

Node.js と Twitter-API を用いた
ユーザーインターフェースの開発
講義資料

1. 授業に関する単語の説明

この授業では Node.js と Twitter-API を用いて、ユーザーインターフェースの開発を行います。まず初めに、この授業に出てくる単語について、簡単ではありますがまとめておきましたので分からない単語については確認しておいてください。

ユーザーインターフェースとは？

- ユーザーインターフェース (User Interface: UI) とは、機器やソフトウェア、システムなどとその利用者の間で情報をやり取りする仕組みのことです。

Node.js とは？

- Node はスケーラブルなネットワークアプリケーションを構築するために設計された非同期型のイベント駆動の JavaScript 環境です。簡単に言えば、サーバーサイドの JavaScript 実行環境です。
- Node.js は JavaScript を使用して処理しています。したがって、Node.js を操作する上で使用する言語は JavaScript です。(※ フロントエンドの JavaScript ではなく、サーバーサイドの JavaScript です。)

Express とは？

- Node.js 上で動作する Web アプリのフレームワークです。
(フレームワークを利用することで、少ないコードで意図する機能やデザインが実現できます。)

EJS とは？

- EJS はテンプレートエンジンと呼ばれるツールの 1 つで、JavaScript のような書き方を取り入れつつ HTML が書けるという特徴を持っています。
(テンプレートエンジンとは、テンプレートと呼ばれる HTML のひな形を元にプログラムで加工し、画面に出力するためのライブラリです。)

2. Express と Twitter-API の使い方

この授業では大きく分けて2種類のファイル形式のものを扱います。それは、Node.js による **Javascript** と **EJS** です。この章では、Node.js による Javascript で扱う **Express** と **Twitter-API** について説明していきます。

まず初めに、Javascript で Express を扱うためには、Javascript ファイル（拡張子：js）に以下の記述をする必要があります。
（`/* */`の部分はコメントなので、書かなくても大丈夫です。）

```
/* express モジュールをロードし、インスタンス化して app に代入。*/  
var express = require("express");  
var app = express();
```

次に、HTTP リクエストボディのデータを取得するために、Javascript ファイルに以下の記述をする必要があります。

```
var bodyParser = require('body-parser')  
app.use(bodyParser.urlencoded({ extended: false })))
```

次に、EJS ファイルを Node.js で利用するために、テンプレートエンジンを設定する必要があるので、Javascript ファイルに以下の記述をしてください。

```
var ejs = require('ejs');  
app.engine('ejs',ejs.renderFile);
```

Node.js で Twitter-API を操作するには、Javascript ファイルに以下の記述をする必要があります。

```
var twitter = require('twitter');
var client = new twitter({
    consumer_key:      'Input Your Consumer Key',
    consumer_secret:   'Input Your Consumer Secret Key',
    access_token_key:   'Input Your Access Token Key',
    access_token_secret: 'Input Your Access Token Secret Key'
});
```

ここまでの流れとして、Express や Twitter-API を利用するために必要なものを `require()` を使って設定しました。

ここからは、実際にプログラムの処理について説明していきます。

プログラムの処理の説明として、Express による **ルーティング** と Twitter-API の使い方について触れていきます。

まずは Express によるルーティングについて説明します。基本のルーティング指定方法は以下の通りとなっています。

構文

```
app. METHOD(PATH, CALLBACK);
```

上記のものは構文であって、実際は各自で行いたい処理を書いてください。
`app` よりも右に書かれている各機能の役割は下記に記しておきます。

METHOD : HTTP リクエストメソッドを小文字で指定します。例 : `get`, `post`, `put`, `delete`

PATH : URL ルーティングを指定します。

CALLBACK : URL ルーティングが一致したときに実行する処理を指定します。

例えば、ルートパス / へのルーティングを指定する場合、以下のように指定します。

```
app.get("/", function (req, res) {  
  // GET 処理  
  res.send("Hello World");  
});
```

次に Twitter-API の使い方について説明します。

この資料において、Twitter-API の consumer key や access token key などを入れることを説明したので、その作業をやっていることを前提として説明します。

```
client.METHOD(PATH, PARAMS, function (error, tweets, response) {  
  if (!error) { /*実行する処理を書く*/ }  
  else { /*エラーに対する処理を書く*/ }  
});
```

Twitter-API には、**GET 処理**や **POST 処理**などがあります。

METHOD の部分に対応する HTTP リクエストメソッドを書いてください。

PATH の部分には実際に使用する API のパスを入力してください。

PARAMS の部分はパラメーターを入力します。各 API で必ず記述しなければならないパラメーターがあるので記述漏れのないように気を付けてください。

METHOD の部分に記述するもの：get, post, put delete の中の対応するもの。

PATH の例：'search/tweet' (キーワードによるツイートの検索)

PARAMS の例： params = {q: 'Node.js'} (キーワードの指定)

3. EJS の使い方

続いては、EJS の使い方について説明していきます。

Express では、画面表示のテンプレート（拡張子：ejs）を「views」フォルダに格納する決まりがあるので、画面表示用のファイルを作成する場合、必ず「views」フォルダの中に作成してください。

EJS を使うことで以下のことができるようになります。

- 数や文字列や配列やオブジェクトの宣言と処理(JavaScript と同じ)
- HTML への値の展開
- ループと条件分岐
- 外部ファイルの読み込みとパラメーターの引き渡し

EJS を使うには、<% %>を用意し、<% %>の中にループや外部ファイルの読み込みなどの処理を書く必要があります。

具体的な使い方はサンプルプログラムの動きを確認してみる、もしくは関連サイトに詳しく載っているので、そちらを参考にしてください。

4. さいごに

この資料はあくまでも必要最低限の内容しか網羅できていないので、分からない部分については、SA に聞いてください。
残りおよそ 1 か月ですが、自分の作りたいものを作るように頑張っていきましょう！

<関連サイト>

授業用 GitHub (サンプルプログラムや資料有り)

https://github.com/s1240112/Twitter_API/

Twitter-API の一覧 (API の一覧及びパラメーターの確認可能)

<http://westplain.sakuraweb.com/translate/twitter/>

テンプレートエンジン EJS で使える便利な構文まとめ

https://qiita.com/y_hokkey/items/31f1daa6cecb5f4ea4c9