Reactを動かす

環境作成

1. 任意のterminalを起動してchatbotディレクトリに移動
2. 以下のコマンドでイメージを作成

```
docker-compose build
```

3. 以下のコマンドでコンテナを作成 chatbotサービスを動かしてreactのプロジェクトを作成する

```
docker-compose run --rm study sh -c 'npx create-react-app study-app'
```

4. 以下のコマンドでコンテナを起動

```
docker-compose up -d
```

5. 問題なさそうであれば、ブラウザ上で http://localhost:3000 にアクセスする *アクセスできるまでに5分ほどかかる場合がある

6. HotReloadを有効にするためpackage.jsonのscriptsを以下の内容にする

```
"scripts": {
   "start": "WATCHPACK_POLLING=true react-scripts start",
```

```
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  }
```

各種勉強

package-lock.json

Node.jsにおいてインストールするパッケージが記述されたファイル
環境が変わってもコマンド一発で元々あったパッケージをインストールすることが可能
pythonのrequement.txt的なもの

- パッケージをインストールした時に新規作成・更新される（npm installコマンド実行時）
- 実際にインストールしたパッケージ情報が記載されている（npm installコマンド実行時）
- node_modules の中に入っているモジュールのすべてが記載されている
- 直接編集してはいけない
- 基本的に無視していいファイル

JavaScriptのモジュール機能

- 原則は1ファイル = 1モジュール
- 必要な時に必要なモジュールのみを読み込む

React

- import文の肥大化を防ぐためにindex.jsというエントリポイントが存在する
- JSXという記述をすることでhtmlライクに書くことできる
- propsはコンポーネントに引数を渡すことができる
- stateはコンポーネント内部で宣言・制御される値
- useEffectはdomがレンダリングされる際に実行される処理を定義することができる

命名規則

- ファイル名とコンポーネントはパスカルケース(Ex HogeHuga.js)
- 関数や変数名はローワーキャメルケース(Ex getHuga)

データモデル設計の始め方

- データモデルはViewから設計する -> 手書きやfigmaでやってみる