Harnessing the power of the Web

Web automation and Libwww-perl

Nadav Har'El

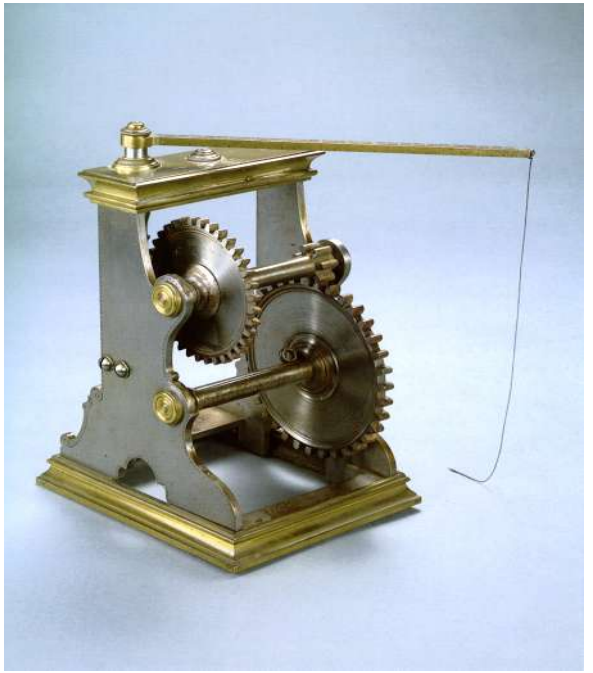
IBM, February 24, 2004

Outline of this talk

- Web automation:
  - What is it?
  - Why is it useful?
  - Examples
  - Implementations
- A Libwww-perl primer

Programming & Automation

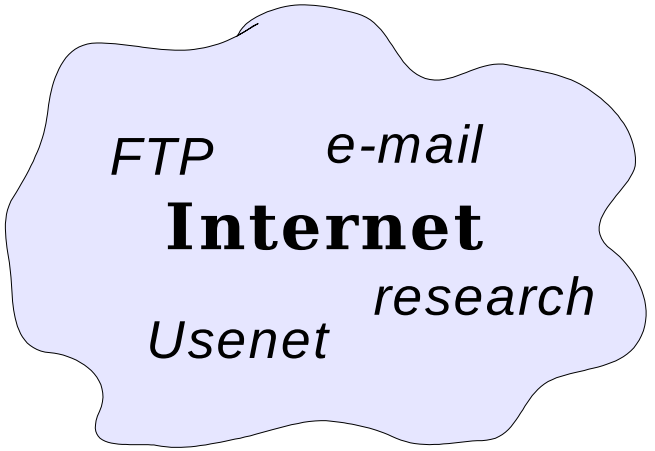
- Programming is fun, more so when useful.
- Automation is useful.



Historic Survey

1969-1993 – pre-Web Internet:

- ARPANET went online in 1969.
- Internet separate from real-life:



Real-life

bank
friends
family
government
neighbor
phone company

Historic Survey

1969-1993 – pre-Web Internet:

- ARPANET went online in 1969.
- Internet separate from real-life.
- 1989 – Tim Burners-Lee, WWW
  - *Concepts:* URL, HTML
  - *Pros:* Intuitively jump from content to content. Not just text. Interactive.
  - For ordinary people, not just experienced researchers.

Historic Survey

1969-1993 – pre-Web Internet:

- ARPANET went online in 1969.
- Internet separate from real-life.
- 1989 – Tim Burners-Lee, WWW.
- 1990-1992 – Tim Burners-Lee, HTTP
- 1993 – Marc Andereessen (NCSA) – Mosaic, first graphical *Browser*.
- Supply and demand spiral begins:

Historic Survey

1993-1998 – early Web

- Growth in content and readers:
  - content ⇒ curious users try Mosaic
  - ordinary people get commercial ISPs
  - Mosaic used ⇒ people and companies want homepage
  - Starting with small advertising page
  - More users ⇒ interactive pages, services, commerce.

Historic Survey

Today – In industrialized countries,

- Internet is commonplace
- Much of population connected
- Companies and government expected to provide info and services online.

