



# React Context API 개요

## React Context API란?

컴포넌트 트리 상에서 props를 전달하지 않고 필요한 data를 컴포넌트에 전달하는 방법을 제공한다.

### 주의사항

- 단지 몇몇 컴포넌트에서 props를 전달하는 것을 피하기 위하여 사용하지 않을 것!
  - Redux나 Mobx를 사용할 수 있음. Redux나 Mobx의 내부는 Context API로 이미 구성되어 있다.
- 여러 계층에 걸쳐 많은 component 들이 동일 데이터에 접근해야 하는 경우에 사용
- 정말 Global Data 인 경우에 사용

### 장점

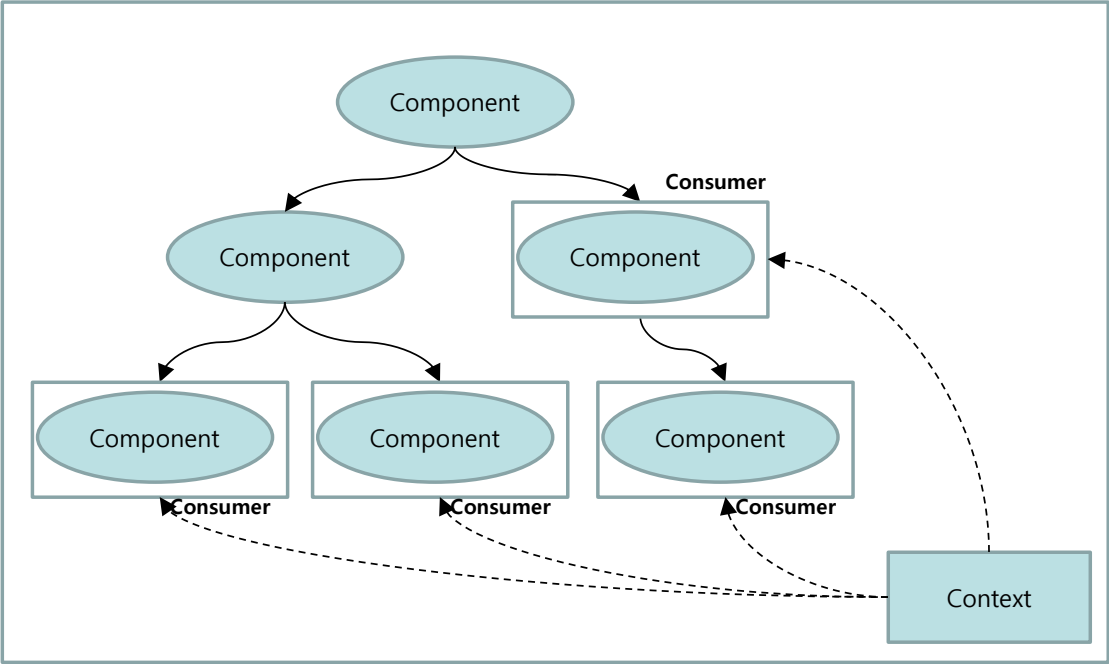
- props - props - props로 전달하지 않아도 됨.
- 공유 상태, 액션을 이용하고 싶을 때는 Consumer를 이용해 접근한다.

## 예제

• todolist-context-api-test 프로젝트(7장 예제)



Provider



# Provider, Context 객체 생성



## src/context/ToDoListContext.js 확인 검토

```
import React, { Component } from 'react';

const ToDoListContext = React.createContext();
class ToDoListProvider extends Component {
  constructor(props) {
    super(props);
    this.state = { todolist : [ ..... (생략) ] }
  }
  addTodo = (todo) => { ..... (생략) }
  deleteTodo = (no) => { ..... (생략) }
  toggleDone = (no) => { ..... (생략) }
  render() {
    const values = {
      state : this.state,
      actions : { addTodo:this.addTodo, deleteTodo:this.deleteTodo, toggleDone:this.toggleDone }
    }
    return (
      <ToDoListContext.Provider value={values}>
        {this.props.children}
      </ToDoListContext.Provider>
    );
  }
}
const ToDoListConsumer = ToDoListContext.Consumer;
export { ToDoListProvider, ToDoListConsumer };
```

# Provider 적용



index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import "bootstrap/dist/css/bootstrap.css";
import "./index.css";
import App from "../components/App";
import { TodoListProvider } from "../context/TodoListContext";

import * as serviceWorker from './serviceWorker';

ReactDOM.render(
  <React.StrictMode>
    <TodoListProvider>
      <App />
    </TodoListProvider>
  </React.StrictMode>,
  document.getElementById('root')
);
serviceWorker.unregister();
```

## Consumer 적용(1)



### App.js

- App이 직접 데이터를 이용하지 않기 때문에 props나 state가 없음
- 자식 컴포넌트로 props를 전달하지 않음.

```
import React from "react";
import InputTodo from "../InputTodo";
import TodoList from "../TodoList";

const App = () => {
  return (
    <div className="container">
      <div className="card card-body">
        <div className="title">:: Todolist App</div>
      </div>
      <div className="card card-default card-borderless">
        <div className="card-body">
          <InputTodo />
          <TodoList />
        </div>
      </div>
    </div>
  );
};
export default App;
```

## Consumer 적용(2)



### TodoList.js

```
import React from "react";
import { TodoListConsumer } from "../context/TodoListContext";
import TodoItem from "./TodoItem";

const TodoList = () => {
  return (
    <TodoListConsumer>
      { (value) => (
        <div className="row">
          <div className="col">
            <ul className="list-group">
              { value.state.todolist.map(item => (
                <TodoItem key={item.no} todoItem={item}
                  deleteTodo={value.actions.deleteTodo} toggleDone={value.actions.toggleDone} />
              ))}
            </ul>
          </div>
        </div>
      )}
    </TodoListConsumer>
  );
};

export default TodoList;
```

# useContext hook(1)



## Context API의 value를 쉽게 이용할 수 있도록

- todolist-context-api2 예제 검토
- `const value = useContext(context)`
  - 전달 인자는 Provider나 Consumer가 아닌 Context이다.

## src/context/TodoListContext.js 변경 내용 검토

```
import React, { Component } from "react";
import produce from "immer";

const TodoListContext = React.createContext();

class TodoListProvider extends Component {
  .....(생략)
}
export { TodoListProvider };
export default TodoListContext;
```

## useContext hook(2)



### 기존 코드 : 예) TodoList 컴포넌트

```
import React from "react";
import { TodoListConsumer } from "../context/TodoListContext";
import TodoItem from "../TodoItem";

const TodoList = () => {
  return (
    <TodoListConsumer>
      {value => (
        <div className="row">
          .....(선택)
        </div>
      )}
    </TodoListConsumer>
  );
};
export default TodoList;
```



## useContext hook(3)



### 변경된 코드

```
import React, { useContext } from "react";
import TodoListContext from "../context/TodoListContext";
import TodoItem from "./TodoItem";

const TodoList = () => {
  const value = useContext(TodoListContext);

  return (
    <div className="row">
      <div className="col">
        <ul className="list-group">
          {value.state.todolist.map(item => (
            <TodoItem key={item.no} todoItem={item}
              deleteTodo={value.actions.deleteTodo} toggleDone={value.actions.toggleDone} />
          ))}
        </ul>
      </div>
    </div>
  );
};

export default TodoList;
```