

part.6 テキスト

WebサービスAPI勉強会

「テクニック集とマッシュアップ」

1

PHPでの復習を兼ねて、新Webサービスにチャレンジしてみよう

従来の商品検索やホテルの空室検索など、APIに条件をパラメータとしてリクエストし、その結果をブラウザに表示させるという流れを何度かやってきました。

基本的な考え方は他のAPIでも応用しやすく、その違いや注意すべきポイントを押さえておけば、今後、様々なAPIを組み合わせてのサイト制作も夢ではありません。

今回はリンクシェアが6月17日にリリースした「セール&クーポンサーチ Webサービス」を使ってみたいと思います。これは、リンクシェアに参加している企業のセール、キャンペーン、クーポンといったアフィリエイトの実績に繋がりがやすい情報を検索、取得できるWebサービスです (<http://www.bento-box.jp/lsapi/salecoupon/>より)。

まず事前準備するものとしてリンクシェアおよび多くのECサイトと提携関係のあるご自身のサイトのトークンを取得しダッシュボード(アフィリエイト管理画面)から"利用申請"をしておいてください。申請が通過しないと使えません。

また、そのサイトと提携している企業のものしかでないのでご注意ください(提携申請もしておいてください)。

このWebサービスで主に二つの情報を返してくれます。

ひとつはカテゴリやプロモーションタイプマスター一覧を取得するもの。もうひとつはメインとなるセール情報やキャンペーン情報(特典付きやクーポンなど含む)を取得出来るものがあります。

ワーク

新しいWebサービスを利用するときに必要な手順をまとめています。
下記手順で準備をしていきましょう。

①【情報収集】新しくチャレンジするWebサービスの仕様・情報をブックマークするか、プリントアウトするか、PDF等でダウンロードしておく。また、サンプルソース(SDK)等があれば、ダウンロードしておく。

<http://www.bento-box.jp/lisapi/salecoupon/>

<http://linkshare.okweb3.jp/EokpControl?&tid=285859&event=FE0006>

(ページの下のほうに仕様書のPDFダウンロードリンクがあり)

②【トークン取得】そのWebサービスを利用するためのトークン取得(ユーザ登録やそれらが無い場合もある)

③【構築準備】トークンや設定を書き込むファイルapiconfig.phpにそのトークン情報を書き込んでおく(通常のリンクシェアクロスオーバーサーチと同じトークンです)

```
61 |  
62 | // linkshare クロスオーバーサーチ  
63 | // http://www.linkshare.ne.jp/service  
64 | $l$token = "d2aa4d6dbbd3952c57fc34c5  
65 |  
66 |
```

apiconfig内の\$token="...";の中にトークンを入れて保存しておいてください。

2

カテゴリやプロモーションタイプの取得<リクエスト編>

カテゴリやプロモーションタイプを取得して表示してみます。解説ページを見ていてもある程度、内容はわかりますが、自分の手を動かしてどんなデータ構造で返ってくるかを見て、さらにそれを使ってどんなアプリケーションが設計できるかを考えてみましょう。

ここでは復習も兼ねてリクエストURLを組み立ててみましょう。

```
-<couponfeed>
  -<network id="11">
    <name>Japanese Network</name>
    -<categories>
      <category id="100301">送料無料・送料割引</category>
      <category id="100303">特典・プレゼント</category>
      <category id="100306">新発売</category>
      <category id="100304">初回購入限定</category>
      <category id="100302">値引き・割引</category>
      <category id="100305">サプライズ</category>
      <category id="100307">クーポン</category>
      <category id="100308">その他</category>
    </categories>
    -<promotiontypes>
      <promotiontype id="1002">花粉症</promotiontype>
      <promotiontype id="3001">秋冬ファッション</promotiontype>
      <promotiontype id="3002">秋のセール</promotiontype>
      <promotiontype id="4008">福袋</promotiontype>
      <promotiontype id="2001">父の日</promotiontype>
      <promotiontype id="1008">母の日</promotiontype>
      <promotiontype id="1006">春夏ファッション</promotiontype>
      <promotiontype id="1004">春休み</promotiontype>
      <promotiontype id="1001">春のセール</promotiontype>
      <promotiontype id="1003">新生活</promotiontype>
```

