

Amazon Web サービスを使って表紙画像を表示する

杉山 智章*

Amazon が提供する API である Amazon Web サービスは、アソシエイト・プログラムと組み合わせて利用できる。図書館などの Web サイトは、これらを自館のリソースとマッシュアップすることにより、新たなサービスを創出する可能性を持っている。本稿では、その基本技術となる図書表紙画像の表示の仕方について述べる。Web サーバに実装するための PHP スクリプトをオープンソースで作成し、その解説を行う。また、Amazon の商品コードの ASIN と ISBN との関係について、ISBN-13 と ISBN-10 の併存の状況を取り上げる。そのことによる、プログラム作成上での変換方法や問題点についても PHP 関数を作成し解説する。

キーワード：Amazon Web サービス, API, アフィリエイト, 図書館, 表紙, オープンソース, PHP, ISBN

1. Amazon Web サービス

Amazon が提供する API (Application Programming Interface) を Amazon Web サービス¹⁾ (以下 AWS) という。無料で提供されているため、ID さえ取得すれば誰でも使うことができる。API の具体例を紹介した書籍²⁾などでも、AWS はトップに載ることの多い人気の高い API である。SOAP と REST という 2 つのプロトコルに対応しているが、ほとんどの場合は手軽な REST で使用されることが多い。今回作成したスクリプトも REST を使っている。AWS の基本は Web サイトに Amazon の商品情報 (特に商品画像) を表示することである。同等の機能は図書館の Web サイトで OPAC の検索結果画面³⁾や新着図書、選定図書リストなどのページで活用が見込まれる。また、AWS はジャンル一覧の取得や類似商品情報、レビューの取得などもできるので、このような機能と図書館の受入・所蔵情報などを組み合わせれば、図書館の Web サイトもよりアクティブなサイトになる可能性があるのではないかと。

さらに、AWS には Amazon のアカウントを持っているユーザならば、自分のウィッシュリストの取得やショッピングカートの操作ができる機能⁴⁾もあるため、パーソナライズに対応したサイトも作ることができる。例えば、「Amazon で図書を検索して、欲しい図書にチェックを入れておき、後で図書館サイトにログインして、自分の作成したリストの中から図書館に選定や購入を依頼する」という仕組みも実現できるかもしれない。

AWS はたいへん多機能なため、その全機能をここで扱うことはできない。基本であると思われる商品画像の表示を自サイトで行うサンプルスクリプトを作成してみたい。AWS の ID は amazon.com で登録する。日本版サイトに対応したスクリプトを作成する場合でも、ID の登録先は

amazon.com である。

登録 URL :

<https://aws-portal.amazon.com/gp/aws/developer/registration/index.html>

2. アソシエイト・プログラム

AWS と関係するサービスに Amazon アソシエイト・プログラム⁵⁾がある。自分の Web サイトから Amazon へのリンクを作成すると、そのリンクをたどって第三者が Amazon で買い物をすれば、「適格収益」の 3~7% を紹介料として手にすることができるというものである。一般にアフィリエイトと呼ばれる機能である。紹介料は Amazon ギフト券または銀行振込どちらかを選択できるようになっている。アソシエイト・プログラムのアカウントも無料で取得できる。なお、アソシエイト・プログラムを使わなくても AWS の機能の使用はできる。日本版サイトで登録したアソシエイト・プログラムのアカウントは、日本版サイトでのみ有効である。

登録 URL :

<https://affiliate.amazon.co.jp/gp/flex/associates/apply-login.html>

3. 利用条件

AWS を使うときは、必ず利用許諾⁶⁾を確認しておかなければならない。利用許諾は英語で書かれているが、特にスクリプト作成の上で注意しておかなければならない点を示すと次のようになる。

- 取得データの改変はできない。
- Amazon サイトの当該商品へのリンクを作成する。
- 取得したデータに Amazon 以外への商用ページのリンクを作成できない。
- 取得データはサーバに 24 時間キャッシュできるが、画像そのもののキャッシュはできない。一部項目については 1 ヶ月間のキャッシュができる。
- 入手条件や価格情報のキャッシュについては、1 時間以

*すぎやま ともあき 静岡大学附属図書館

〒422-8529 静岡市駿河区大谷 836

Tel. 054-238-4476

(原稿受領 2008.2.19)

