

BBS 掲示板の仕様書

24G1055 桑原拓也

利用者向け

掲示板を利用するには、サーバーが開いているとき、web ブラウザで `http://localhost:8080/public/bbs.html` にアクセスする。アクセスすると図 1 のような画面のサイトにアクセスされる。掲示板では名前とメッセージの投稿、投稿の確認、投稿の削除・更新、投稿の検索をすることができる。

機能の説明

- ・投稿機能は、名前の右の欄に掲示板で使用する名前を入力し、メッセージの右の欄に投稿したい内容を入力し、送信ボタンを押すことで、サーバーに内容が送られる。
- ・投稿確認では、投稿チェックボタンを押すことで投稿された名前と内容、編集、削除が上に表示される。複数投稿があった場合には新しく投稿された内容が前の投稿の下に表示されるようになっている。例として、図 2 はテストという名前でテストとメッセージを送信したものである。
- ・編集機能では、投稿された内容が表示された横に空欄のテキストボックスがあり、テキストボックスに変更したい内容を入力し編集ボタンを押すことで、その投稿の内容をテキストボックスに書き込んだ内容に変更し投稿される。
- ・削除機能では、投稿された内容が表示された横に削除ボタンがあり、削除ボタンを押すことでその投稿を消すことができる。投稿を削除した場合復元をすることはできない。
- ・検索機能では、キーワードを入力した右の欄に検索したい投稿内容の 1 部を入力し、検索ボタンを押すことで、入力したキーワードの内容が含まれる投稿のみを表示する。また、検索リセットボタンを押すと、もう一度すべての投稿を表示する。

Figure 1 shows the BBS site interface. It includes input fields for '名前:' (Name) and 'メッセージ:' (Message), a '送信' (Send) button, a '投稿チェック' (Check Post) button, and a search section with 'キーワードで検索:' (Search by keyword), a '検索' (Search) button, and a '検索をリセット' (Reset Search) button.

図 1 掲示板のサイトの画面

Figure 2 shows an example of posts displayed on the BBS site. It includes a post by 'テスト' (Test) with a message 'テスト' (Test). The post has buttons for '編集' (Edit) and '削除' (Delete). Below the post, there is a search section with '名前:' (Name) set to 'テスト1', 'メッセージ:' (Message) set to 'テスト', a '送信' (Send) button, a '投稿チェック' (Check Post) button, and a search section with 'キーワードで検索:' (Search by keyword), a '検索' (Search) button, and a '検索をリセット' (Reset Search) button.

図 2 投稿を表示した例

管理者向け

掲示板を使用するに当たってプログラムの書かれたサーバーを起動する必要がある。掲示板のサーバーを起動するのに必要なファイルは、表 1 に示す 4 つのファイルである。サーバーの起動方法は、まずターミナルを起動しプログラムの入った適切なファイルに移動する。次に、ターミナルで `node app7.js` を入力し掲示板のサーバーを起動する。ソースコードのファイルは https://github.com/KuwabaraTakuya/webpro_06 の github 内に置いてある。

表 1 掲示板のファイル一覧

ファイル名	説明
app7.js	サーバーのプログラム
public/bbs.js	クライアントのプログラム
public/bbs.html	掲示板の表示画面
public/bbs.css	HTML のスタイルシート

開発者向け

プログラムの概要

この掲示板は、テキストを書き込みボタンをクリックすることによってサーバーとクライアントの双方向でデータを送受信し、テキストを保存したり、表示させたり、編集・削除したりするプログラムである。図3は、掲示板システムの全体の流れを表したものである。web ブラウザで web ページを取得したあとは、BBS クライアントと BBS サーバが双方向でデータを送受信することでシステムができています。掲示板システムのソースコードのファイルは https://github.com/KuwabaraTakuya/webpro_06 の github 内に置いてある。

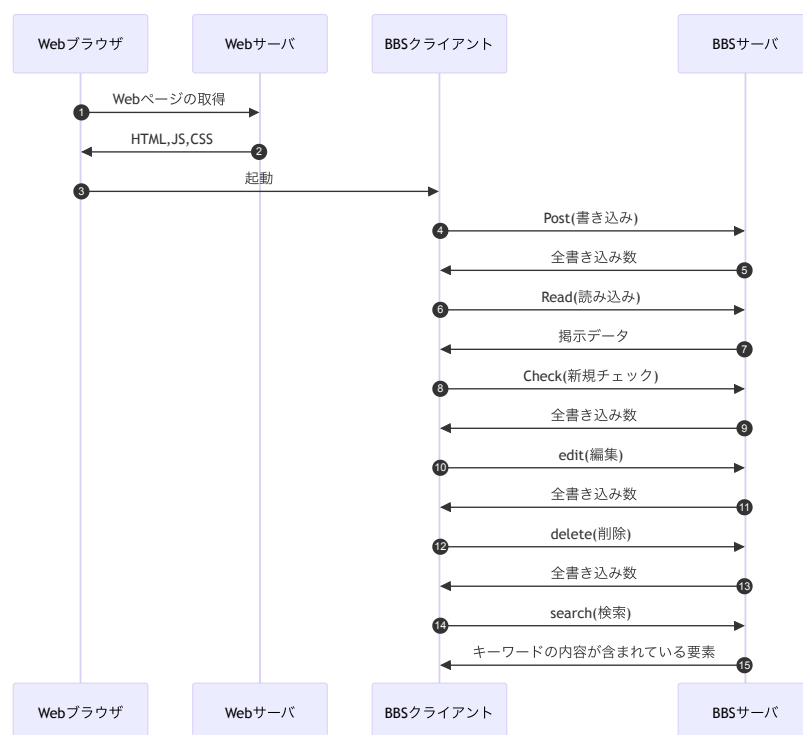


図3 掲示板システム全体の流れ

プログラムの詳細

送信ボタンを押したときの流れ

まずクライアント側でメッセージの書き込みされた内容を受け取り、POST メソッドを使用し body に受け取った内容を格納しサーバーへと送信する。サーバーでは body の内容を受け取り、bbs の配列の中に格納する。このとき、bbs の長さを id として bbs の中に格納する。そしてサーバー側から json 形式でクライアント側に bbs の長さを返す。クライアント側は bbs の長さを受け取ったら、メッセージを書き込むテキストボックスを空にする。

