# **SNSシステム開発 - いいね機能**

次はタイムラインの仕上げに、いいね機能の実装を進めていきます。

## **概要**

今回は、いいね機能の実装をします。ここでは、いいね機能の作成を題材に下記について学習していきます。

1. JavaScript(jQuery)基礎
2. Ajaxを用いた非同期処理

## **要件確認**

いいね機能の要件は以下の通りです。

* ユーザーは各投稿に「いいね」することができる
* いいね済みの投稿の場合は「いいね」を取り消すことができる
* 各投稿は「いいね」されている数が表示される
* いいねボタン、いいねを取り消すボタンを押した後は画面遷移せずに内容を変える

## **テンプレートの修正**

まずはテンプレートを修正します。

**timeline.php**

<?php include('navbar.php'); ?>

<span hidden id="signin-user"><?php echo $signin\_user['id']; ?></span>

<div class="container">

・・・

<span hidden class="feed-id"><?php echo $feed['id']; ?></span>

<button class="btn btn-default js-like"><span>いいね！</span></button>

どの投稿が、誰に、いいねされるのか情報が必要になるため、それぞれ必要な値をhiddenで埋め込んでおきます。追記する場所に気をつけてください。投稿IDは各投稿情報を出力する箇所ですが、サインインユーザーは共通の部分、つまりforeachの外になります。

spanタグは後ほど必要になるので、buttonタグの中に記述しておいてください。

## **JavaScriptファイルの作成**

次にJavaScriptファイルを作成します。新しくapp.jsというファイルを作成してassets/jsの配下に保存しましょう。

**app.js**

$(function() {

$(document).on('click', '.js-like', function() {

console.log('ボタンが押されました。');

});

});

app.jsが出来たら、phpから読み込みます。

**layouts/footer.php**

<script src="assets/js/jquery-3.1.1.js"></script>

<script src="assets/js/jquery-migrate-1.4.1.js"></script>

<script src="assets/js/bootstrap.js"></script>

<script src="assets/js/app.js"></script>

上記二つの処理を記述した後、ChromeのDeveloperToolを起動して、Consoleタブを開いてください。JavaScriptのconsole.logはPHPのechoのようなもので、その出力内容はConsoleタブに表示されます。

それでは、タイムライン画面から「いいね」ボタンを押してみましょう。「いいね」をクリックした時にConsoleタブに「ボタンが押されました」と表示されれば成功です。

## **いいね機能**

ボタンが正しく動作していることが確認できたら、次はボタンが押された後の機能を作成します。

**値の取得**

**app.js**

$(function() {

$(document).on('click', '.js-like', function() {

var feed\_id = $(this).siblings('.feed-id').text();

var user\_id = $('#signin-user').text();

console.log(feed\_id);

console.log(user\_id);

});

});

hiddenで隠していた値からjQueryを用いて値を取得しています。Consoleタブを開いて値が正しく取得できていることを確認しましょう。

**サーバへの送信**

次は取得できた値をサーバへ送信していきます。

まずはlike.phpを下記の通り修正しておきましょう。

**like.php**

<?php

echo json\_encode('test');

続いて、JSファイルを修正します。

**app.js**

$(function() {

$(document).on('click', '.js-like', function() {

var feed\_id = $(this).siblings('.feed-id').text();

var user\_id = $('#signin-user').text();

$.ajax({

// 送信先や送信するデータなど

url: 'like.php',

type: 'POST',

datatype: 'json',

data: {

'feed\_id': feed\_id,

'user\_id': user\_id

}

}).done(function (data) {

// 成功時の処理

console.log(data);

}).fail(function (e) {

// 失敗時の処理

console.log(e);

})

});

});

実装ができたら「いいね」ボタンを押してみてください。Consoleタブに「test」を表示されれば成功です。

これは、まずajaxメソッドからlike.phpにデータが送られています。

次にlike.phpの処理が実行されます。ここでは単にecho文だけが書かれています。

like.phpにエラーがなければ再びapp.jsに戻り、成功時のdoneメソッドが呼ばれます。

doneメソッド内に出てくるdataという変数はサーバ側からのレスポンスが入ります。つまり、like.phpのecho文の結果がこのdataに入っているということです。

**DBへの保存**

非同期でのサーバ/フロントの通信が確認が出来たので、次はDBに値を保存します。

**like.php**

<?php

require('dbconnect.php');

$feed\_id = $\_POST['feed\_id'];

$user\_id = $\_POST['user\_id'];

$sql = 'INSERT INTO `likes` (`feed\_id`, `user\_id`) VALUES (?, ?)';

$data = [$feed\_id, $user\_id];

$stmt = $dbh->prepare($sql);

$res = $stmt->execute($data);

echo json\_encode($res);

実装できたらボタンを押してみましょう。likesテーブルに値が入っていれば成功です。

**いいね数を増やす**

ボタンが押されたタイミングでいいね数が一つ増えるように修正しましょう。

**app.js**

$(function() {

$(document).on('click', '.js-like', function() {

var feed\_id = $(this).siblings('.feed-id').text();

var user\_id = $('#signin-user').text();

var like\_btn = $(this);

var like\_count = $(this).siblings('.like\_count').text();

