 실행

useEffect(() => {

  fetch('주소')

    .then((response) => { response.json() })  //json 형태로 정제

    .then((data) => setResData(data))  //안의 data값

}, []);

//출력

{

  resData.map((list, idx) => {

    <ul>

      <li key={idx}>{list.value}</li>

      <li key={idx}>{list.value2}</li>

      <li key={idx}>{list.value3}</li>

      <li key={idx}>{list.value4}</li>

    </ul>

  })

}

* **CORS 에러**
* **origin(출처)가 다르다(포트)**
* **서버에서 CORS 해결 – 라이브러리**
* **npm I cors (서버에서)**
* **const cors = requird(‘cors’)**
* app.use(cors())

**데이터 보내기**

데이터 입력

        <form onSubmit={onSubmitFunc} name='' method='' action=''>

          <input name="id"></input>

          <input name="title"></input>

          <input name="cont"></input>

          <input name="done"></input>

          <input type="sumbit">add</input>

        </form>

입력한 데이터 보내기

//서버에 보낼 데이터 저장

//form 에서 데이터 받기

const onSubmitFunc = (e) => {

  e.preventDefault(); // 기본동작막기 submit태그에서 get요청을 보냄

  const id = e.target.id;

  const title = e.target.title;

  const cont = e.target.cont;

  const done = e.target.done;

  //서버에 요청

  fetch('주소', {method : 'POST',

  headers: {    "Content-Type": "application/json",  },

  body: JSON.stringify({

      id,

      title,

      cont,

      done

    })

  })

}

**axios**

**데이터 받기**

useEffect(() => {

  axios.get('주소').then((response) = setResData(reponse.data));

}, []);

**데이터보내기**

//서버에 보낼 데이터 저장

//form 에서 데이터 받기

const onSubmitFunc = (e) => {

  e.preventDefault(); // 기본동작막기 submit태그에서 get요청을 보냄

  const id = e.target.id;

  const title = e.target.title;

  const cont = e.target.cont;

  const done = e.target.done;

  //서버에 요청

  axios.post('주소', { id, title, cont, done })

}

**react-query 라이브러리 사용하면 편함**