

Lightweight Language Saturday



Apacheログ解析 (Ruby)

2003-8-9

たけうちひとし

hitoshi@namaraii.com

処理の流れ

```
66 | if __FILE__ == $0
67 |   include Report
68 |   ip_addr = ARGV.shift
69 |   access_log = AccessLog.new(ARGV)
70 |   access_log.ip_addr = ip_addr
71 |   report(access_log.sort)
72 | end
```

- コマンドライン引数から解析対象のIPアドレスを取得(68)
- ARGV(69)
 - コマンドライン引数で指定された文字列をファイル名と解釈
 - ひとつかたまりの仮想的なIOオブジェクトとして扱える
- 指定されたIPアドレスの"アクセス日時"を抽出(71)
- "滞在時間"と"訪問回数"を算出(71)

AccessLogクラス

```
3 | class AccessLog
4 |   include Enumerable
5 |   attr_accessor :ip_addr
7 |   LogFormat = /%A(%S+) %S+ %S+ %[[^%]]+%.*/
8 |   DateFormat = %r!%A(%d%d)/([A-Z][a-z][a-z])/(%d%d%d%d):(%d%d):(%d%d):(%d%d)!
9 |   Month      = %w(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec)
10 |
11 |   def initialize(f)
12 |     @logfile = f
13 |     @ip_addr = ""
14 |   end
15 |
16 |   def parse_time(time)
17 |     m = DateFormat.match(time)
18 |     Time.local(m[3].to_i, Month.index(m[2])+1, m[1].to_i, m[4].to_i, m[5].to_i, m[6].to_i)
19 |   end
20 |
21 |   def each
22 |     @logfile.each do |line|
23 |       m = LogFormat.match(line)
24 |       yield parse_time(m[2]) if m and m[1] == @ip_addr
25 |     end
26 |   end
27 | end
```

アクセス日時の抽出 (イテレータ)

```
3 | class AccessLog
    (略)
21 | def each
22 |     @logfile.each do |line|
23 |         m = LogFormat.match(line)
24 |         yield parse_time(m[2]) if m and m[1] == @ip_addr
25 |     end
26 | end
27 | end
```

```
3 | class AccessLog
4 |   include Enumerable
      (略)
27 | end
      (略)
69 |   access_log = AccessLog.new(ARGV)
70 |   access_log.ip_addr = ip_addr
71 |   report(access_log.sort)
```

- 要素ごとの繰り返しを行うクラスのためのモジュール
- eachメソッドを定義してinclude
- collect, find, sortなどが追加される(Mix-in)

レポート出力

```
38 | def report(log)
39 |   visits = count = elapsed_time = 0
40 |   first_time = previous_time = log.first
41 |
42 |   log.each do | current_time |
43 |     if current_time - previous_time > INTERVAL
44 |       elapsed_time += output_log(first_time, previous_time, count)
45 |       first_time = previous_time = current_time
46 |       visits += 1
47 |       count = 1
48 |     else
49 |       count += 1
50 |       previous_time = current_time
51 |     end
52 |   end
53 |   if first_time == previous_time or count > 1
54 |     visits += 1
55 |     elapsed_time += output_log(first_time, previous_time, count)
56 |   end
57 |
58 |   puts <<EOS
59 | total: #{elapsed_time.to_i} sec
60 | visits: #{visits}
61 | average: #{"%.2f" % (elapsed_time/[visits, 1].max)} sec
62 | EOS
63 | end
```

- 本格的なテキストフィルタから使い捨てのフィルタまで
 - ARGF
 - 文字列操作
 - 正規表現