

## 16.2 Finding Photos on Flickr

### Problem

You want to use Rubycode to find freely reusable photos: perhaps to automatically illustrate a piece of text.

### Solution

The Flickr photo-sharing web site has a huge number of photos and provides web services for searching them. Many of the photos are licensed under Creative Commons licenses, which give you permission to reuse the photos under various restrictions.

There are several Ruby bindings to Flickr's various web service APIs, but its REST API is so simple that I'm just going to use it directly. Given a tag name (like "elephants"), this code will find an appropriate picture, and return the URL to a thumbnail version of the picture.

First, a bit of setup. As with Amazon and Google, to use the Flickr API at all you'll need to sign up for an API key (see below for details).

```
require 'open-uri'
require 'rEXML/document'
require 'cgi'
FLICKR_API_KEY = 'Your API key here'
```

The first method, `flickr_call`, sends a generic query to Flickr's REST web service. It doesn't do anything special: it just makes an HTTP GET request and parses the XML response.\*

```
def flickr_call(method_name, arg_map={}.freeze)
  args = arg_map.collect {|k,v| CGI.escape(k) + '=' + CGI.escape(v)}.join('&')
  url = "http://www.flickr.com/services/rest/?api_key=%s&method=%s&%s" %
  [FLICKR_API_KEY, method_name, args]
  doc = REXML::Document.new(open(url).read)
end
```

Now comes `pick_a_photo`, a method that uses `flickr_call` to invoke the `flickr.photos.search` web service method. That method returns a REXML Document object containing a `<photo>` element for each photo that matched the search criteria. I use XPath to grab the first `<photo>` element, and pass it into `small_photo_url` (defined below) to turn it into an image URL.

```
def pick_a_photo(tag)
  doc = flickr_call('flickr.photos.search', 'tags' => tag, 'license' => '4',
  'per_page' => '1')
  photo = REXML::XPath.first(doc, '//photo')
  small_photo_url(photo) if photo
end
```

Finally, I'll define the method, `small_photo_url`. Given a `<photo>` element, it returns the URL to a smallish version of the appropriate Flickr photo.

```
def small_photo_url(photo)
  server, id, secret = ['server', 'id', 'secret'].collect do |field|
  photo.attribute(field)
  end
  "http://static.flickr.com/#{server}/#{id}_#{secret}_m.jpg"
end
```

Now I can find an appropriate photo for any common word:

```
pick_a_photo('elephants')
# => http://static.flickr.com/32/102580480_506d5865d0_m.jpg
pick_a_photo('what-will-happen-tomorrow')
# => nil
```

\* Some of Flickr's APIs let you do things like upload photos and add comments. You'll need to use POST requests to make these calls, since they modify the state of the site. More importantly, you'll also need to authenticate against your Flickr account.

## Discussion

It's nice if there's a predefined Ruby binding available for a particular REST-style web service, but it's usually pretty easy to roll your own. All you need to do is to craft an HTTP request and figure out how to process the response document. It's usually an XML document, and a well-crafted XPath statement should be enough to grab the data you want.

Note the clause `license=4` in `pick_a_photo`'s arguments to `flickr_call`. I wanted to find a picture that I could publish in this book, so I limited my search to pictures made available under a Creative Commons "Attribution" license. I can reproduce that picture of the elephants so long as I credit the person who took the photo. (Nick Scott-Smith of London. Hi, Nick!)

Flickr has a separate API call that lists the available licenses (`flickr.licenses.getInfo`), but once I looked them up and found that "Creative Commons Attribution" was number four, it was easier to hardcode the number than to look it up every time.

## See Also

- The first few recipes in Chapter 11 demonstrate different ways of extracting data from XML documents; XPath (Recipe 11.4) and Rubyful Soup (Recipe 11.5) let you extract data without writing much code
- Recipe 14.1, "Grabbing the Contents of a Web Page"
- Sign up for a Flickr API key at <http://www.flickr.com/services/api/key.gne>
- Flickr provides REST, XML-RPC, and SOAP interfaces, and comprehensive documentation of its API (<http://www.flickr.com/services/api/>)
- The Flickr URL documentation shows how to turn a `<photo>` element into a URL (<http://www.flickr.com/services/api/misc.urls.html>)
- `Flickr.rb` (<http://redgreenblu.com/flickr/>; available as the flickr gem), the libyws project (<http://rubyforge.org/projects/libyws>; check out from CVS repository), and `rflickr` (<http://rubyforge.org/projects/rflickr/>; available as the rflickr gem)
- A brief explanation of the Creative Commons licences (<http://creativecommons.org/about/licenses/meet-the-licenses>)