

```

import React, { useState, useEffect } from 'react'

//import './TravelsForm.css'

//import styles from './TravelsForm.module.css'
//import Travels, { travels } from './Travels'
//import { travels as tarr } from './Travels';
const TravelsForm = () => {
  //const [travels, setTravels] = useState(tarr);
  const [travels, setTravels] = useState([]);
  const [insertMode, setInsertMode] = useState(true);
  const [inputUpdateForm, setInputUpdateForm] = useState(
    {
      id: '', name: '', image: '', key: ''
    }
  );

  useEffect(
    () => {
      fetch('http://localhost:3100/travels') // Promise처리, GET /travels Request
      처리(Read)
        .then(response => response.json())
        .then(jsonData => setTravels([...jsonData]))
    },
    []
  );

  // 수정폼: 콤포넌트 내부에 자식 컴포넌트 구현
  const UpdateForm = (props) => {
    return (
      <>
        <form
          onSubmit={
            e => {
              e.preventDefault();
              const _id = e.target.id.value;
              const _name = e.target.name.value;
              const _image = e.target.image.value;
              console.log(_id, _name, _image);

              // 일반적인 배열 데이터의 수정 처리-> 인덱스 기반의 데이터 액세스
              // _id값과 인덱스값 -1 차이

              // [1] 먼저 기존 배열 데이터를 복사해서 새 배열 데이터로 작성 [...기
              존배열(travels)]
              // [2] 수정 처리할 새 t 객체를 하나 생성 --> t(travel) 객체
              // [3] 배열을 map함수로 t.id가 같은 경우 수정처리
              // [3-1] 새배열 수정
              // [3-2] 기존 travels 배열을 setter로 업데이트
              // [4] 모드 변경 : insertMode 토글 처리
              //[1]
              const newTravels = [...travels]

```

```

    // [2]
    const updatedTravel = {
      key: crypto.randomUUID(), // 에러해결: 추가 입력
      id: _id,
      name: _name,
      image: _image
    };
    //const id = updatedTravel.id;

    // update처리
    fetch(`http://localhost:3100/travels/${updatedTravel.id}`, {
      method: 'PATCH', // PUT
      headers: { 'Content-type': 'application/json' },
      body: JSON.stringify(updatedTravel),
    })
      .then(res => { console.log(res) });

    // [3-1]
    travels.map(t =>
      t.id === _id ? (newTravels[t.id - 1] = updatedTravel) : (t)
    )
    // [3-2]
    setTravels(newTravels);
    // [4]
    setInsertMode(!insertMode);
  }
}
>
<div className="input-group input-group-lg mb-2">
  <span className="input-group-text">인덱스 번호</span>
  <input type="text" className="form-control" name='id'
autoComplete='off' defaultValue={props.travel.id} />
</div>
<div className="input-group input-group-lg mb-2">
  <span className="input-group-text">여행지 국가</span>
  <input type="text" className="form-control" name='name'
autoComplete='off' defaultValue={props.travel.name} />
</div>
<div className="input-group input-group-lg mb-2">
  <span className="input-group-text">대표 이미지</span>
  <input type="text" className="form-control" name='image'
autoComplete='off' defaultValue={props.travel.image} />
</div>
<div className="input-group input-group-lg mb-2">
  <img src={props.travel.image} alt={props.travel.name} height='150px'
/>
</div>
<input type="submit" className="btn btn-primary" value="수정하기" />
<hr />
</form>
<div
  style={{
    border: '1px solid blue', position: 'relative'
  }}

```

```

    >
    <button
      onClick={e => {
        //window.location.reload(); // 여행지정보의 리셋으로 수정한 내용은 반영
        안됨
        setInsertMode(!insertMode)
      }}
      className='btn btn-outline-info'
      style={{ position: 'absolute', top: '-68px', right: '0' }}
    >
      뒤돌아가기
    </button>
  </div>
  <br />
  <br />

  </>
);
}

const clickUpdateBtn = (params, event) => {
  event.preventDefault();
  console.log('수정! ', params);
  setInsertMode(!insertMode);

  setInputUpdateForm({
    id: params.id,
    name: params.name,
    image: params.image
  })
}

const clickDeleteBtn = (id, e) => {
  e.preventDefault();
  console.log("삭제! ", id);
  console.log(typeof id)
  // Delete처리
  fetch(`http://localhost:3100/travels/${id}`, {
    method: 'DELETE',
    headers: { 'Content-type': 'application/json' },
  })
  .then((res => { console.log(res) })))

  setTravels(
    travels.filter(t => t.id !== id)
  );
}

return (
  <div className="container mt-5">
    <h3>여행지 정보 목록</h3>
    <hr />
    {