Historic Survey

- Wielding a Web browser, the world is at your fingertips:
  - Stock quotes
  - Newspapers
  - Bank statement
  - Send SMSs
  - Order a book from seller abroad
  - Order food from local grocery store

Historic Survey

- New Internet-only tools beginning to impact “real-life” activities and relationships:
  - ICQ
  - Search engines
  - Ebay (Person-to-person selling)

Today - The power of automation

- With advent of Web information and services, comes unique opportunity: **Automation**.
- Finding when bank balance is low:
  - Hard, annoying in real-life (teller, ATM)
  - Easy, annoying with Web interface
  - Easy when surfing session automated.
- Harness the power of the web.

Today - The power of automation

- Programmers can create automatons themselves.
- Sites appear that do nothing but automate other sites. (book renewal, bid sniping, etc.)
- In the future, might be simple enough for non-programmers.

The future

- The automation described so far: software mimics a human browsing.
- We have
  - Two computers (Web server, automation program)
  - Communicating through human language (Web pages with text and graphics).

The future

- The problem:
  - Wasteful, complicated.
  - Deal with human-aimed UI changes.
- Example: Amazon.com
  - Virtual-Store builders wanted to extract book lists and information.
  - Web site looks and interface changed often.

The future

- The proposed solution:
  - Dubbed “*Web Services*”
  - Requests and answers are in XML, in strict formats.
  - Aimed for computer, no visual “junk”, stable interface.
- Amazon.com started Web Services interface in 2002.
- Parallel to its normal Web interface.

The future

- Will Web Services be adopted?
- Problem: On most sites,
  - Normal Web interface is done first.
  - *Web Services* done as afterthought.
  - Doesn't cover everything, if at all.
- Solutions? (in the future)
  - Do Web Services first, build Web interface on it. Cf. Unix philosophy.
  - Develop them together.

Examples

- Some real-life useful examples
- Done by my friends or me.
- Implemented with Libwww-perl and other mechanisms.

Example 1
Renewing Library Books

- Early 90s: “Aleph” library network. Ad-hoc Telnet interface.
- Why not renew automatically?

Example 1
Renewing Library Books

- Early 90s: “Aleph” library network. Ad-hoc Telnet interface.
- Why not renew automatically?
- Expect/TCL automation – renewal.
- Central renewal service.
- Recently, Aleph Web interface.
- One page fetch renews books (*curl*)

Example 2
Sending SMSs

- How are we to be informed of event? E.g., library book cannot be renewed
- Email ill-suited for both light and heavy users.
- Many people do not use email.
- Pagers – good but did not catch on.

Example 2
Sending SMSs

- In 1999, “Short Message Service” becomes available in Israel.
- modems no longer in vogue. Mobile providers give Web interface.
- SendSMS script automates it.
- SendSMS used for notification, including email.
- SendSMS still works, and free, today.

Example 3
Checking your bank balance

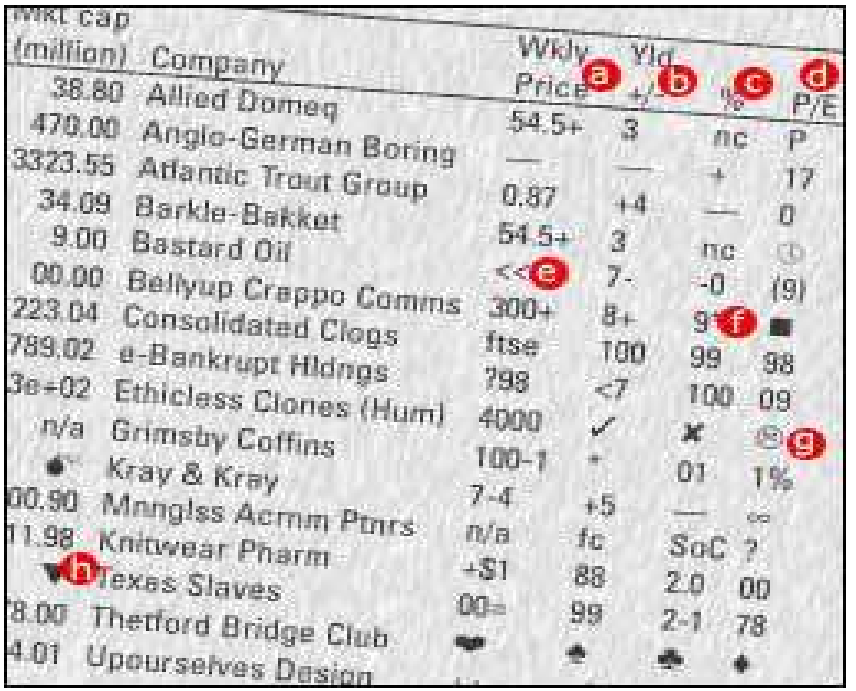
- “Long ago” – bank records on paper.
- Until mid 90s: phone, or ATM.
- Mid 90s: modem connection, propriatary software.
  - Check account balance
  - Check investments, stocks, etc.
- End of 90s: easier, standard, more flexible, Web interface.

