



Web2.0勉強会
第二次ミーティング

REST API + XSLT

エンタープライズ・マッシュアップの実際

Amazon, hon.jp 等 公開されているAPIの活用事例を中心に

2006-04-11

XMLコンソーシアム・エバンジェリスト

日本ユニシス(株)

小林 茂

© XML Consortium

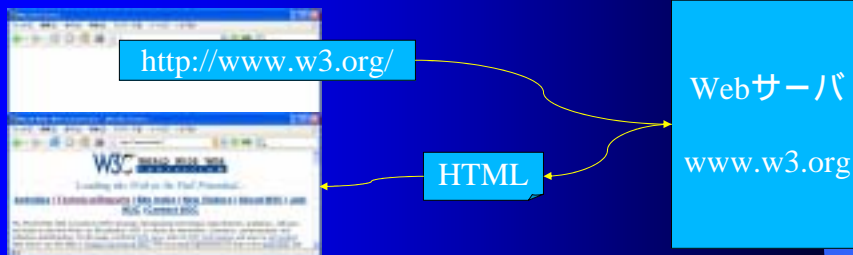


本日の内容

- **REST(GET)APIを利用する**
 - hon.jp
 - Amazon.co.jp
 - Google Map
 - Exif to RDF(kanzaki.com)
- **REST + XSLT**
 - hon.jpでのサンプル
- **マッシュアップのサンプル**
 - hon.jp + Amazon.co.jp
 - Exif to RDF(kanzaki.com) + Google Map + hon.jp

REST

- REST: REpresentational State Transfer
- HTTPプロトコル: GET, POST, PUT, ...
- ここでは, HTTP(GET)を使う
- Webブラウザのアドレス欄にURLを入れて送信するとHTMLで返ってくる



RESTのパターン

- URLを送ると, 何かが返ってくる。
- HTML, JPEG, ZIP, XML, ...
- Webブラウザは,

- HTML: ページの表示
- JPEG: イメージの表示
- ZIP: 保存する?
- XML: ソース表示
 - インデント, 折り畳み表示



応答がXMLならば

- あるプログラムがHTTP (GET)で要求し、XMLで応答されれば、そのXMLデータを何に利用できるのでは！
- その手段は、
 - javascript (ECMAScript)
 - XMLHttpRequest オブジェクト, Ajax
 - XSLT (XMLを異なる形式に変換する)
 - document()関数

詳細は後で

AmazonのREST API

- まずはこの要求を Amazon呼び出し
- この要求 URL を解剖すると
 - [http://webservices.amazon.co.jp/onca/xml?](http://webservices.amazon.co.jp/onca/xml?Service=AWSECommerceService)
 - Service=AWSECommerceService** (サービス名)
 - &AWSAccessKeyId=**アクセスキーを指定
 - &Version=**バージョンの指定
 - &ResponseGroup=**応答の内容を限定
 - &Operation=**操作の種類
 - &・・・=**操作ごとに異なるパラメタ指定

パラメタ = その値

パラメタの内容(Amazon)

- この要求 URL の内容は

&Version=2006-03-08

バージョン名がXML名前空間に設定される

&ResponseGroup=Small

Medium, Large, もっと限定したもの有り

&Operation=ItemLookup

特定商品の情報獲得
他に, ItemSearch, など有り

&IdType=ASIN

商品の型。この場合ASIN。アマゾン固有の型。

&ItemId=4798010545

商品のID。この場合ISBN。書籍の場合には, ASIN = ISBN。

パラメタとその値

応答XMLの内容

ItemLookupResponse

OperationRequest

HTTPHeaders

RequestId

Arguments

RequestProcessingTime

Items

Request

Item

Item の繰り返し

応答XML

呼び出し時のパラメタの内容

結果状況, 要求パラメタの内容

個々の情報

hon.jp のREST API

- hon.jp : 電子書籍ポータル
 - 株式会社hon.jp(インプレスグループ)
- REST APIの例

hon.jp

API呼び出し

`http://hon.jp/srch/k/ebk_rest.php?`
`uid=ユーザ識別子`
`&keyword=キーワードの文字列`
`&max=表示最大件数`
`&page=ページ番号`(複数ページにわたる場合)

パラメタの内容 (hon.jp)