```

```

insertMode ?
(
  <form
    onSubmit={
      event => {
        event.preventDefault(); // 새로고침(reload)방지
        const { id, name, image } = event.target;
        const _id = id.value;
        const _name = name.value;
        const _image = image.value;
        console.log(_id, _name, _image);
        const newArticle = {
          key: crypto.randomUUID(),
          id: _id,
          name: _name,
          image: _image,
        }
        console.log(newArticle);

        /*          console.log(travels);
                   travels.push(newArticle);
                   console.log(travels); */

        console.log(travels);

        // POST /travels (Create하는 endpoint)
        fetch('http://localhost:3100/travels', {
          method: 'POST', headers: { 'Content-type': 'application/json'
},
          body: JSON.stringify(newArticle), //params, query
        })
          .then(
            (res) => { console.log(res); }
          );

        setTravels(
          [...travels, newArticle]
        )
        console.log(travels);
        /*  event.target.id.value = '';
            event.target.name.value = '';
            event.target.image.value = ''; */
        event.target.reset();

      }
    }
  >
  { /* <p><input className={styles} type="text" name='name'
placeholder='여행지 국가 이름' autoComplete='off' /></p>
    <p><input className={styles} type="text" name='image' placeholder='여행지
대표이미지' autoComplete='off' /></p> */
    <div className="input-group input-group-lg mb-2">
      <span className="input-group-text">인덱스 번호</span>

```

```

        <input type="text" className="form-control" name='id'
autoComplete='off' />
      </div>
      <div className="input-group input-group-lg mb-2">
        <span className="input-group-text">여행지 국가</span>
        <input type="text" className="form-control" name='name'
autoComplete='off' />
      </div>
      <div className="input-group input-group-lg mb-2">
        <span className="input-group-text">대표 이미지</span>
        <input type="text" className="form-control" name='image'
autoComplete='off' />
      </div>
      <input type="submit" className="btn btn-primary" value="여행지 정보
입력하기(기존에 밑 추가)" />
      <hr />
      {/* {console.log(travels)} */}
      <ul style={{ listStyle: 'none', paddingLeft: '40px' }} >
        {/* <li>
          <span style={{ display: 'inline-block', width: '140px' }}>1.</span>
          <span style={{ display: 'inline-block', width: '140px', fontWeight:
'bold' }}>Korea</span>
          <img src='https://cdn.pixabay.com/photo/2022/01/13/07/05/house-
6934535_960_720.jpg' width="50%" />
        </li> */}
        {
          travels.map(
            t => (
              <li key={t.key}
                style={{
                  border: '1px solid', position: 'relative'
                }}
              >
                <span>{t.id}</span>
                <span
                  style={{
                    display: 'inline-block',
                    width: '140px',
                    fontWeight: 'bold',
                    marginBottom: '70px',
                    textTransform: 'capitalize', // korea -> Korea
                  }}
                >{t.name}</span>

                <img src={t.image} height='140px' alt={t.name} />
                <div>
                  <button
                    className='btn btn-outline-warning'
                    style={{
                      position: 'absolute', top: '50px', right: '102px',
zIndex: '2'

                    }}
                    onClick={e => { clickUpdateBtn(t, e) }}
                >

```

```
        Update
      </button>
      <button
        className='btn btn-outline-danger'
        style={{
          position: 'absolute', top: '50px', right: '24px'
        }}
        onClick={(e) => { clickDeleteBtn(t.id, e) }}
      >
        Delete
      </button>
    </div>

  </li>
)
)
}
</ul>
</form>
) :
// (<h1>수정화면</h1>)
(<UpdateForm travel={inputUpdateForm} />)
}
</div>
)
}

export default TravelsForm
```