



리액트와 함께 장고 시작하기 / 리액트

Context API와 Reducer 패턴

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

useContext 훅

Context API의 Consumer 컴포넌트와 유사한 활용

```
const PostContext = React.createContext();

const PostDetail = () => {
  const post = React.useContext(PostContext);
  return (
    <div>{JSON.stringify(post)}</div>
  )
};

const App = () => {
  const post = { title: "제목", content: "내용" };
  return (
    <PostContext.Provider value={post}>
      <PostDetail />
    </PostContext.Provider>
  );
};
```

useReducer 훅

컴포넌트의 상태값을 리덕스의 리듀서처럼 관리 가능

```
const INITIAL_STATE = { name: 'Tom', age: 10 };
```

```
const reducer = (state, action) => {  
  switch ( action.type ) {  
    case 'setName': return { ...state, name: action.name };  
    case 'setAge': return { ...state, age: action.age };  
    default: return state;  
  }  
};
```

```
const Person = () => {  
  const [state, dispatch] = React.useReducer(reducer, INITIAL_STATE);  
  return (  
    <div>  
      <p>name : {state.name}</p>  
      <p>age: {state.age}</p>  
      <input type="text" value={state.name} onChange={e => dispatch({ type: 'setName', name: e.currentTarget.value })} />  
      <input type="text" value={state.age} onChange={e => dispatch({ type: 'setAge', age: e.currentTarget.value })} />  
    </div>  
  );  
};
```

Context API를 통해,
깊은 트리의 자식 컴포넌트에
dispatch 전달을 할 수 있겠다.

Life is short.
You need Python and Django.

I will be your pacemaker.