ワーク

ベースとなるcategory-promo.phpを用意しているので、そこに追記していくようなイメージで書き込んでいってください。

①ベースURLを確認しリクエストURLを書き始める

②トークンのパラメータの書式を確認して「?」をまず付けて「token=・・・と」書く

③リンクシェアのトークンを入れている変数名を確認。””(ダブルクォーテーション)内で変数として認識されるものは展開されることを利用して、変数名をそのまま入れる。

④プロモーションカテゴリーを表示するためのパラメータを追加します。また固定でレスポンス向上のためにネットワークも指定します。「&promocat=1」「&network=11」

⑤リクエストURLを組み立ててその内容を確認するために、一旦
echo url;
と表示し、リクエストURLの確認をする。

⑥ブラウザにそのURLを入力してXML表示を確認する。
※ただし1日50回まで(後述)

3

キャッシュを利用する

リクエストURLが固まって、返ってくるXMLの表示確認できたら、表示する部分もそのまま試してみたいところです。しかし、このWebサービスは1日のリクエスト制限もあり、1日50回までとシビアです。

もしかしたら、実際に公開する以前に開発中・テスト中に1日で使い果たすことになってしまうかもしれません。

それを防ぐにはキャッシュを利用するのが望ましいです。キャッシュがあれば、リクエスト回数を減らすことができます。キャッシュ時間を適切に設定しないと古い情報が出たり、リクエスト回数が多くなったりなど不具合が出てしまいます。

とりあえず、キャッシュの有効期限を1日と設定し、たっぷりと試行錯誤しながら開発&テストを進めていきましょう。

キャッシュを利用するには使用するサーバにPEAR/Cache_Liteがインストールされている必要があります。ローカルのXAMPPでは入っていますが、公開するタイミングでそういった理由で使えない場合が有ります。またインストールされていてもパスが違う場合もあるので、その場合はレンタルサーバ会社のサイトなどで調べてみてください。



ワーク

①cacheフォルダを作成

②パーミッションを777もしくはwindowsローカルの場合はフォルダを右クリックしてプロパティの「読み取り専用」のチェックを外す(つまりcacheフォルダ内にキャッシュが書き込めるようにする)

③require_onceでキャッシュロジックを呼び出す

④キャッシュを使う(Part2の復習)

⑤キャッシュされているかどうか問わず、XMLデータを取り出して表示させてみる

print_r(\$xml);等でブラウザに表示させます。

```
promocat=1&network=11SimpleXMLElement Object ( [network] => SimpleXMLElement Object (
[attributes] => Array ( [id] => 11 ) [name] => Japanese Network [categories] => SimpleXMLElement
Object ( [category] => Array ( [0] => 送料無料・送料割引 [1] => 特典・プレゼント [2] => 新発売 [3] => 初回
購入限定 [4] => 値引き・割引 [5] => サプライズ [6] => クーポン [7] => その他 ) ) [promotiontypes] =>
SimpleXMLElement Object ( [promotiontype] => Array ( [0] => 花粉症 [1] => 秋冬ファッション [2] => 秋
のセール [3] => 福袋 [4] => 父の日 [5] => 母の日 [6] => 春夏ファッション [7] => 春休み [8] => 春のセール [9]
=> 新生活 [10] => 敬老の日 [11] => 夏休み [12] => 夏のボーナス [13] => 夏のセール [14] => 冬休み [15] =>
冬のボーナス [16] => 冬のセール [17] => 健康 [18] => レジャー [19] => ポジョレーヌーヴォー [20] => ホワイ
トデー [21] => バレンタインデー [22] => ハロウィーン [23] => トラベル [24] => ダイエット [25] => スポーツ
[26] => ゴールデンウィーク [27] => コスメ [28] => クリスマス [29] => アウトレット [30] => お歳暮 [31] =>
お正月 [32] => お中元 [33] => おせち [34] => General Promotion ) ) ) )
```

4

キャッシュの設定を変更してみる

キャッシュの設定は\$cacheOptionsという変数に格納して設定されています。これはapiconfig.php内で定義されています。これは勉強会用にシンプルにするために外部ファイルにして、共通利用できるように仕組みにしようという考えからです。

