

ジャンルの登録/削除							
達成要件				実装の例			
<p>■ 入力された文字を genreName という state へセットする、「handleChangeGenre」を作成してください。</p> <p>■ 「×」アイコンを押した時に、削除する API をリクエストし、ジャンルを削除してください。</p> <p>■ 「追加」を押した時に、state へセットされている値をパラメーターとして、ジャンルを作成する API をリクエストし、ジャンルを作成してください。</p>				<pre>export const GenreBody = () => { const [genreName, setGenreName] = useState<string>(""); // 入力情報を管理する state const { data, dispatch } = useContext(DataContext); const onClickSubmit = async () => { const newData: GenreType = { // 入力情報をセット }; try { const genres: GenreType[] = await genreRequest("createGenres", { data: newData, }); dispatch({ type: "success", payload: { genre: genres } }); setGenreName(""); } catch (err) { dispatch({ type: "failure", payload: { error: err.message } }); } }; const handleOnDelete = async (id: number) => { // onClickSubmit を参考に、ジャンルを削除する処理を記述 }; const handleChangeGenre = (val: any) => { // 入力情報を state へセット }; return (<div className="flex direction_column horizontal_center vertical_center"> <h2 className="input_menu">ジャンル編集</h2> { /* ジャンルをすべて表示する処理を記述する */ } <li className="genre_title flex space_between" key={ジャンル<u>の id</u>}> {ジャンル<u>の name</u>} <CancelIcon onClick={() => handleOnDelete(ジャンル<u>の id</u>)} /> <input type="text" value={genreName} onChange={handleChangeGenre} /> <input className="input_submit" type="button" value="追加" onClick={onClickSubmit} /> </div>); };</pre>			