

データを格納する仕組みの実装							
達成要件				実装の例			
<p>■ useReducerを使用して、「タスク」と「ジャンル」のデータを格納してください。</p> <p>■ データの内容は、下記のような構成にしてください。</p> <ol style="list-style-type: none"> <li>「タスク」と「ジャンル」のデータ。</li> <li>「データの取得ができたか」の情報。</li> <li>HTTPリクエストでエラーが起きた際の、エラー情報。</li> </ol> <p>■ アクションは「request / success / failure」としてください。</p> <p>request・・・リクエストをした場合 success・・・リクエストが成功した場合 failure・・・リクエストが失敗した場合</p> <p>■ パラメーターにはレスポンスの内容をセットできるようにし、successの場合はデータを更新できるようにしてください。</p>				<pre> export type DataAction = {   type: 'request'   'success'   'failure'   payload: { hoge?: any[]; error?: string } }  export const useDataReducer = (): [Data, ({ type, payload }: DataAction) =&gt; void] =&gt; {   const initialState = {     hogeData: [{ name: '' }],     isLoading: false,     error: '',   }    const reducer = (data: Data, action: DataAction) =&gt; {     switch (action.type) {       case 'request':         return {           ...data,           isLoading: true,         }       case 'success':         return {           ...data,           hogeData: action.payload.hoge    data.hogeData,           isLoading: true,         }       case 'failure':         return {           ...data,           isLoading: true,           error: action.payload.error,         }     }   }    const [data, dispatch] = useReducer(reducer, initialState)   return [data, dispatch] } </pre>			