もちろん自作でキャッシュまわりのロジック作るときは、変数名を変えたり、複数の設定を使い分けたりなど自由に設定できます。

// キャッシュの設定情報をここに入れる

```
$cacheOptions = array (  
    'cacheDir' => './cache/',  
    'lifeTime' => '3600',// 60秒  
    'automaticCleaningFactor' => 200  
);
```

現在設定されている設定はこの内容になります。設定を変更する場合、サイト全体に適用するのであればapiconfig.php内のファイルを修正します。個別のページごとに設定を変えるのであれば、個別ページで、\$cacheOptionsを上書きで設定情報を書き込めばOKです。

cacheDir・・・キャッシュされたファイルの保存場所

lifeTime・・・キャッシュの有効期間。有効期間が切れていたら、新しいキャッシュを作成する

automaticCleaningFactor・・・有効期間が切れているファイルを一括削除する頻度。

ワーク

今回はapiconfig.phpを他の勉強でもそのまま使いたいの
で、apiconfig.php内は修正しないこととします。

①apiconfig.phpのキャッシュ設定部分と同じ部分をキャッシュのやり取りをする部分の直前に書き込む。つまり、変数の上書きを利用します。

```
// キャッシュの設定情報をここに入れる
$cacheOptions = array (
    'cacheDir' => './cache/',
    'lifeTime' => '86400', // 60秒×60分×24時間
    'automaticCleaningFactor' => 200
);
```

②キャッシュの有効時間をとりあえず1日にします。lifetimeの単位は「秒」です。

5

カテゴリやプロモーションタイプの取得<表示編>

カテゴリやプロモーションタイプをリクエストできたので、今度はそれをブラウザ上に表示させます。閲覧者向けのナビゲーションを作ったりとか予め「お歳暮」に特化した情報を取得したいときの絞り込み用カテゴリIDを調べる時に使えるかもしれません。

大きくわけてカテゴリとプロモーションの二通りのデータが取得出来ます。今までのように商品情報だけ、プラン情報だけ、というわけではなく、二通りの情報がひとつにまとまっているので、どのように取り出していくかを説明していきます。

```

- <categories>
  <category id="100301">送料無料・送料割引
  <category id="100303">特典・プレゼント</category>
  <category id="100306">新発売</category>
  <category id="100304">初回購入限定</category>
  <category id="100302">値引き・割引</category>
  <category id="100305">サプライズ</category>
  <category id="100307">クーポン</category>
  <category id="100308">その他</category>
</categories>
- <promotiontypes>
  <promotiontype id="1002">花粉症</promotiontype>
  <promotiontype id="3001">秋冬ファッショ
  <promotiontype id="3002">秋のセール</promotiontype>
  <promotiontype id="4008">福袋</promotiontype>

```

カテゴリとプロモーションタイプは同列に並んでいる(同時に取得出来る)。リアルタイム性は無いので、一度取得しておけば、しばらくキャッシュで保持しておくのが懸命です。

※改めて表示された中身を見るとその性質からそんなに頻繁に更新されるとは考えられません。これこそキャッシュは相当長めにとっても良いのかもしれない。

ワーク

①XMLデータを眺めて二通りのデータ階層に降りた変数を用意して代入する。

```
$categories = $xml->network->categories->category;  
$promotiontypes = $xml->network->promotiontypes->promotiontype;
```

※一個ずつ取り出して繰り返すことが出来る階層の単位まで降りる

※一旦、`print_r($categories);`などを一度入れてブラウザで表示させてみて、正しく階層を降りて情報が格納されているか、確認してみても良い。

②それぞれでforeachで回して、結果を表示する。

```
<?php foreach ($categories as $category) { ?>  
<?php echo h($category); ?><br />  
<?php } ?>
```

送料無料・送料割引

特典・プレゼント

新発売

初回購入限定

値引き・割引

サプライズ

クーポン

その他

6

idの数値も取得して表示させる

HTMLのタグと同じように返ってくるXMLデータには属性がつけられていることがあります。

```
<category id="100301">送料無料・送料割引</category>
```

