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. 授業に関係する単語の説明

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。）

1. 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）

1. 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。

1. 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れ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>