表 2 送信の際に使用した変数

変数名	説明
bbs	名前, 投稿内容, id の保存
name	入力された名前の記録
message	入力された投稿内容の記録
params	HTTP リクエストの詳細
id	bbs の配列の長さを記録

投稿チェックボタンを押したときの流れ

まずクライアント側で POST メソッドを使用しサーバーへと送信する。サーバーでは, json 形式で bbs の長さをクライアントへ返す。bbs の長さを受け取り, bbs の長さと number 違った場合, POST メソッドを使用し body に number を格納しサーバーへと送信する。サーバーでは, number を受け取り 0 であれば json 形式で bbs を返し, そうでなければ json 形式で bbs の配列の number 番目から最後までをクライアントへ返す。クライアント側で bbs を受け取り, bbs に格納されていた名前とメッセージを表示させる。このとき編集するためのテキストボックスとボタン, 削除するためのボタンを表示させ, onclick によってボタンを押したときそれぞれの機能を実行する関数を呼び出すように設定した。onclick は, HTML 要素に対してクリックイベントを監視し, そのイベントが発生したときに, 値として関数を代入することで動作を指定できるものである。onclick を使うことで, 1 つ 1 つのボタンに別の id を割り振らずに対象の投稿に動作させることができるため採用した。

表 3 投稿チェックの際に使用した変数

変数名	説明
bbs	名前, 投稿内容, id の保存
number	これまで読み込んだ投稿の数
params	HTTP リクエストの詳細
cover	各投稿を div.cover の中にまとめる
name_area	投稿者の名前表示のための span 要素
mes_area	投稿内容表示のための span 要素
edit_area	編集内容を書き込むためのテキストボックス
edit.button	編集するためのボタン要素
delete.button	削除するためのボタン要素

編集ボタンを押したときの流れ

編集ボタンを押したとき onclick によって editPost 関数を呼び出す。このとき, 編集のテキストボックスの内容と bbs に格納されていた id を引数とする。まずクライアント側で POST メソッドを使用し body に編集のテキストボックスの内容を格納しサーバーへと送信する。サーバーでは, 編集のテキストボックスの内容と id を受け取り, bbs の配列の中の id が同じ要素を取り出し, その要素のメッセージを編集のテキストボック

スの内容に変更し、json 形式で bbs の長さをクライアントへ返す。クライアント側は bbs の長さを受け取ったら、HTML を空にして number を 0 にしてから投稿チェックをすることで、すべての投稿を再読込して編集した内容を表示させる。bbs の配列の中の id が同じ要素を取り出すため find() を使用した。find() とは関数を満たす配列内の最初の要素を返すメソッドである。

表 4 編集の際に使用した変数

変数名	説明
bbs	名前, 投稿内容, id の保存
id	メッセージの id の記録
newMessage	編集内容の記録
params	HTTP リクエストの詳細
post	bbs 内に同じ id を持つ要素
number	これまで読み込んだ投稿の数

削除ボタンを押したときの流れ

編集ボタンを押したとき onclick によって deletePost 関数を呼び出す。このとき、bbs に格納されていた id を引数とする。まずクライアント側で POST メソッドを使用しサーバーへと送信する。サーバーでは、編集のテキストボックスの内容と id を受け取り、bbs の配列の中の id が同じ要素を取り出し、その要素を配列から取り除き、json 形式で bbs の長さをクライアントへ返す。クライアント側は bbs の長さを受け取ったら、HTML を空にして number を 0 にしてから投稿チェックをすることで、すべての投稿を再読込して編集した内容を表示させる。

表 5 削除の際に使用した変数

変数名	説明
bbs	名前, 投稿内容, id の保存
id	メッセージの id の記録
params	HTTP リクエストの詳細
post	bbs 内に同じ id を持つ要素
number	これまで読み込んだ投稿の数

検索ボタンを押したときの流れ

まずクライアント側で検索するためのキーワードの内容を受け取り、POST メソッドを使用し body に受け取った内容を格納しサーバーへと送信する。サーバーでは、bbs の配列の中からキーワードの内容が含まれている内容を取り出し、その要素を json 形式でクライアントへ返す。クライアント側はキーワードの内容が含まれている要素を受け取り、HTML を空にしてから要素に含まれる名前とメッセージを表示する。また、検索リセットボタンを押すと HTML を空にして number を 0 にしてから投稿チェックをすることで、すべての投稿を再読込して編集した内容を表示させる。bbs の配列の中からキーワードの内容が含まれている内容を取り出す際に、filter() と includes() を使用した。includes() は、配列において特定の値が含まれているかどうか

かを判定するためのメソッドであり，今回では，キーワードの内容が bbs 内のメッセージに含まれているかを判別するため採用した．`filter()` は特定の条件の中で配列内の要素をフィルタリングし，新しい配列として返すメソッドである．今回では，配列の中にキーワードを含む要素をフィルタリングして，新しい配列として返すことでキーワードを含む内容だけを表示させるため採用した．

表 6 検索の際に使用した変数

変数名	説明
bbs	名前，投稿内容，id の保存
keyword	検索キーワードの記録
params	HTTP リクエストの詳細
results	bbs 内に同じメッセージを含む要素
number	これまで読み込んだ投稿の数