上経ったときは、どの時点のデータかわかるようにする。
 ・1秒間に1コール、または40kバイトを超えたコールをしてはいけない。

4. AWS にアクセスしてみる

AWS へは Internet Explorer などの Web ブラウザからも

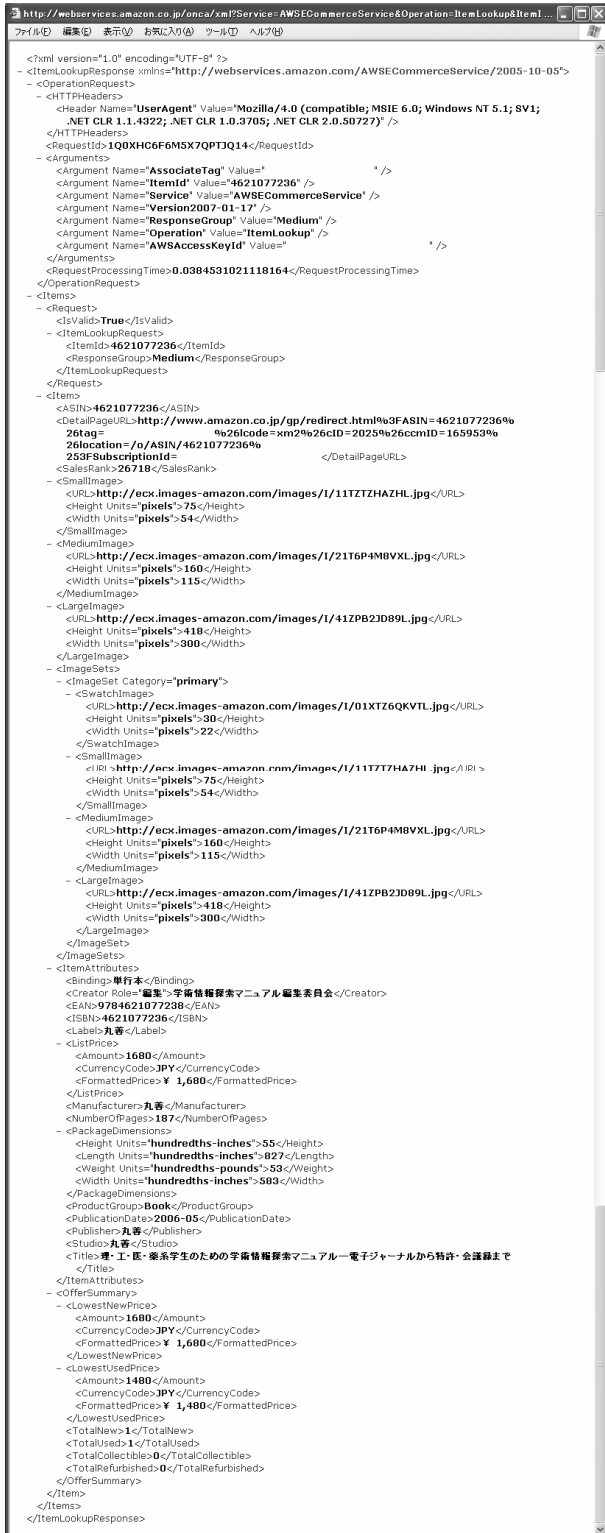


図 1 AWS レスポンスデータ

直接アクセスできる。通常の Web サイトにアクセスする場合と同じ感覚で次のようなアドレスを入力してみる。

http://webservices.amazon.co.jp/onca/xml?Service=AWSECommerceService&AWSAccessKeyId=AWS 登録 ID&AssociateTag=アソシエイト用アカウント&Version2007-01-17&Operation=ItemLookup&ItemId=4621077236&ResponseGroup=Medium

すると、図 1 のようなページが表示される。コンピュータ間でデータの受け渡しをしやすいように xml 形式で出力されるようになっていいることがわかる。この xml をそれぞれの API の規則にしたがって解析すれば、必要な個々のデータを取り出すことができるのである。通常、図 1 のデータは AWS サーバから Web サーバへのレスポンスデータとなる。