とあれば、今までと同じように「送料無料・送料割引」を表示するのは簡単ですが、id=・・・の「100301」の値、文字列を取得したい場合は次のような方法を使います。

```
echo h($category);
```

でタグの中身「送料無料・送料割引」が表示されるとすれば

```
echo h($category["id"]);
```

と属性名を指定すると、その値・文字列である「100301」を表示出来ます。

ワーク

先程のカテゴリやプロモーションタイプ一覧に、名称だけでなく、そのIDも出力させておく。

「プロモーション名→プロモーションID」

「カテゴリ名(カテゴリID)」

などの書式で、表示させてみよう。

なお、表示された内容を何らかの方法で保存しておきましょう。あまり更新されるものでもないのですが、とりあえずメモ帳で保存しても良いかもしれません。

(次のステップでそのカテゴリやプロモーションタイプで絞り込みするときに必要になるかもしれません)

7

セール・クーポン情報を表示する

同じWebサービスですが、セール・クーポン情報を取り出すには、先程のcategory-promo.phpをベースに作っていくと速いでしょう。多くの部分が使い回し出来ます。

ファイルをコピーして、saleinfo.phpを作成し、セール・クーポン情報を表示するためだけの最小限の手直しだけをします。キャッシュのロジックはそのまんま使いまわせます。

まずリクエストURL組み立て。ベースのURLは変わりません。カテゴリやプロモーションマスター一覧を取得するためのパラメータ「&promocat=1」を取り除きます。あとはmidやカテゴリIDなども追加できますが、絞り込んで結果が0件だと、作業を進めるのに残念なことになりますので、まずは絞り込まずに進めましょう。

一旦リクエストURLを組み立て、それをブラウザに表示させて、XMLデータの階層構造を見て行きます。

今後はデータはセール・クーポン情報と一元化されているので、従来通りのデータの取り出し方の方法で取り出せます。逆に言えば、先程二通りの情報を取得しましたが、一通り分削除して、今回のXMLファイルに合うように書き換えるだけです。

このように一度作ったものが流用できることを覚えておくと、作れば作るほど、加速度的に組み合わせることに集中が出来、生産性が上がります。

ワーク

①category-promo.phpファイルをコピーしてsaleinfo.phpを作成します。二通り分処理しているところを一通り分に減らしておきます。

②特に絞り込むためのパラメータを付けずにリクエストURLを手直しし、echoで一度そのURLが正しく組み立てられているか確認します。

③XMLの階層構造を見ながら違うところを修正していきます。

④link階層に個別データがセットされているので、その階層まで降りて、変数\$linksに格納します。

```
$links = $xml->link;
```

⑤従来通りforeachで\$linksからデータセットを一個ずつ取り出し\$linkに代入し、さらにそこからテキストやリンクURLを取り出して表示させていきます。

■セール&クーポン

[ニューヨーク近代美術館\(MoMA\)での「ティム・バートン」展を記念して関連アイテムを集めました！](#)

[選べるお得なウェディング特典付！ MoMAstoreの引出物](#)

[期間中サイト内のアンプレラは全て送料無料！](#)

[パーフェクトスーツファクトリー](#)

[ヌーブラ正規品2個セットで2980円☆送料無料！](#)

[MoMAstore SALE](#)

[セールスSalusのブーツ・パンプス・ワンピースなどのセール会場](#)

[今なら、Web限定クーポンご利用で、AY301が 8,400円OFF！ \(2010630 17:00まで\)](#)

[今なら、Web限定クーポンご利用で、NP12が 6,300円OFF！ \(2010630 17:00まで\)](#)

[今なら、Web限定クーポンご利用で、NP11が 6,300円OFF！ \(2010630 17:00まで\)](#)

[今なら、Web限定クーポンご利用で、Na03miniが 5,250円OFF！ \(2010630 17:00まで\)](#)

[数量限定！在庫限り！サンワダイレクトのわけあり大特価品・アウトレット情報](#)

[FMV-DESKPOWER・FMV-BIBLOの旧機種を、特別価格・安心のメーカー保証付でご提供](#)