$.ajax({

// 送信先や送信するデータなど

url: 'like.php',

type: 'POST',

datatype: 'json',

data: {

'feed\_id': feed\_id,

'user\_id': user\_id

}

}).done(function (data) {

// 成功時の処理

if (data == 'true') {

like\_count++;

like\_btn.siblings('.like\_count').text(like\_count);

}

}).fail(function (e) {

// 失敗時の処理

console.log(e);

})

});

});

## **いいね解除の処理**

ここまででひとまずいいねを押すとDBに値が入り、数が増えるようになりました。次はいいね済みだった場合に「いいねを取り消す」ボタンが表示されるように改修していきます。

**いいね済みかどうかを取得**

各投稿がいいね済みか確認するための情報をDBから取得します。各投稿の情報なので、コメント情報を取得した時と同じように投稿情報を取得するwhile文の中に処理を書いていきましょう。

**timeline.php**

while (true) {

$record = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($record == false) {

break;

}

・・・

// いいね済みかどうか

$like\_flg\_sql = 'SELECT \* FROM `likes` WHERE `user\_id` = ? AND `feed\_id` = ?';

$like\_flg\_data = [$signin\_user['id'], $record['id']];

$like\_flg\_stmt = $dbh->prepare($like\_flg\_sql);

$like\_flg\_stmt->execute($like\_flg\_data);

$is\_liked = $like\_flg\_stmt->fetch(PDO::FETCH\_ASSOC);

$record['is\_liked'] = $is\_liked ? true : false;

$feeds[] = $record;

}

・・・

<?php if ($feed['is\_liked']): ?>

<button class="btn btn-default btn-xs js-unlike">

<span>いいねを取り消す</span>

</button>

<?php else: ?>

<button class="btn btn-default btn-xs js-like">

<span>いいね!</span>

</button>

<?php endif; ?>

上記の改修が済んだらページをリロードしましょう。いいね済みの投稿のボタンが「いいねを取り消す」ボタンになります。

**いいねを取り消す**

いいねを取り消すボタンの実装をしていきます。

**app.js**

$(document).on('click', '.js-unlike', function() {

var feed\_id = $(this).siblings('.feed-id').text();

var user\_id = $('#signin-user').text();

var like\_btn = $(this);

var like\_count = $(this).siblings('.like\_count').text();

$.ajax({

url: 'like.php',

type: 'POST',

datatype: 'json',

data: {

'feed\_id': feed\_id,

'user\_id': user\_id,

'is\_unlike': true

}

}).done(function (data) {

if (data == 'true') {

like\_count--;

like\_btn.siblings('.like\_count').text(like\_count);

}

}).fail(function (e) {

console.log(e);

})

});

いいねの時と違うのはajaxで送るdataにis\_unlikeというキーバリューを用意した点と成功時にいいね数を減らすという処理です。

続いて、like.phpを修正しましょう。

**like.php**

$feed\_id = $\_POST['feed\_id'];

$user\_id = $\_POST['user\_id'];

if (isset($\_POST['is\_unlike'])) {

$sql = 'DELETE FROM `likes` WHERE `user\_id` = ? AND `feed\_id` = ?';

} else {

$sql = 'INSERT INTO `likes` (`feed\_id`, `user\_id`) VALUES (?, ?)';

}

$data = [$feed\_id, $user\_id];

$stmt = $dbh->prepare($sql);

$res = $stmt->execute($data);

echo json\_encode($res);

実行するSQL文が変わるだけなので、if-elseにはSQLの定義のみ行います。ブラウザから実行して、DBへの作成と削除が行われていることを確認しましょう。

## **ボタンの切り替え処理**

今のままだと、一度いいねをした後ブラウザを更新しないと解除が行えません。ボタンを押したタイミングで切り替わるように修正していきましょう。

**app.js**

・・・

}).done(function (data) {

if (data == 'true') {

like\_count++;

like\_btn.siblings('.like\_count').text(like\_count);

like\_btn.removeClass('js-like');

like\_btn.addClass('js-unlike');

like\_btn.children('span').text('いいねを取り消す');

}

}).fail(function (e) {

・・・

}).done(function (data) {

if (data == 'true') {

like\_count--;

like\_btn.siblings('.like\_count').text(like\_count);

like\_btn.removeClass('js-unlike');

like\_btn.addClass('js-like');

like\_btn.children('span').text('いいね！');

}

}).fail(function (e) {

・・・

## **いいね数の表示**

最後に、いいね数の表示を直していきます。

**timeline.php**

while (true) {

$record = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($record == false) {

break;

}

・・・

$record['is\_liked'] = $is\_liked ? true : false;

// 何件いいねされているか確認

$like\_sql = 'SELECT COUNT(\*) AS `like\_cnt` FROM `likes` WHERE `feed\_id` = ?';

$like\_data = [$record['id']];

$like\_stmt = $dbh->prepare($like\_sql);

$like\_stmt->execute($like\_data);

$like = $like\_stmt->fetch(PDO::FETCH\_ASSOC);

$record['like\_cnt'] = $like['like\_cnt'];

$feeds[] = $record;

}

・・・

<span class="like-count"><?php echo $feed['like\_cnt']; ?></span>

以上でいいね機能は完成です。