

Day 7-1. Create_1 (Mar 2nd, 21)

create read update는 클릭했을 때 특정한 페이지로 이동해서 실행된다.

하지만 delete는 링크를 사용하면 문제가 생기는 경우가 생길 수 있기 때문에, 링크와 같은 페이지 개념이 아닌, 버튼과 같은 operation 개념을 사용한다.

App.js

```
<ul>
  <li><a href="/create">create</a></li>
  <li><a href="/update">update</a></li>
  <li><input type="button" value="delete"></input></li>
</ul>
```

위의 코드를 Control 컴포넌트로 따로 분리한다.

그 다음, Control 컴포넌트의 링크나 버튼을 클릭했을 때 각각 create, update, delete이벤트가 일어날 수 있도록 해준다.

OnChangeMode라는 이벤트를 Control 컴포넌트에 정의한다.

App.js

```
<Control onChangeMode={function(){
  }.bind(this)}></Control>
```

Control 컴포넌트의 create, update, delete을 각각 클릭했을 때, 각각 onChangeMode가 실행되도록 onClick을 설치해준다.

Control.js

```
import React, { Component } from 'react';
class Control extends Component {
  render(){
    console.log('Subject render');
    return (
      <ul>
        <li><a href="/create" onClick={function(e){
          e.preventDefault();
          this.props.onChangeMode('create');
        }}>
```

```

    }.bind(this)}>create</a></li>
    <li><a href="/update" onClick={function(e){
      e.preventDefault();
      this.props.onChangeMode('update');
    }.bind(this)}>update</a></li>
    <li><input type="button" onClick={function(e){
      e.preventDefault();
      this.props.onChangeMode('delete');
    }.bind(this)} type="button" value="delete"></input></li>
  </ul>
);
}
}
export default Control;

```

App.js의 onChangeMode가 호출될 때, 현재 상태를 전달받을 수 있는 인자를 mode라는 변수로 선언 해준다.

```

App.js

<Control onChangeMode={function(_mode){
  this.setState({
    mode:_mode
  });
}.bind(this)}></Control>

```

여기까지 하면, create update delete를 클릭했을 때, mode의 값을 create update delete로 변경된다.