

今日のご飯 仕様書

24G1037 小山 絢

2025 年 1 月 6 日

1 不足内容リスト

1.1 利用者向けの不足内容

1. そもそも何かを記述していない.
2. 何ができるのかを記述していない.
3. SPA の画面の写真などが載っていない.

1.2 管理者向けの不足内容

1. git hub の利用の仕方や node のインストール方法などを記述していない.

1.3 開発者向けの不足内容

1. 通信するデータの形式や例は節 2.3 に記述する.
2. 通信内容の例を図や表を用いてわかりやすくする.

2 システムの説明

2.1 利用者向け

開発したシステムを利用するための説明を行う。本課題では SPA を 1 つ開発した。まず開発したシステムで何ができるのかを説明する。本システムでは今日のご飯を選択することができる。次に本システムの利用方法を説明する。本システムの web ページにアクセスするための url は <http://localhost:8080/public/report.html> である。アクセスするにはプログラムの取得及びサーバーを立ち上げる必要があるが、その手順は後述する。図 1 は本システムの web ページ画面である。同じ色で囲われているボタンが対応していることを示している。アクセスすると家と外の文字の下に内容というボタンが置かれている。内容ボタンを押すことで今日のご飯は何にするかを選択することができる (家だとハンバーグ、外だと焼き肉など)。内容ボタンを押したあとには、左上の戻るボタンから戻ることができる。

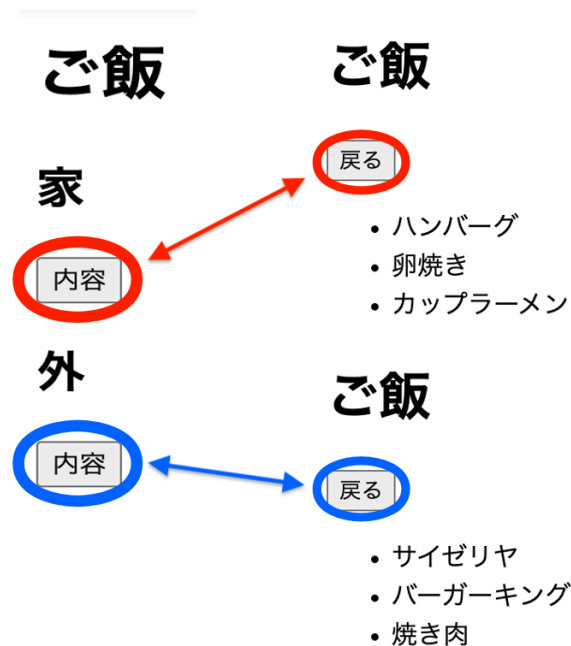


図 1 開発した SPA の画面

2.2 管理者向け

開発した SPA を管理するための説明を行う。web ページにアクセスするためには、サーバーを起動させる必要がある。本システムのソースコードは https://github.com/ken4510471/webpro_06 に載せている。ソースコードの取得のため、ターミナルを起動し `git clone https://github.com/ken4510471/webpro_06` を入力する。次にターミナルで JavaScript を利用するために node.js のインストール手順を説明する。ターミナルを起動し、以下の 4 つのコマンドを順番に一つずつ実行する。

1. `nodebrew install stable`
2. `nodebrew ls`
3. `nodebrew use v22.9.0`
4. `npm install -g npm`

node.js のインストールが終了したら後、ターミナルで本システムのソースコードがあるディレクトリに移動し、`node app9.js` と入力する。そうすることでサーバーを起動することができる。サーバーが立ち上がったら、節 2.1 で記述した手段で web ページにアクセスすることができる。また、サーバーを閉じる際には、`node app9.js` を入力したターミナルで `ctrl+C` を入力する必要がある。

2.3 開発者向け

本システムの作り方や変数の意味、通信時に流れるデータの形式や例の説明を行う。

本システムでは `public` ファイル内の `report.html`, `report.js`, そして `views` ファイル内の `app9.js` の 3 つの

ファイルを使用している。

まず public ファイル内の report.html, report.js の説明を行う。

report.html には web ページでのタイトルやボタンが配置されている。また, report.js という JavaScript ファイルを指定し読み込んでいる。

report.js にはクライアントの操作に応じたプログラムが開発されている。「家/外」ボタンが押されると fetchContent(house/outside) 関数が実行される。fetch 関数ではサーバーに POST リクエストを送信している。サーバーは JSON 形式で「家/外」を送る。レスポンス処理を行いメイン画面を非表示にし詳細画面を表示する。その後コンテンツの内容が削除され、「家/外」のデータを配列として新たに表示する。また、「戻る」ボタンが押されると詳細画面を非表示にしメイン画面を表示される。

views ファイル内の app9.js の説明を行う。app9.js にはサーバー側のプログラムが開発されており、クライアントからの POST リクエストを受信し、JSON 形式でデータを送る役割がある。/house に POST リクエストが送信された場合、message: ["ハンバーグ","卵焼き","カップラーメン"] を JSON 形式で返す。/outside に POST リクエストが送信された場合、message: ["サイゼリヤ","バーガーキング","焼き肉"] を JSON 形式で返す。

図 2 は、本システムの通信時に流れるデータの例を簡易的に示している。

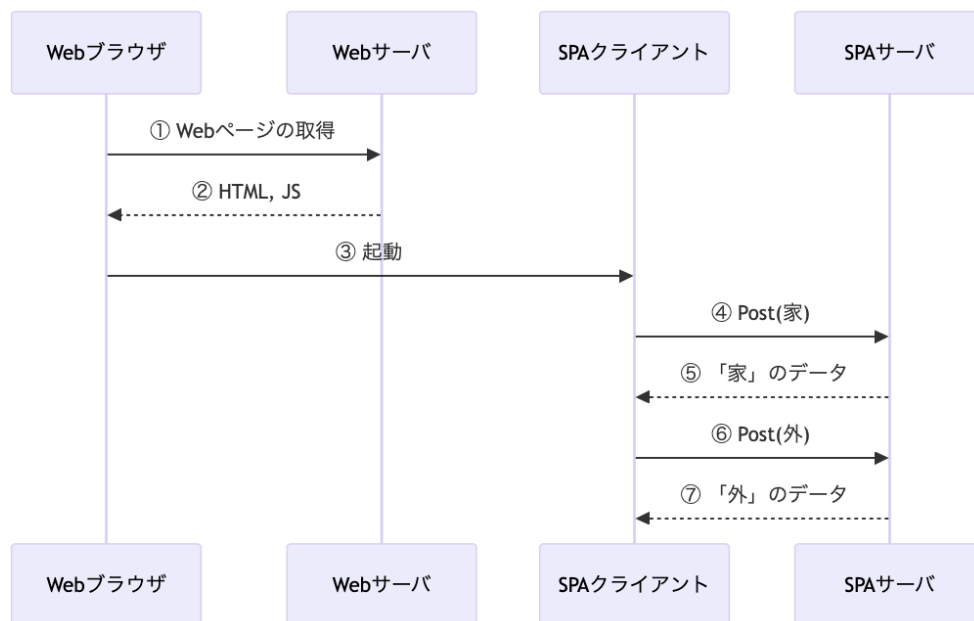


図 2 通信時に流れるデータの例