Example 3
Checking your bank balance

- Israeli bank sites automated with libwww-perl (Dan Kenigsberg, Alon Altman). Example uses:
  - Get notified when balance is low
  - Get balance every day
  - Get notified when a check is cashed
  - Extract information quickly, without manual navigation of Web site

Example 4
Stocks, funds and price indices

- Newspapers dedicate a few pages to latest prices of
  - Stocks and bonds
  - Mutual funds
  - Foreign currency
- Also, monthly:
  - Price index
- Tedious to follow daily.



Mkt cap (million)	Company	Wkly Price (a)	Yld +/- (b)	% (c)	P/E (d)
38.80	Allied Domeq	54.5+	3	nc	P
470.00	Anglo-German Boring	—	—	+	17
3323.55	Atlantic Trout Group	0.87	+4	—	0
34.09	Barkle-Bakket	54.5+	3	nc	ⓘ
9.00	Bastard Oil	<<(e)	7-	-0	(9)
00.00	Bellyup Creppo Comms	300+	8+	9(f)	■
223.04	Consolidated Clogs	ftse	100	99	98
789.02	e-Bankrupt Hldngs	?98	<7	100	09
3e+02	Ethicless Clones (Hum)	4000	✓	✘	(g)
n/a	Grimsby Coffins	100-1	+	01	1%
♠	Kray & Kray	7-4	+5	—	∞
00.90	Mnngiss Acmm Ptnrs	n/a	fc	SoC	?
11.98	Knitwear Pharm	+$1	88	2.0	00
(h)	Texes Slaves	00=	99	2-1	78
8.00	Thetford Bridge Club	♥	♠	♣	♦
4.01	Upourselves Design				

Example 4
Stocks, funds and price indices

- Easier to follow online:

Splits:01-Jun-79 [4:1], 28-May-97 [2:1], 27-May-99 [2:1]

Last Trade:	**97.31**
Trade Time:	Feb 20
Change:	↓ 0.49 (0.50%)
Prev Close:	97.80
Open:	98.60
Bid:	N/A
Ask:	N/A
1y Target Est:	108.50

Day's Range:	97.19 - 98.60
52wk Range:	73.17 - 100.43
Volume:	5,690,200
Avg Vol (3m):	5,356,636
Market Cap:	167.41B
P/E (ttm):	22.43
EPS (ttm):	4.339
Div & Yield:	0.64 (0.65%)

Example 4
Stocks, funds and price indices

- Even easier when automated:
  - Get daily quotes of stocks of interest
  - Get notified on certain event (e.g., some stock changed by 10%)
- Not only easy, also free.

