차세대 챗봇 **Lab I**

Lecture-3 리액트 앱의 전역상태관리

Sungja Choi,
School of Computing, Gachon University
2023-1

가천대학교 Gachon University

학습내용

리액트의 기본내용에 대해 학습합니다.

전역상태관리 라이브러리

Recoil 설치

Atom (공유 상태)

selectors (순수함수)

리코일 프로젝트 적용기법

Lecute-3. 리액트 앱의 전역상태관리

리액트 프로젝트를 관리하기위한 리코일(Recoil)에 대해 학습합니다. 리코일은 페이스북에서 출시한 Flux 패턴을 사용하는 최신 전역상태관리 라이브러리 입니다.

[참조]
https://ko.reactjs.org/
https://recoiljs.org/

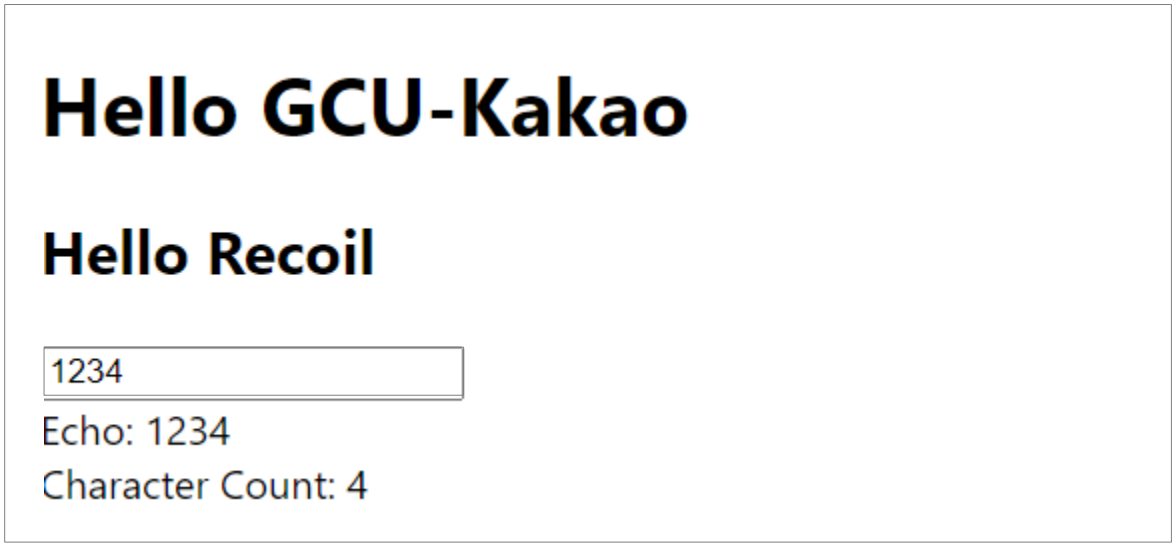
Practice

>> 리코일 라이브러리 설치
npm i recoil

실습-1 문자열 상태관리

리코일 전역 상태관리 라이브러리를 사용하여 입력으로 들어오는 문자열 길이를 반환

[실행화면]



Practice

Coding(1) 라이브러리 임포트 및 루트지정

>> 리코일을 설치했다면 리코일 사용을 위해 해당 모듈에 대해 임포트

- **RecoilRoot** : 최상위 루트에 정의하여 리코일 사용을 알림
- **Atom**: 상태를 저장
- **Selector**: atom의 변화된 값을 사용
- **Recoil Hook**
  useRecoilState
  useRecoilValue

App.js

```
import {
  RecoilRoot,
  atom,
  selector,
  useRecoilState,
  useRecoilValue
  } from "recoil";
…
return(
<RecoilRoot>
      <div className="App">
        <h1>Hello GCU-Kakao</h1>
        <h2>Hello Recoil</h2>
        <CharacterCounter />
      </div>
</RecoilRoot>
);
```

Practice

❶ **atom**을 사용해서 textState 상태변수를 지정, 초기값은 null
❷ **Selector**를 사용해서 고기능유의 ID를 정의해서 다른 곳에서 참조하도록 하고, **get**메소드를 정의해서 상태변수를 가져와서 문자열의 개수를 반환

Coding(2) Atom 및 Selector 정의

App.js

```
const textState = atom({❶
  key: "textState",
  default: ""
});

const charCountState = selector({❷
  key: "charCountState" ,  // unique ID (with respect to other atoms/selectors)
  get: ({get}) => {
    const text = get(textState);

    return text.length;
  },
});
```