それでは図 2 のような Web ページに表紙画像を表示する AWS.php というスクリプトファイルを作成してみる。スクリプトで出力したい html は以下のようにになる。URL は次のように ISBN と画像サイズ (S, M, L) を指定できるようにする。

http://○○○/AWS.php?ISBN=4621077236&Size=M

したがって、ISBN のある図書であることが条件であるのでご了承されたい。

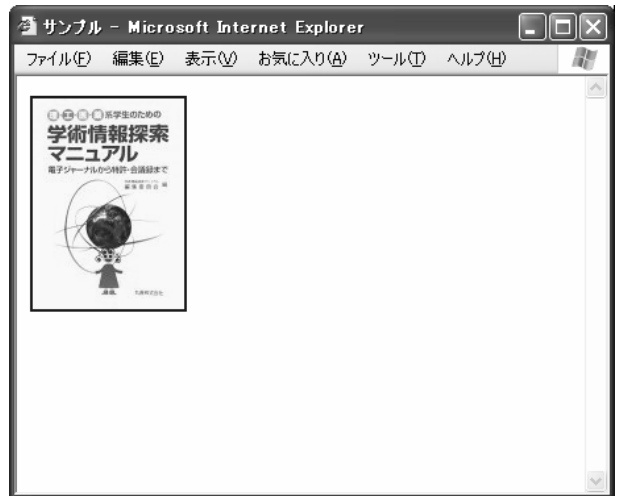


図 2 html 表示例

出力される html

```

1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4 <title>サンプル</title>
5 </head>
6 <body>
7 <a href="Amazonの商品詳細ページへのリンク"></a>
8 </body>
9 </html>
    
```

5. 動作環境

AWS の REST を使うには、Web サーバの機能と xml に対応したプログラムがあればよいが、すべてオープンソースで揃えてみる。OS には Linux の Fedora、Web サーバには Apache、スクリプト言語には PHP というように、最も普及している環境の 1 つと考えられるものを選んだ。動作確認を行ったのは次のバージョンである。PHP については、バージョン 4 と最新のバージョン 5 では仕様が異なる点が多いため、両方で動作確認を行った。

- Fedora Core 6
- Apache 2.2.6
- PHP 4.3.11, 5.1.6
- (PEAR) Services_Amazon 0.7.1

PEAR は PHP で書かれた利用頻度の高いプログラム部品にあたるもので、任意の PEAR をサーバにインストールし、用いることでスクリプト作成の作業量を軽減することができる。Services_Amazon は、AWS を使ったスクリプト作成のための PEAR である。これと依存している PEAR をインストールしておく。また、キャッシュを使うための Cache という PEAR も使用するので合わせてインストールする。

Fedora での関係 PEAR インストールコマンド

```
①yum (パッケージ管理) で PEAR 本体をインストール
# yum install php-pear
②XML_Util, XML_Parser をインストール
# pear install XML_Util
# pear install XML_Parser
③XML_Serializer をインストール
# pear install XML_Serializer-0.18.0
④Services_Amazon をインストール
# pear install Services_Amazon-0.7.1
(キャッシュ機能を使う場合) Cache をインストール
# pear install Cache
```

6. スクリプトの作成

それでは、PHP スクリプトを書いていく。編集に用いるソフトは、テキストファイルを編集できるものであれば、何でもよい。左の数字は行数である。

6.1 ファイルと設定

まず、PHP スクリプトファイルの最初に「header」により出力形式を明記する。この場合は、文字コード UTF-8 で書かれた html ファイルとして出力するということである。AWS は UTF-8 で書かれているため、作成するスクリプトも UTF-8 であればエラーを起こすリスクが低い。SJIS や EUC-JP など他の文字コードで作成する場合は、適切な文字コード変換が必要になるので注意したい。

次に AWS に表紙画像がない場合の代替表示用に「\$no_image」として画像ファイルを指定する。この画像ファイルはあらかじめ作成しておく。

「\$_GET['ISBN']」と「\$_GET['Size']」に以下のような

```
AWS.php
1 <?php
2 // ヘッダ出力
3 header ('Content-Type: text/html; charset=UTF-8');
4
5
6 // 表紙なし画像
7 $no_image = "/no_image.jpg";
8
9 ??
10 <html>
11 <head>
12 <meta http-equiv="Content-Type"
13     content="text/html; charset=UTF-8">
14 <title>サンプル</title>
15 <body>
16 <?php
17
18 // GET リクエスト取得
19 if(!$_ISBN = $_GET['ISBN']) { die('ISBN が指定されて
20     いません。');}
21 if(!$_Size = $_GET['Size']) { $_Size = 'M';}
21
```