[限定アイテムが目押し。お買い得コーナー！台数限定、パソコン・プリンター・ソフトなどの売れ筋商品が盛り](#)

[い！](#)

[ネットショップ限定アウトレットスーツ10,500\(税込\)～](#)

[アウトレットスーツ10,500\(税込\)～](#)

[イ物、掘り出し物が安い！家電オンラインの掘り出し市](#)

8

クーポンコードが存在すれば クーポンコードを表示する

セール・クーポン情報の中に、クーポンコードの情報も付いてくる情報もあります。クーポンコードを使うことによって割引などの特典が受けられます。XMLデータの中では下記のような表記で入っています。

```
<couponcode>XXXX-XXX-XXX</couponcode>
```

このクーポンコードが存在するものと存在しないものがありますが、存在する情報についてはクーポンコードも表示させるようにしてみます。

そのためには条件分岐という方法を使います。条件分岐とは、ある条件が成立するかしないかで、実行する内容を変えてしまうものです。

条件分岐のif文の書式は下記の通りです。

if (条件式) {条件式に合致したら実行する内容}

※一行にまとめられるが読みやすく複数行に分けることが多いです

ここではレスポンスにクーポンコードというものが存在していたら、「クーポンコードは……です」と表示させてみましょう。

if文で変数に代入してみたときにそれが成立するかどうかで条件分岐を行います。

ワーク

「セール情報……

クーポンコードがあればクーポンコードを表示する」
というレイアウトで表示させてみます。

セール・クーポン情報を表示した箇所のすぐ下にクーポンコードがあるかどうかを調べて、あれば表示する内容を記述していきます。

①複数行に渡るPHPを書いていくので「<?php」と「?>」をまず記入し、その間にPHP文を書いていきます。

②条件分岐の基本形を入れます。

```
if ( ) {
```

```
}
```

※括弧など開始と終了がセットになるように最初にご書いておきます。

③条件式と条件が成立した場合の実行する内容を書きます。

```
if ($couponcode=h($link->couponcode)) {  
echo "クーポンコードは".$couponcode."です<br />";  
}
```

④PC系ECサイトなどでクーポンコードを使って見積もりをしてみても実際に動作するか確認してみます。

9

複数APIに同時に通信するための 並列通信の必要性

これまではシンプルにひとつずつリクエストURLを組み立てて、XMLファイルを取得してきました。PHPのソース上もほぼライブラリも使わずにPHP5のsimpleXMLを利用してそんなの通りにシンプルにAPIを扱ってきました。

しかし、これには問題があり、同じページ上で扱うAPIの数が増えると、その数だけ重くなってしまいます。このように順番に通信していくことを直列通信と言います。

これを解決する方法はいくつかあるのですが、今回はPHPだけで処理出来るcurl_multiという方法で並列通信を使ってみたいと思います。同時に通信を行い、返ってきた値を処理していきます。(一定時間内に帰ってこなかったら無視する)

注意:Windowsローカルのxamppではデフォルトでは使えないので使えるように設定を変更します。

▽XAMPPでcurlやopensslなどのPHPエクステンションを有効にする方法

<http://phpspot.org/blog/archives/2008/02/xamppcurlopenssl.html>



■参考文献

▽APIとの通信効率をよくする実装例(1) curl_multi

http://techblog.yahoo.co.jp/cat207/cat209/api1_curl_multi/

ここでのサンプルソースを基本的に踏襲します。

■7つのAPIでの直列通信と並列通信の比較サンプル

直列通信

<http://musilog.net/chokuretsu/>

並列通信(別課題だが3秒でタイムアウトしている)

<http://musilog.net/heiretsu/>

時間帯を変えて試したり、なんか検索してみると、並列通信の有効性がわかります。キャッシュが効いているので、同じキーワードで何度も試さないでください。

※タイムアウトさせるとデータが一切とれないので、あえて、タイムアウトさせずにデータが取得出来るまで待ってキャッシュを使いまわすのもひとつの手です。

10

並列通信のための手順

複数のAPIからデータを取得する際、Part2では直列通信で商品検索結果、カテゴリ表示、カテゴリ別ランキング表示を行いました。これは一個ずつリクエストURLを組み立ててXMLファイルを取得するという流れでした。