- この要求 URL の内容は

`&keyword=web`

キーワードの文字列。日本語を含む場合、URLエンコードが必要

`&max=5`

結果は、5件以内で表示

`&page=1`

最初のページを要求

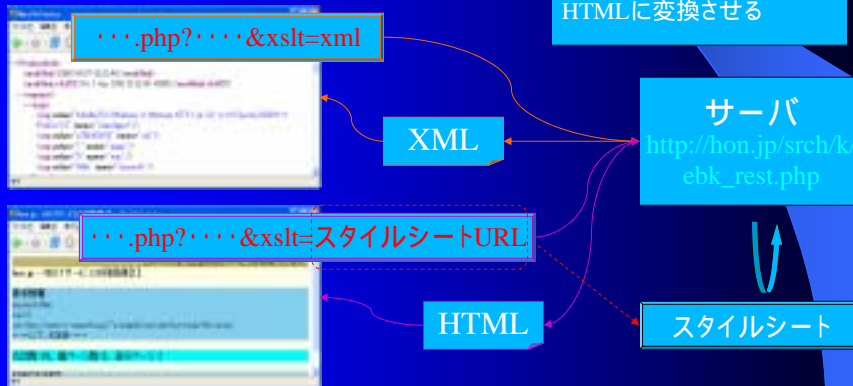
`&xslt+xml`

スタイルシート(XSLT)を指定できる。
"xml"を指定するかこのパラメタが無い場合、XMLで応答される

XSLTパラメタの指定

- **&xslt=スタイルシートのURL**

hon.jpサーバからスタイルシートを読み込んで、結果のXMLにそのXSLTでHTMLに変換させる



REST (HTTP get)の呼び出し

- パラメタを付加したURLを生成し呼び出す
 - HTMLのformタグを使う
 - HTMLのaタグを使う
 - Javascript , ECMAScriptからXMLHttpRequestを使う
 - 日本語を含む場合 , URLエンコードが必要
 - XSLTのdocument()関数を使う
 - XSLT 1.0ではURLエンコードにする機能が無い

hon.jp RESTサービスの呼び出し



この画面の
呼び出し

hon.jp 応答XMLの確認(1)

<ProductInfo>

<request>

<args>

<arg value="1" name="page"/>

<arg value="5" name="max"/>

<arg value="Web" name="keyword"/>

</args>

</request>

<totalResults>28</totalResults>

<totalPages>6</totalPages>

<results>

…… 以下、結果情報 ……

名前空間の指定は無い

/ProductInfo/request/args/arg[@name='page']/@value

XPath表現

/ProductInfo/totalResults

XPath表現

一部省略

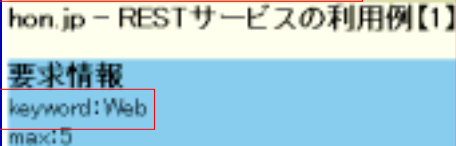
スタイルシート (1)

```

<xsl:template match="ProductInfo">
  ..... 中略 .....
  <xsl:call-template name="request"/>
  ..... 中略 .....
</xsl:template>

<xsl:template name="request">
  <div class="request">
    <div class="title">要求情報</div>
    <div>
      <xsl:text>keyword :</xsl:text>
      <xsl:apply-templates
        select="request/args/arg[ @name='keyword']/@value"/>
    </div>
    ..... 中略 .....
  </xsl:template>

```



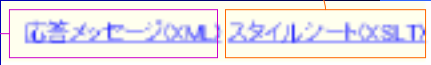
スタイルシート (2)

```

<xsl:template name="dispMsgStyle">
  <div class="xmlReq">
    <a>
      <xsl:attribute name="href">
        <xsl:call-template name="genRestUrl4RequestPageNo">
          <xsl:with-param name="pageNo" select="$thisPageNo"/>
          <xsl:with-param name="xslt">xml</xsl:with-param>
        </xsl:call-template>
      </xsl:attribute>
      <xsl:text>応答メッセージ(XML)</xsl:text>
    </a>
    <xsl:if test="$xsltUrl">
      <a href="{ $xsltUrl }">スタイルシート(XSLT)</a>
    </xsl:if>
  </div>
</xsl:template>

```

要求されたときのパラメタを参照して、XML要求を呼び出すURLを生成する。現在のページとスタイルをwith-paramで渡す。



スタイルシート (3)

```

<xsl:template match="request">
  <table>
    <xsl:for-each select="args/arg">
      <tr>
        <th>
          <xsl:value-of select="@name" />
        </th>
        <td>
          <xsl:value-of select="@value" />
        </td>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>