URL 中の「4621077236」と「M」がそれぞれ取得できるようになる。

`http://○○○/AWS.php?ISBN=4621077236&Size=M`

また URL で「Size」が指定されない場合の、デフォルトサイズを「M」としている。

```
AWS.php
22
23 // ISBN の英字を大文字に変換
24 $_ISBN = mb_strtoupper($_ISBN);
25 // ISBN の半角数字と「X」以外を削除
26 $_ISBN = mb_ereg_replace('[^0-9X]', '', $_ISBN);
27
28 // ISBN フォーマット確認
29 if( mb_strlen($ ISBN) == 10 ){
30
31     if( ISBN10ckdgt( substr($_ISBN, 0, 9) ) ==
32         substr($_ISBN, 9, 1) ){
33         $ItemId = $_ISBN;
34     }else{
35         die('ISBN10 のチェックデジットが不正です。');
36     }
37 } elseif( mb_strlen($_ISBN) == 13 ){
38
39     if( ISBN13ckdgt( substr($_ISBN, 0, 12) ) ==
40         substr($_ISBN, 12, 1) ){
41         $ItemId = ISBNtoold($_ISBN);
42     }else{
43         die('ISBN13 のチェックデジットが不正です。');
44     }
45 } else {
46     die('ISBN のフォーマットが不正です。');
47 }
48
```

6.2 ISBN のチェック

ここでは、URL から取得した ISBN のチェックを行う。

取得した「\$ISBN」について、英字を大文字に変換した上で、0 から 9 の数字と「X」以外は削除する。これは ISBN に使用されない文字を除去するためである。なお、本スクリプトでは ISBN は半角で入力されていることを前提としている。URL に全角文字を使用したい場合は、URL エンコード/デコードの解決をしなければならないが、今回は省略した。

そして ISBN の検算⁹⁾である。ISBN のケタ数が 10 の場合は ISBN-10 方式でのチェックデジットが正しいことを確認する。ケタ数が 13 の場合は ISBN-13 方式でチェックデジットの確認を行い、正しければ ISBN-10 に変換する。10、13 ケタ以外の場合や、チェックデジットが不正な場合はエラー表示する。なお、ISBN-13 を ISBN-10 に変換する理由と、取り扱う際の問題については 6.6 で述べる。

6.3 Services_Amazon の読み込み

```
AWS.php
49
50 // AmazonECS4 を読み込む
51 require_once 'Services/AmazonECS4.php';
52
53 // ID をセット
54 $amazon = new Services_AmazonECS4('AWS 登録 ID',
  アソシエイト用アカウント);
55
56 // ロケールを指定
57 $amazon->setLocale('JP');
58
59 // バージョン指定
60 $amazon->setVersion('2007-01-17');
61
62 // Cache の使用
63 $amazon->setCache('file', array('cache_dir' => '/tmp/'));
64
65 // キャッシュ有効期限
66 $amazon->setCacheExpire(86400);
67
```

ここからは、AWS にアクセスするために「Services_Amazon」を使用する。「require_once」により「Services_Amazon」を読み込んだら、続けて関係設定を行う。前述した AWS 登録 ID とアソシエイト用アカウントをセットする。アソシエイト用アカウントをセットしない場合は、「アソシエイト用アカウント」に代えて「null」をセットすればよい。ロケールについてはどの国の Amazon サイトにアクセスするかを設定する。US, UK, DE, JP, FR, CA から選択できるので、日本版サイト (amazon.co.jp) を表わす JP をセットする。アソシエイト用アカウントを使用する際は、アカウントを取得したロケールと一致する必要がある。つまり日本版サイトのアカウントならば、JP を選択しなければならない。

AWS は頻繁にバージョンアップを繰り返しており、バージョンは日付で表わされている。アクセスする際は、どのバージョンの規則でリクエストするかを指定することができる。「Services_Amazon」のデフォルトバージョンは

「2005-10-05」がセットされており、このままでも基本的な機能には問題がない。しかし、2007 年以降の ISBN-13 に対応したバージョンの方が付加機能を使用できるので「2007-01-17」をセットする。

キャッシュ機能は、PEAR の Cache がインストールされていれば有効となる。「file」形式で「/tmp/」ディレクトリに 86,400 秒 (=24 時間) 有効とする。キャッシュの有効期限は AWS の利用許諾に沿ったものである。キャッシュを行うことで同一リクエストの重複を減らし、AWS サーバの負荷軽減にもなる。