Example 5
Following bills

- Credit-based services, variable and periodic bills:
  - Phone, cellular
  - Credit card, calling card
  - Cable TV
  - Electricity, water, gas

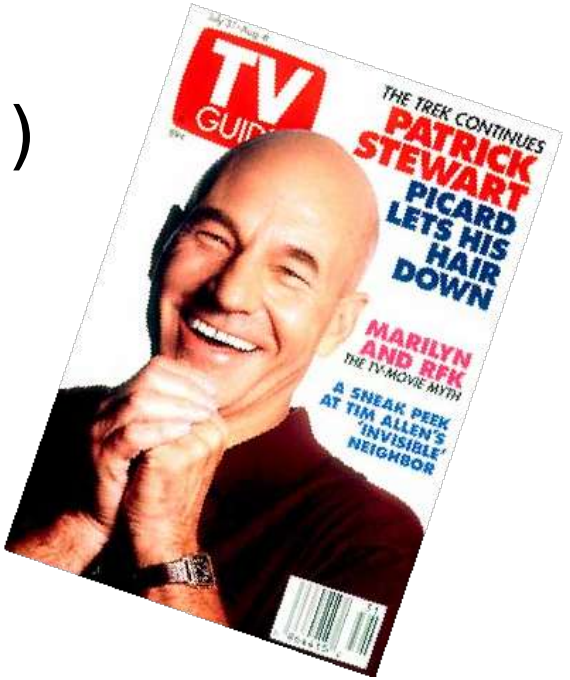
- Websites provide up-to-the-minute bills.

Example 5
Following bills

- Some uses of automation:
  - Daily summary of credit card charges.
  - Monitor child's cellphone bill.
  - Check for suspicious activity (e.g., someone using your phone during the night)

Example 6
Directories and schedules

- Real-life information. Now on the Web, a few clicks away:
  - Phone directories (411, 144)
  - Zipcode directories
  - Bus and train schedules
  - TV schedules
  - Movie screening times
- All this information is free.



Example 6
Directories and schedules

- Example ways to automate:
  - Get weekly mail of your favorite show's airing times.
    Get SMS a few minutes before it starts.
  - Find phone numbers of list of people.
  - Get alert when some movie comes to a cinema near you.
  - Fetch schedule of your favorite bus, without a lengthy browsing session.

Example 7
Electronic ballot stuffing

- During 2000, “picture of the week”.
- December 2000 – early 2001: “The Year in Pictures 2000”.

Example 7
Electronic ballot stuffing

- Could have been an uneventful poll
- but became a political battleground because of one candidate:

Example 7
Electronic ballot stuffing

"A death in Gaza"

- Sep 30, Gaza strip. Jamal and Mohammed Al-Durah.

Example 7
Electronic ballot stuffing

- Political battle ensued:
  - Palestinian plea: vote "*A death in Gaza*"
  - "*A death in Gaza*" takes lead.
  - Israeli plea: defeat Palestinian voting campaign – vote for anything else.
  - Israeli chain letter claiming:
    - Palestinians organized voting.
    - Nobody can vote twice.
    - Vote, and ask your friends to vote.

Example 7
Electronic ballot stuffing

– One Israeli takes this as a challenge, automates voting. 1000s votes/hour, several millions in a week.

