

# 今日のご飯

24G1037 小山 絢

2025 年 1 月 6 日

## 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。まず開発したシステムの 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。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。

## 2 管理者向け

開発したシステム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](https://github.com/ken4510471/webpro_06) に載せている。webpro\_06 内の `app9.js` を起動させる。起動にはターミナルを使用する。ターミナルを立ち上げたら、適切なディレクトリに移動し、`node app9.js` を入力する。そうすることでサーバーを起動することができる。

## 3 開発者向け

開発した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説明を行う。本システムには `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 ブラウザ上での表示内容を開発している。`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。web ブラウザを立ち上げた際の画面をメイン画面とし、家の内容ボタンや外の内容ボタンを押すとそれぞれに対応したコンテンツ画面に移動することができる。また、戻るボタンを押すことでコンテンツ画面からメイン画面に戻ることができる。家の内容ボタンと外の内容ボタンを押すと `fetchContent(type)` 関数が実行される。関数内の `/${type}` は押したボタンに応じて変わる。ボタンを押した際に画面を変えるために、このような関数を利用している。コンテンツ画面で表示する内容は複数あるため、配列を表示できるように if 文以降のプログラムを利用している。

`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を開発している。通信時に流れるデータとして、`app.post` では `/house` と `/outside` の POST リクエストを受け付け、配列の内容を JSON 形式で返している。また、サーバーの起動は 8080 番ポートを使用している。

## 4 通信するデータの形式や例

通信するデータの形式は POST でリクエストを行い、JSON 形式で返している。本システムを例にすると、クライアントが家の中で何を食べるかを選択するために POST メソッドでリクエストを行っている。そして JSON 形式で選択内容の候補を返している。