

## ジャンルを管理するための HTTPリクエストを管理する関数

達成要件	実装の例
<p>■ genreRequest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>■ genreRequestの第1引数には、リクエストの種類である action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"><li>1. fetchGenres...ジャンルを取得する。 (例: http://localhost:3000/genres)</li><li>2. createGenres...ジャンルを作成する。</li><li>3. deleteGenres...ジャンルを削除する。</li></ol> <p>※switchを使用し、actionの値に応じて適切な処理が実行されるようにする。</p> <p>■ genreRequestの第2引数には、リクエストのパラメータ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、axiosBase.create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) =&gt; { 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>