並列通信の場合はまずリクエストURLを組み立ててしまい、それらのURLをセットします。その後、号令をかけるかのようにして同時にXMLファイルを取得するというイメージです。

しかし、少し手続きは従来のfile_get_contents関数を使うより少し複雑です。URLをセットする部分や後始末をする手順などが増えます。また、同時に複数のAPIのデータを取り扱うのでリクエストURLを仕込む変数や帰ってきたXMLデータを入れる変数を別々に独立させて用意しておく必要があります。

ワーク

今回のワークではリンクシェア、楽天、ヤフーに同時にキーワードによる商品検索を行うプログラムを作ります。

まずはリクエストURIを組み立てましょう(Part2と同じやり方です)。今回は並列通信の作業に集中するため、キャッシュは利用しません。

①仮のキーワードを設定して\$keywordという変数に格納します。

```
$keyword="バナナ";
```

②キーワードをエンコードした変数を\$query4url という変数に格納します。

```
$query4url = urlencode($keyword);
```

③リクエストURLを組み立てます。

それぞれのリクエストURLを格納する変数名は区別しやすいようにわかりやすい名前を付けると良いでしょう。

```
$lurl = "http://feed.linksynergy.com/productsearch?token=$ltoken&keyword=$query4url";
```

```
$rwsurl = "http://api.rakuten.co.jp/rws/3.0/  
rest?developerId=$rwsdevid&affiliateId=$rwsa  
ffid&operation=ItemSearch&version=2009-04-  
15&keyword=$query4url";
```

…等

④リクエストURLが正しいか確認します。

11

curlの利用手続きと設定内容

並列通信を行う際、それを利用開始する手続きと利用終了する手続きが必要です。

利用開始時

```
$mh = curl_multi_init();
```

利用終了時

```
curl_multi_close($mh);
```

それらの間にリクエストURLをセットしたり、実際に各APIへ並列通信してXMLファイルを取り出したりします。

```
$ch_1url = curl_init($1url);  
curl_setopt($ch_1url, CURLOPT_RETURNTRANSFER,  
TRUE);  
curl_setopt($ch_1url, CURLOPT_TIMEOUT, 3);  
curl_multi_add_handle($mh, $ch_1url);
```

このようにリクエストURLをセットするだけでなく、タイムアウト設定が出来ます。この例では5秒間タイムアウトしたら、通信をやめてしまうことが出来ます。

ワーク

①リクエストURLが組み立てられた後に、利用開始時に必要な記述をします。

```
$mh = curl_multi_init();
```

②3API分の下記組み合わせを記述します。変数はそれぞれのAPIごとに書き換える必要があります。また、新たに\$ch_...という変数を作成します。

```
$ch_1surl = curl_init($1surl);  
curl_setopt($ch_1surl, CURLOPT_RETURNTRANSFER,  
TRUE);  
curl_setopt($ch_1surl, CURLOPT_TIMEOUT, 3);  
curl_multi_add_handle($mh, $ch_1surl);
```

③通信を実行します。

```
do { curl_multi_exec($mh, $running); } while ( $running );
```

④3API分のXMLファイルを変数に格納します。

```
$xml_1surl = curl_multi_getcontent($ch_1surl);
```

⑤後始末します。

```
curl_multi_remove_handle($mh, $ch_1surl);  
curl_close($ch_1surl);
```

⑥利用終了手続きをします。

```
curl_multi_close($mh);
```

12

3時間目ワーク

1時間目と2時間目では、実際に画面に表示する部分が貧弱です。そこで、少しデザインを加えて実際に使える形にもっていきましょう。

<案>

・セール情報のブログパーツ化

→特定のECサイトに絞ったり、特定のイベントに絞ったりしたものを用意する等。

・クーポンコードをコピペしやすいようにinputフォームを使って表示

・とりあえずセール情報を枠で囲んでバナーっぽくする

・並列通信を使って、ひとつの検索キーワードでWeb検索、ブログ検索、商品検索の結果を表示させるような仕掛けを作る

・Part2でカテゴリー一覧とカテゴリランキングを取得していたのを、並列通信化して、表示速度を上げてみる。