6.4 Amazon データの取得

```
AWS.php
68
69 // 表示条件
70 $Options['ResponseGroup'] = 'Medium';
71 // 詳細表示実行
72 $result = $amazon->ItemLookup($ItemId, $Options);
73 if (PEAR::isError($result)) {
74     echo htmlspecialchars($result->message,
  ENT_QUOTES);
75     exit();
76 }
77
78 $item          = "";
79 $asin          = "";
80 $detailpageurl = "";
81 $s_image       = "";
82 $m_image       = "";
83 $l_image       = "";
84 $image         = "";
85
86 $item = $result['Item'][0];
87
88 $asin      = $item['ASIN'];
89 $detailpageurl = $item['DetailPageURL'];
90 $s_image   = $item['SmallImage']['URL'];
91 $m_image   = $item['MediumImage']['URL'];
92 $l_image   = $item['LargeImage']['URL'];
93
```

図 1 で示したような AWS からのレスポンスを取得するため、「ItemLookup」リクエストを行う。リクエストの種類を「Operation」と呼ぶ。「Operation」には、検索を行う「ItemSearch」やジャンル一覧を表示する「BrowseNodeLookup」などに加えて、お気に入りやカート情報取得のためのリクエストなどがある。それらのリクエストを組み合わせれば、Amazon の機能を活用した Web サイトの作成ができるようになる。今回使用する「ItemLookup」の「Options」は簡潔で、「ResponseGroup」にレスポンスデータの詳細レベル「Medium」をセットした。このレスポンスは「\$result」に配列として格納される。さらに、図 1 の<Item></Item>の中の内容を「\$item」に格納する。このうち「ASIN」、リンクのための URL と 3 種の大きさの画像 URL を取得する。この他にも、例えばタイトル情報を得たいのであれば、

```
$item['ItemAttributes']['Title']
```

などと指定すればよい。このようにしていけば、AWS の図書の商品情報から書誌データが取得できるともいえる。

6.5 画像サイズの選択と出力

```
AWS.php
94
95 //画像サイズ選択
96 switch($Size) {
97     case 'L':
98         $image = $l_image;
99         break;
100    case 'M':
101        $image = $m_image;
102        break;
103    case 'S':
104        $image = $s_image;
105        break;
106    default:
107        $image = "";
108 }
109
110 //画像ファイルがない場合
111 if(!$image){$image = $no_image;}
112
```

ここでは、実際に表示する画像サイズを選択する。画像サイズは図 3 のように 3 つから選択することができる。

Amazon には一致する ISBN の図書があっても、表紙画像が登録されていない場合がある。そこで、そのようなと



図 3 サイズ別表紙画像

きは代替として、はじめに指定した「\$no_image」画像を表示することにする。以上で表示に必要な情報がすべて揃った。

```
AWS.php
113
114 // html 出力
115 echo "<a href=¥\"$detailpageurl¥\">
    <img src=¥\"$image¥\" alt=¥\"$asin¥\"></a>¥n";
116
```

出力する html の中に取得データを埋め込む。なお、「img」タグの「alt」には画像の代替文字として「ASIN」を記述した。

6.6 ASIN と ISBN の問題について

```
AWS.php
117
118 //ここから関数の定義
119
120 //// ISBN-13→ISBN-10 変換関数
121 function ISBNtoold($ISBN){
122
123     if ( preg_match('/^(¥d{13})$/', $ISBN) ) {
124
125         $int = substr($ISBN, 3, 9);
126         return ( $int . ISBN10ckdgt($int) );
127     } else {
128         return false;
129     }
130 }
131
132 }
133
```

図 1 を確認すると、図書の場合は商品コードである<ASIN>は ISBN-10 に等しく、<ISBN>にも ISBN-10 が入力されている。一方 ISBN-13 は<EAN>に入力されていることがわかる。AWS で「ItemLookup」により図書の詳細情報を取得しようとするとき、あらかじめ ISBN-10 がわかっていたら、そのまま「ItemId」にセットすればよい。ただし、リクエストする ISBN が ISBN-13、または ISBN-10 と ISBN-13 の混合のときは気をつけなければならない。

ISBN-13 で「ItemLookup」を行うことも、2007 年から可能にはなっている。検索時の「Options」として「IdType=ISBN」と「SearchIndex=Books (和書) または ForeignBooks (洋書)」を指定することで結果が得られる。ただし、このとき検索する ISBN は ISBN-13 でなければならない。さらに、「SearchIndex」に「Books」または「ForeignBooks」を指定しなければいけないことは、和書と洋書の横断検索ができないことを意味している。そこで、本スクリプトでは ISBN-13 の場合には、4~12 ケタ目をもとに ISBN-10 のチェックデジットを算出し、ISBN-10 に変換して AWS にリクエストしている。