```

要求したパラメータ値を表示

要求情報	
UserAgent	Mozilla/5.0 (Windows; U...
uid	199U00015
page	1
xslt	http://www1.u-netsurf.n...
max	5
keyword	Web

要素名をそのまま表示

スタイルシート (4)

```

<xsl:template match="results">
  <div class="title">
    結果 [<xsl:value-of select="position()" />]
  </div>
  <table class="hon">
    <xsl:apply-templates select="*[local-name()!='details']"/>
    <xsl:for-each select="details">
      <tr>
        <th>詳細 [<xsl:value-of select="position()" />]</th>
        <td>
          <div>
            <table>
              <xsl:apply-templates/>
            </table>
          </div>
        </td>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>

<xsl:template match="results/* | details/*">
  <tr>
    <th>
      <xsl:value-of select="local-name()" />
    </th>
    <td>
      <xsl:value-of select="."/>
    </td>
  </tr>
</xsl:template>

```

結果[1]	
title	必携 HTML/CSS/JavaScr
title.kana	ひし携 HTML/CSS/Java
creator	佐藤 和人
creator.kana	さとうかずと
sourceISBN_10	4944318195
publisher	インプレス
issue_date	2004-02-16
issue_date_rfc822	Mon, 16 Feb 2004 12:00:00
url	http://hon.jp/arch/1/abk...
type	コンピュータWeb関係
hon_jp_cd	000727
url	http://

スタイルシート (5)

- HTTP (get)は、ステートレス。前後のページへリンクするには、前後用のURLを生成する必要がある
- 応答XML中にある要求パラメタ情報が役立つ。現在のページ、最後のページの情報も。

```

<request>
  <args>
    <arg value="..." name="UserAgent"/>
    <arg value="..." name="uid"/>
    <arg value="1" name="page"/>
    <arg value="xml" name="xslt"/>
    <arg value="5" name="max"/>
    <arg value="Web" name="keyword"/>
  </args>
</request>
<totalResults>28</totalResults>
<totalPages>6</totalPages>

```

1 >次 >>最終

先読<< 前<<3>次 >>最終

先読<< 前<<4

genRestUrl4RequestPageNo
が役立つ

スタイルシート (6)

```

<xsl:for-each select="args/arg">
  <xsl:variable name="requestItemLabel">
    <xsl:choose>
      <xsl:when test="@name='page'">ページ</xsl:when>
      <xsl:when test="@name='keyword'">検索文字列</xsl:when>
      .....
      <xsl:otherwise>
        <xsl:value-of select="@name"/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:variable>
  <tr>
    <th title="{@name}">
      <xsl:value-of select="$requestItemLabel"/>
    </th>
    <td>
      <xsl:value-of select="@value"/>
    </td>
  </tr>
</xsl:for-each>

```

変数の定義
(日本語を値とする)

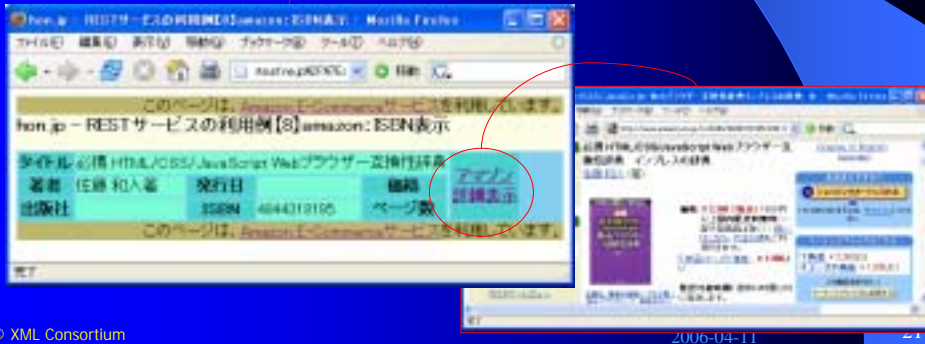
要求情報	
ユーザエージェント	Mozilla/5.0 (
ユーザ識別子	#TT9UC0015
ページ	1
XSLT指数	http://www1
表示最大件数	5
検索文字列	Web

