

# 3

## 3회차 Basic Mission - 노진현

### 모달 창 외 클릭 시 모달을 닫게 만드는 custom hooks 생성하기

#### 1. 아이디어 1

- wrapper-modal div에 onClick 속성 부여하고 클릭된 target이 wrapper-modal class를 포함하고 있으면 modal창 닫기

```
<div
  className="wrapper-modal"
  onClick={(e) => {
    if ((e.target).classList.contains("wrapper-modal")) {
      setModalOpen(false)
    }
  }}
>
```

⇒ custom hook으로 구현할 수 없음

#### 2. 아이디어 2

- useRef() hook을 이용해서 modal class를 갖는 div를 ref로 선택해주고
- useEffect() hook을 이용해서 ref와 setModalOpen 값이 변할 때, window 객체에 클릭 이벤트를 생성하고 클릭 대상이 앞서 설정한 ref를 포함하고 있지 않으면 `setModalOpen()` 모달창을 닫는다.

아이디어 2를 custom hook으로 생성해보면 다음과 같다.

custom hook에서는 모달창 대상이 되는 `ref` 값과 해당 모달창의 Open 여부를 결정하는 `setModalOpen` 함수를 넘겨받고 화면 클릭에 대한 모달창 닫기 여부를 결정한다.

```
import { useEffect } from "react";

export const useModalClose = (ref, setModalOpen) => {
  useEffect(() => {
    const handleClickOut = ({ target }) => {
      console.log(target);
      if (!ref.current.contains(target)) setModalOpen(false);
    };

    window.addEventListener("click", handleClickOut);

    return () => {
      window.removeEventListener("click", handleClickOut);
    };
  }, [ref, setModalOpen]);
};
```

MovieModal > index.js 에서는 custom hook에 넘겨줄 ref에 대한 설정이 필요하고, 이를 핸들링하는 함수를 넘겨준다.

```
const MovieModal = ({
  backdrop_path,
  title,
  overview,
  name,
  release_date,
  first_air_date,
  vote_average,
  setModalOpen,
}) => {

  const modalOut = useRef();

  useModalClose(modalOut, setModalOpen);

  ...

  return (
    <div className="presentation">
      <div className="wrapper-modal">
        <div className="modal" ref={modalOut}> //ref 선택
      </div>
    </div>
  )
}
```

이렇게 작성해봤는데, 모달창이 중첩하여 생성될 때 효과적으로 사용할 수 있는 custom hooks인지는 확실하기 어려운 것 같다.

아이디어 2는 여기 를 참고하여 작성할 수 있었다.