PerlでWWW::MechanizeとWeb::Scraperを使いWebページのクロールをやってみる

2012/06/11
PerlでWWW::Mechanize と Web::Scraperを使ったサイトクロールのサンプル。 内容はRSSで取ったほうが早いけど、あくまでサンプルなので。

2013/04/06
こちらのドキュメントは管理都合によりPerl関連ドキュメントのネームスペースへ移動しました。

Yahoo!ニュースのトピック切り出し

とりあえずこんなコードは良く見かけますが例を挙げておきますyatopic.plとでもしておきましょうか。

yatopic.pl

```
use Encode;
use WWW::Mechanize;
use Web::Scraper;

my $baseUrl  = 'http://dailynews.yahoo.co.jp';
my $yahooUrl = 'http://dailynews.yahoo.co.jp/fc/';
my $encstr   = 'euc-jp';
my $mech     = WWW::Mechanize->new(autocheck=>1);

my $topicsMenuParse = scraper {
      process '//div[@id="globalNav"]/ul[@id="gnSec"]//li',
"menulinks[]"=> scraper {
           process "a", href=>'@href';
           process "a", text=>"TEXT";
      }
};

my $topicsLinkParse = scraper {
      process
'//div[@id="topics"]/div[@class="topicsList"]/ul[@class="clr"]//li',
"topiclinks[]"=> scraper {
           process "span", date=>"TEXT";
           process "a",    href=>'@href';
           process "a",    text=>"TEXT";
      }
};

my $menuResult;

  $mech->agent_alias("Windows Mozilla");
```

```
$mech->get( $yahooUrl );
$menuResult = $topicsMenuParse->scrape( $mech->content );

foreach my $categoryLink ( @{ $menuResult->{menulinks} } )
{
    my $topicResult;
    my $text = encode($encstr, $categoryLink->{text});
    my $href = encode($encstr, $categoryLink->{href});

      printf("*** %s ( %s )\n",  $text, $href);

      $mech->get( "".$categoryLink->{href} );
      $topicResult = $topicsLinkParse->scrape( $mech->content );

      foreach my $topicLink ( @{ $topicResult->{topiclinks} } )
      {
         my $date = encode($encstr, $topicLink->{date});
         my $text = encode($encstr, $topicLink->{text});
         my $href = encode($encstr, $topicLink->{href});

           printf("%s %s ( %s )\n",  $date, $text, $baseUrl.$href);
      }
}
```

説明

yatopic.plでは、WWW::MechanizeでYahoo!ニュースのトピックスページを取得、Web::Scraperで利用するHTML要素の切り出しを行っています。

URL http://dailynews.yahoo.co.jp/fc/ がYahoo!ニュースのトピックスページになります。 トピックスは

- 国内
- 海外
- 経済
- エンターテインメント
- スポーツ
- コンピュータ
- サイエンス
- 地域
- バックナンバー
- 編集センター

に別れている為、各々のトピックスへのリンクを取得する必要があります。 HTML中から該当する箇所を切り出ししますが、そのHTMLの切り出し位置をXPathで指定します。

```
//div[@id="globalNav"]/ul[@id="gnSec"]//li
```

DIVタグのid属性が “ globalNav”で、その内包要素のULタグ、id属性“gnSec”のLIタグにそのリンクがあり

ますので、各LIタグ以下のコレクションを作りキーmenulinksで保持します。そのコレクションの内容は、Aタグのhref属性、テキスト、となっています。

次にこのコレクションのリンクで指すページ(HTML)を一つずつ取得し、トピックの情報を切り出し、出力します。

```
//div[@id="topics"]/div[@class="topicsList"]/ul[@class="clr"]//li
```

id属性が“ topics”のDIVタグ以下にあるid属性“topicsList”のDIVタグにULタグ□class属性“clr”があります。この配下のLIタグに日時とテキスト、記事へのリンクががありますので、各LIタグ以下のコレクションを作りキーtopiclinksで保持します。そのコレクションの内容は、SPANタグのテキストにある日時□Aタグのhref属性、テキスト、となっています。

技術資料, Perl, Mechanize, Scraper, スクレイピング