Practice

>> useRecoilState를 사용하여 text 입력창에 리코일 textSate 상태변수를 지정

Coding(3) useRecoilState 훅 사용

App.js

```
function TextInput() {
  const [text, setText] = useRecoilState(textState); ❶

  const onChange = (event) => {
    setText(event.target.value);
  };

  return (
    <div>
      <input type="text" value={text} onChange={onChange} />
      <br />
      Echo: {text}
    </div>
  );
}
```

Practice

>> useRecoilValue를 사용하여 문자열 개수를 사용

Coding(4) useRecoilValue 훅 사용

App.js

```
function CharacterCount() {
  const count = useRecoilValue(charCountState);  ❶

  return <>Character Count: {count}</>;
}
```

Practice

Coding(5) 컴포넌트 구성

App.js

```
function CharacterCounter() {
  return (
    <div>
      <TextInput />
      <CharacterCount />
    </div>
  );
}
```

In-Class Exercises(1)

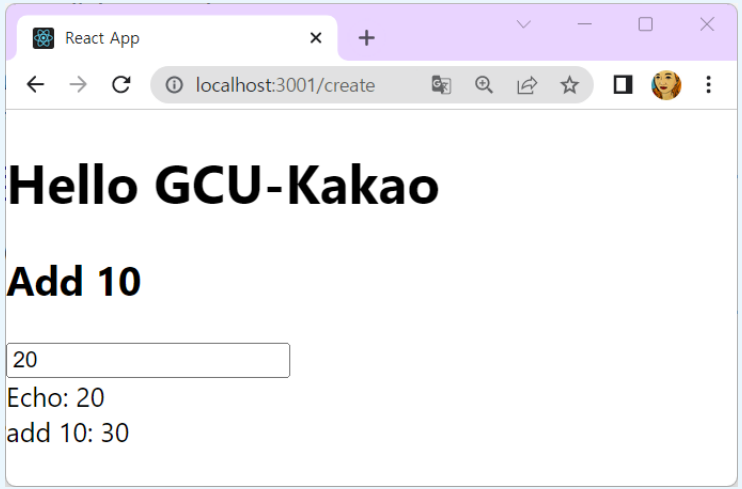
수업시간 내 진행하시기 바랍니다.

Add Ten 리코일 적용해보기

- 입력 받은 숫자에 10을 더해서 출력
- 리코일을 적용해서 상태관리
- 사이버캠퍼스 제출
- 평가(10점)

[REF.] Code reference

```
return <>add 10: {Number.parseInt(digit)+10}</>;
```

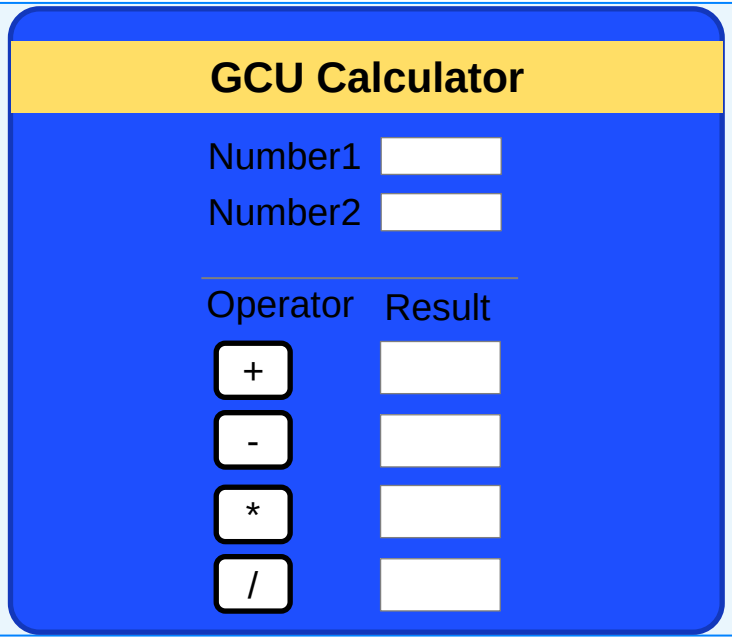


In-Class Exercises(2)

수업시간 내 진행하시기 바랍니다.

