

タスクを管理するための HTTPリクエストを管理する関数を定義

達成要件	実装の例
<p>■ <code>taskRequest</code>を定義し、返り値は HTTPリクエストのレスポンスのデータを返却する。</p> <p>■ <code>taskRequest</code>の第1引数には、リクエストの種類である <code>action</code>を設定できるようにします。また、アクションに対する処理は下記の通りに実装する。</p> <ol style="list-style-type: none">1. <code>fetchTasks</code>・・・タスクのデータを取得する。 (例: <code>http://localhost:3000/tasks</code>)2. <code>createTasks</code>・・・タスクを作成する。3. <code>updateTasks</code>・・・タスクを更新する。4. <code>deleteTasks</code>・・・タスクを削除する。5. <code>updateStatus</code>・・・タスクの進捗状態を変更する。 <p>※<code>switch</code>を使用し、<code>action</code>の値に応じて適切な処理が実行されるようにする。</p> <p>■ <code>taskRequest</code>の第2引数には、リクエストのパラメーターとして使用するデータを渡せるようにする。</p> <p>■ 基本情報は、<code>axiosBase.create</code>を使用して省略する。</p>	<pre>1 import axiosBase from 'axios' 2 import HogeType from '../interfaces/Hoge' 3 4 type action = "fetchHoge" "createHoge" "deleteHoge" 5 type parameter = { id?: number, data?: HogeType } 6 7 const api = axiosBase.create({ 8 baseURL: 'http://localhost:3000/hoge', 9 responseType: 'json' 10 }); 11 12 export const hogeRequest: any = async (action: action, parameter: parameter) => { 13 switch (action) { 14 case 'fetchHoge': 15 const hoge = await api.get('/') 16 return hoge.data 17 case 'createHoge': 18 const createHoge = await api.post('/', parameter.data) 19 return createHoge.data 20 case 'deleteHoge': 21 const updateHoge = await api.delete(`/\${parameter.id}`) 22 return updateHoge.data 23 default: 24 return null 25 } 26 }</pre>