

# ReactとFirebaseを使用したToDoList

- 概要
- 要件定義
  - 機能概要
  - 画面要件
    - 1.リスト初期表示
    - 2.ステータス表示の切り替え
    - 3.リスト項目の追加
    - 4.リスト項目の削除
    - 5.リスト項目の編集
    - 6.モーダルUI（追加, 削除, 編集共通）
- 詳細設計
  - API仕様
    - Response Example
  - シーケンス図
    - 1.リスト初期表示
    - 2.リスト項目のステータス変更
    - 3.リスト項目の追加・編集・削除

## 概要

概要	Reactの習作として制作したToDoリストページ
ページURL	<a href="https://iiishkiii-react-todolist2.vercel.app/">https://iiishkiii-react-todolist2.vercel.app/</a>
Gitリポジトリ	<a href="https://github.com/iiishokoiii/app-react-todolist_v2">https://github.com/iiishokoiii/app-react-todolist_v2</a>
使用技術	<ul style="list-style-type: none"><li>フロントエンド: React, tailwind css</li><li>API: Firebase Realtime Database</li><li>hosting: Vercel</li></ul>

## 要件定義

### 機能概要

- APIからリスト一覧を取得し表示する、リストはフィルタ表示・ソートが行える
- ページネーションが表示され、SPAで次ページ以降が表示される
- ユーザー操作により、リスト項目のステータス変更・追加・編集・削除ができる
- 更新内容は Firebase Database にも反映される
- API取得・更新のエラー時、ローディング中には、各画面が表示される

### 画面要件

#### 1.リスト初期表示

#	仕様詳細	UI
---	------	----

1	<p>API 取得後、リスト項目が一覧に表示される</p> <p>リスト項目は、作成日時の新しい順にソートされ、完了ステータスに応じたスタイルが適用される</p>	
2	<p>ページロードから API取得完了までは、loading 画面が表示される</p>	
3	<p>API post時にエラーとなった場合は、エラーメッセージを表示する</p> <p>エラーメッセージは下記の2パターンとする</p> <p>A. タイムアウトエラー（主にユーザーの通信環境に起因するもの）</p> <p>B. それ以外のエラー（サーバーエラーなど）</p>	


## 2.ステータス表示の切り替え

#	仕様詳細	UI
1	<p>ヘッダーのステータス表示押下で、</p> <ul style="list-style-type: none"> <li>ヘッダーのステータスが「未完了」「完了」「全て」の順に切り替わる</li> <li>ステータスに応じた項目が一覧に表示される</li> </ul>	
2	<p>各リスト項目のチェックボタン押下で、</p> <ul style="list-style-type: none"> <li>該当項目のステータス表示が切り替わる</li> <li>Firebaseのデータにも更新が反映され</li> </ul>	

### 3.リスト項目の追加

#	仕様詳細	UI
1	リストの追加ボタン押下で、追加モーダルが表示される	
2	モーダル内のフォーム入力し「OK」ボタン押下で、 <ul style="list-style-type: none"> <li>リスト一覧に新しい項目が追加表示される</li> <li>Firebaseのデータにも更新が反映される</li> <li>モーダルが閉じる</li> </ul>	1-1と同じ
3	フォームへ未入力状態で「OK」ボタン押下した場合は、エラーメッセージが表示される	


### 4.リスト項目の削除


#	仕様詳細	UI
3-1	リスト項目の削除ボタン押下で、削除モーダルが表示される モーダルのヘッダには、該当項目の名称が表示される	
3-2	モーダル内の「OK」ボタン押下で、 <ul style="list-style-type: none"> <li>リスト一覧から該当項目が削除される</li> <li>Firebaseのデータにも更新が反映される</li> <li>モーダルが閉じる</li> </ul>	1-1と同じ

## 5.リスト項目の編集

#	仕様詳細	UI
1	リスト項目の編集ボタン押下で、編集モーダルが表示される モーダルのヘッダと、フォームのplaceholderには、該当項目の名称が表示される	
2	モーダル内の「OK」ボタン押下で <ul style="list-style-type: none"> <li>リスト一覧から該当項目が削除される</li> <li>Firebaseのデータにも更新が反映される</li> <li>モーダルが閉じる</li> </ul>	1-1と同じ
3	フォームへ未入力状態で「OK」ボタン押下した場合は、エラーメッセージが表示される	

## 6.モーダルUI（追加, 削除, 編集共通）

#	仕様詳細	UI
1	モーダル内の「キャンセル」ボタン、「閉じる」ボタン、背景のオーバーレイを押下で、モーダルを閉じる	

2	<p>API post時にエラーとなった場合は、モーダル内にエラーメッセージと、「OK」ボタンを表示する 「OK」ボタン押下でモーダルが閉じる</p> <p>エラーメッセージは下記の2パターンとする</p> <p>A. タイムアウトエラー（主にユーザーの通信環境に起因するもの） B. それ以外のエラー（サーバーエラーなど）</p>	
---	--	--

## 詳細設計

### API仕様

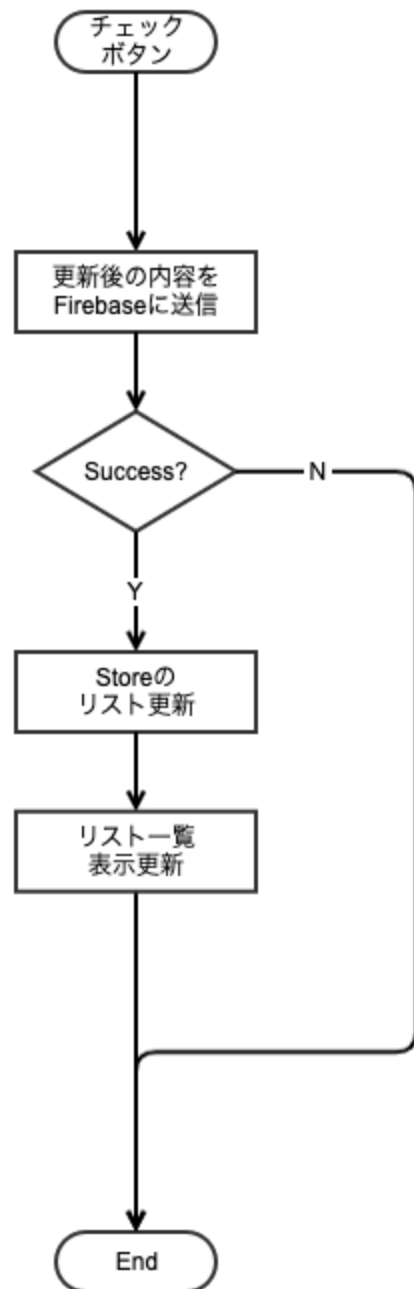
#### Response Example

```
{
  "data": [
    {
      "title": "アイテム 1 "
      "date": "Thu Feb 09 2023 22:52:56 GMT+0900 (日本標準時)",
      "checked": false,
      "id": 22,
    },
    {
      "title": "アイテム 1 "
      "date": "Thu Feb 09 2023 22:52:56 GMT+0900 (日本標準時)",
      "checked": false,
      "id": 22,
    },
  ],
  "status": 200,
  "statusText": "OK"
}
```

### シーケンス図

#### 1. リスト初期表示

#### 2. リスト項目のステータス変更



### 3. リスト項目の追加・編集・削除