Calculator 제작

- 사칙연산을 수행
- 두수를 입력 받음
- 리코일 적용해서 상태관리
- 사이버캠퍼스 제출
- 평가(10점)

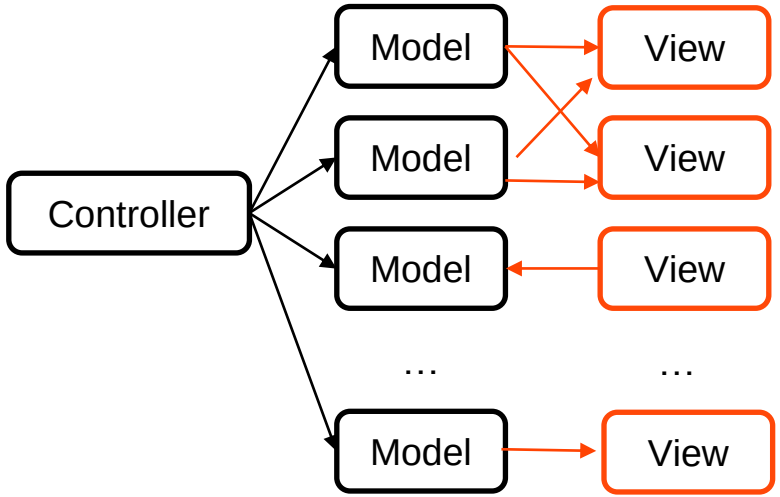


[REF.]

MVC 패턴

Model, View, Controller의 약자로 Model에 데이터를 저장하고, Controller를 이용하여 Model의 데이터를 관리합니다. Model의 데이터가 변경되면 View로 전달하거나, 사용자가 View를 통해 입력하면 모델을 업데이트하여 데이터가 **양방향**으로 흐를 수 있는 구조

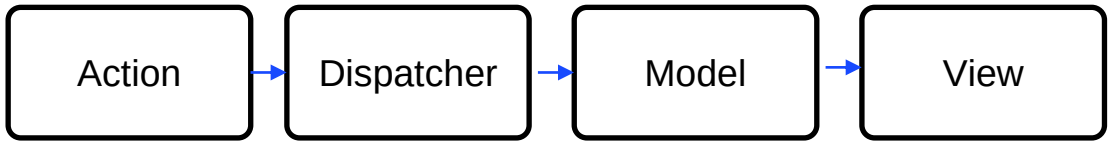
애플리케이션의 규모가 커지면 아래 그림과 같이 복잡한 구조로 됨



[REF.]

Flux 패턴

Facebook에서 MVC 패턴의 문제점 (뷰가 많아짐) 을 해결하기 위해 Flux 패턴 출시



- 상태관리 라이브러리를 제공하여 앱의 상태를 관리
- 리액트에서 단방향 데이터 흐름
- 싱글 페이지 어플리케이션 제공

"**리코일** 상태 관리 라이브러리 활용"

[REF.]

Recoil

Recoil은 Atom으로부터 시작해서 Selector를 거쳐 React 컴포넌트까지 전달되는 하나의 Data-Flow Graph를 생성

- **Atom**: 상태의 단위로 업데이트되면 Atom을 사용하는 있는 모든 컴포넌트들이 새로운 값으로 리-렌더링 되므로 상태를 동일하게 공유해야 함.
  - Key: atom을 구분하기위한 고유한 값
  - default: 상태를 초기화하거나 폴백 값(null, undefined)으로 세팅

- **Selector**: 상태를 변화하도록 제공하는 순수함수 (Pure function)
  순수함수는 모든 경우 입력에 따라 출력이 같아야 함.

- **Functions**
  - useRecoilValue: 값을 읽어오고 싶을 때
  - useSetRecoilState: 값을 업데이트하고 싶을 때
  - useRecoilState: 값을 읽고 쓰려고 할 때 사용

HOMEWORK

수업시간 내 진행하는 것을 원칙으로 합니다.

Lotto System

- 9개의 로또 숫자를 정의
- 임의의 7개의 숫자를 생성하고 사용자로부터 2개의 숫자를 입력
- 로또 숫자 매칭 모듈 제작 (예: 모두 맞추면 대박 출력, 3개이상이면 보너스 출력 등 자율)
- 사용자의 로또 번호를 정렬하는 모듈 제작, 정렬알고리즘은 자율 **(코딩훈련)**
- 리코일 상태관리 적용
- 개별 문제해결 1시간 후 팀별 업그레이드
- 팀별 제출
  - 결과화면 및 주요 코드를 파워포인트로 작성하여 PDF 제출
  - 실행파일 첨부 (node_modules 제외)
  - **팀원 개별 테스트 코드를 30분내 작성 인증샷 첨부** (팀장이 확인)
- 평가 (10점)

CONNECT.
SOLVE.
CREATE. + AI

가천대학교 Gachon University