

# Web プログラミングレポート 2 仕様書

24G1051 久峯 丈

## 実施内容

BBS に 3 つ以上の機能を追加する

## 利用者向け

### 手順

1. ユーザー名とメッセージを入力し，投稿ボタンを押す.
2. 返信したいときは Reply ボタンを押し返信内容を入力し，投稿ボタンを押して返信する.
3. 投稿チェックボタンで今までの投稿を一覧表示

## 管理者向け

1. ターミナルで `.../webpro_06/` に移動し `node app8.js` でサーバを起動する.
2. `Example app listening on port 8080!` と表示されたら起動成功.
3. ブラウザで `http://localhost:8080/public/bbs.html` にアクセスし，掲示板 (BBS) を起動する.

## 開発者向け

### 追加した機能

- 投稿順整理
- 投稿と同時に投稿を一覧表示
- リプライ

### 投稿チェック

#### 関数名

`checkPosts`

## 引数

なし

## 機能

入力した文字を掲示板に投稿する

## 処理

1. サーバーが返す投稿数とクライアントの保持投稿数が一致するか確認し、一致したら終了.
2. POST メソッドで URL”/read”にリクエスト送信し、新しい投稿を取得する.
3. リプライ用のボタンとテキストボックスを配置する. テキストボックスには @投稿者名がデフォルトで入力される.
4. 入力された名前, メッセージ, 投稿順とリプライを表示し, 投稿数を更新.
5. 処理が成功したら [ 'Name1', 'Message1' ] read -> 0 のようにターミナルに出力される. ・投稿するとき (document.querySelector('#post').addEventListener('click', () => { ... }) に checkPosts を呼び出すことで投稿と同時に投稿の一覧を表示する.

## リプライ投稿

### 関数名

replyToPost

### 引数

postId (リプライ対象), replyMessage (入力したメッセージ), replyElement (リプライ投稿の本体)

### 機能

ユーザの投稿に対してリプライ (返信) メッセージを送信する.

### 処理

1. postId と replyMessage をコンソールに出力する.
2. POST メソッドで URL”/reply”にリクエストを送信する.
3. レスポンスが true の場合リプライを表示し, false の場合エラーを出力.