ISBN-13 の先頭ケタについて、「978」の他に付番された

図書が現れるようになると「ASIN」の重複の恐れもあり、本スクリプトも改変の必要が生じる。ISBN と Amazon の「ASIN」の関係が今後どうなるかは、本スクリプト最大の留意点でもある。

6.7 チェックデジット関数の作成

```
AWS.php
134
135 //// 9ケタの数字から ISBN-10 のチェックデジットを算
    出する関数
136 function ISBN10ckdgt($int) {
137
138     if ( preg_match('/^(?d{9})$/', $int) ) {
139
140         $a = chunk_split($int, 1, ',');
141         $a = explode(',', $a);
142
143         $ckdgt = 0;
144         for ($i=10; $i> 1; $i--){
145             $ckdgt += $i * (array_shift($a));
146         }
147         $ckdgt = 11 - ($ckdgt%11);
148         if($ckdgt == '10'){ $ckdgt = 'X';}
149         if($ckdgt == '11'){ $ckdgt = '0';}
150
151         return "$ckdgt";
152     }else{
153         return false;
154     }
155 }
156
157 }
158
```

このように ISBN のチェックデジット算出式を定義しておいて、他のスクリプトでも使用できるようにしておくこと便利である。

最後に、html の終了タグを記述する。

7. おわりに

今回は画像出力のためだけのスクリプトになったが、AWS のような機能を図書館サイトに実装できれば、デザイン性の向上に留まらない、新たな図書館サービスの可能性も開かれるのではないだろうか。

```
AWS.php
159
160 //// 12ケタの数字から ISBN-13 のチェックデジットを
    算出する関数
161 function ISBN13ckdgt($int) {
162
163     if ( preg_match('/^(?d{12})$/', $int) ) {
164
165         $a = chunk_split($int, 1, ',');
166         $a = explode(',', $a);
167
168         $ckdgt = 0;
169         for ($i=13; $i> 1; $i--){
170             $ckdgt += (array_shift($a)) * ($i % 2 == 0? 3 : 1);
171         }
172         $ckdgt = 10 - ($ckdgt % 10);
173         if($ckdgt == '10'){ $ckdgt = '0';}
174
175         return "$ckdgt";
176     }else{
177         return false;
178     }
179 }
180
181 }
182
183 ?>
184 </body>
185 </html>
186
```

参 考 文 献

- amazon.co.jp Web サービス
<http://www.amazon.co.jp/gp/feature.html?docId=451209>
[accessed 2008-02-17].
- 加藤貴之 [ほか]. Web API 実践リファレンスブック. 毎日コミュニケーションズ. 2007, 415p.
- 今後の図書館システムの方向性について. 2007.
http://wwwsoc.nii.ac.jp/janul/j/projects/si/systemwg_report.pdf [accessed 2008-02-17].
- 水野貴明. 俺流 amazon の作り方. アスキー. 2006, 286p.
- amazon.co.jp アソシエイトプログラム
<http://affiliate.amazon.co.jp/gp/associates/join/main.html>
[accessed 2008-02-17].
- Amazon.com: AWS License
<http://www.amazon.com/b?node=3440661>
[accessed 2008-02-17].
- 秋元裕樹. PHP×Web サービス API コネクションズ. ソフトバンククリエイティブ. 2006, 377p.
- 日本図書コード管理センター
<http://www.isbn-center.jp/> [accessed 2007-02-17].

Special feature: Encouragement of software use for information professionals. A display of cover image using Amazon Web Services. Tomoaki SUGIYAMA (Shizuoka University Library, 836 Ohya, Suruga-ku, Shizuoka-shi, Shizuoka 422-8529 JAPAN).

Abstract: The Amazon Web Services is API that Amazon offers. We can use it in combination with Associates program. By mashing up both these and each library's resource, library's Web Sites have the possibility of creating new services. This article describes, a basic technique, how to show cover images. I program PHP script with open source and explain it. And now there is a situation of the concomitance of ISBN-13 and ISBN-10 about ASIN and ISBN. It explains how to convert and the problem for programming.

Keywords: Amazon Web Services / API / affiliate / library / cover / open source / PHP / ISBN