Amazonとマッシュアップ

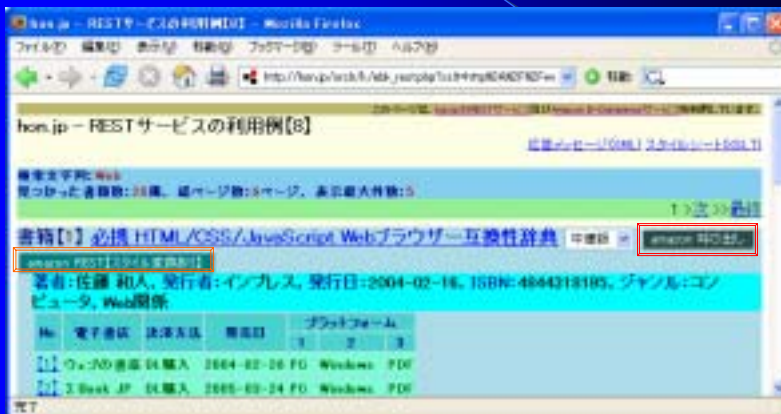
ISBN 4844318195 簡易版 amazon 呼び出し amazon REST【スタイル変更07】

発行者 sourceISBN_10

サンプルメイン
呼び出し(007)

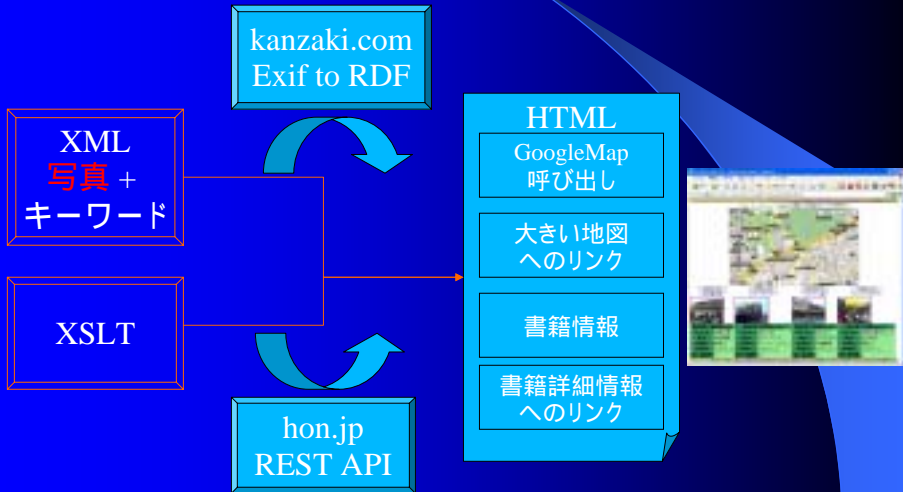


スタイル変更



サンプルメイン
呼び出し(008)

さらにマッシュアップの例 写真+RDF+地図+書籍情報



写真のメタデータ

```

<rdf:RDF
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
>
<foaf:Image rdf:about="http://.../expo2005/SN310580.JPG">
  <dc:date>2005-05-15T11:52:22</dc:date>
  <foaf:topic rdf:parseType="Resource">
    <geo:lat>35.1762194444</geo:lat>
    <geo:long>137.085819444</geo:long>
  </foaf:topic>
</foaf:Image>
  .....以下省略.....
  
```

Exif形式で
メタデータを
含む写真



作成日付

緯度

緯度

写真のメタデータを
Exif to RDFにより取り
出し、RDFとして生成

関連付けXMLデータ

```

<photoMap xmlns="urn:myPhoto">
  ... 中略 ...
  <photo href="http://.../expo2005/SN310580.JPG">
    <title>ピュッフェながくて</title>
    <description>ピュッフェながくて</description>
    <keywords>
      <keyword>長久手</keyword>
    </keywords>
  </photo>
</photoMap>
  
```

Exif to RDF

RDF



写真のタイトル

地図上に表示

hon.jpで検索

電子書籍	電子書籍
タイトル	社会問題(十一)
著者	吉川純吉
出版社	講談社
発行日	2000-01-01
電子書籍ID	6



マッシュアップページ



サンプル呼び出し

地図の切り替え

GoogleMapの表示

地図上の場所移動

写真の表示

関連書籍の表示

マッシュアップの魅力

- 公開されたサービスを容易に利用できる
- 何かをトリガーに複数のサービスを組み合わせる
- 新たな姿で、新たな価値を創出する
- それをヒントに新たなサービスを派生する
- 一人で考えるのではなく、多くの人々の知恵が活かせる