Example 7
Electronic ballot stuffing

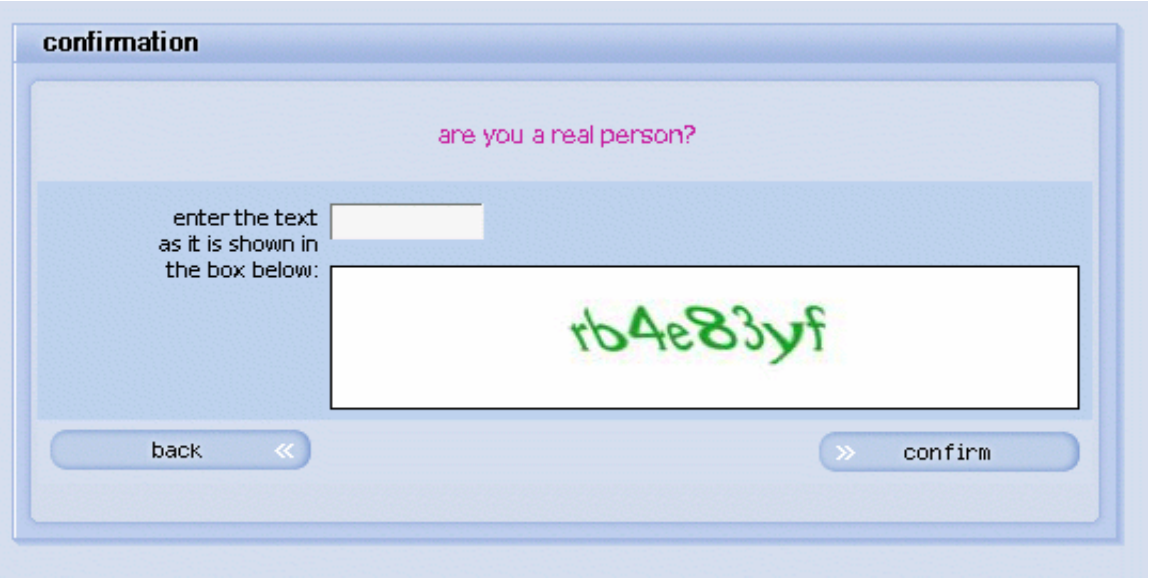
- One Israeli takes this as a challenge, automates voting. 1000s votes/hour, several millions in a week.
- Animal photos take top 5 places

Example 7
Electronic ballot stuffing

- One Israeli takes this as a challenge, automates voting. 1000s votes/hour, several millions in a week.
- Animal photos take top 5 places
- Saudi-Arabian “fights back”
- MSNBC cancels poll.
- Media covers the incident:
  Reported by New York Times, AP, Jerusalem Post, Al-Ahram.

Example 7
Electronic ballot stuffing

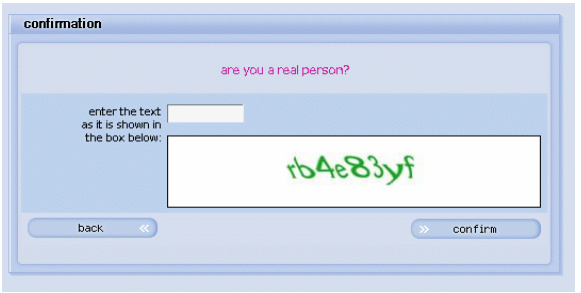
- Some sites now have “human-detection” to resist automatons:



(orkut.com)

Example 7
Electronic ballot stuffing

- Some sites now have “human-detection” to resist automatons:



- But:
  - Some humans can't pass it.
  - Eventually, computers could pass it.

Example 8
Bid sniping

- Competitive ecommerce => try business and pricing models.
- 

: online auction house.
- To a real auction house, you send an agent.
- *Mobile Agents* Have been proposed. Ebay doesn't support them.
- Ebay's agent raises up to max bid.

Example 8
Bid sniping

- Bidding strategy – when to bid?
  - Early:
    - reveals your interest
    - and lets opponent react.
  - Late: (Ebay does not extend auctions)
    - Hides your intention from opponents
    - Opponent has no time to change instructions
- Late is better.
  How bid late, without mobile agent?

Example 8
Bid sniping

- Simple: write program to make bid at prescribed time.
- Commercially termed "*bid sniping*".
- Web automation = non-mobile agent
- Non-mobile agent can be much more sophisticated:
  - React to opponents raised bids.
  - Use of historic data on similar auctions.

Implementations

- Task: write a program that pretends to be a user browsing a Web site.

Implementations

- Task: write a program that pretends to be a user browsing a Web site.
- Solution 1:
  Low-level API for
  - Fetching pages
  - Submitting forms
  - Handling cookies
  - Parsing HTML
  - etc.

Implementations

Solution 1 (Low-level API)

- Libwww-perl (Perl)
- Libcurl (C and other languages)

Implementations

Solution 1 (Low-level API)

- Advantages:
  - Powerful, flexible
- Disadvantages:
  - Relatively hard to program (e.g., forms)
  - Rather explicit (e.g.,. cookie jar)
  - Requires reading HTML and sniffing.
  - Hard to find cause of malfunction.

Implementations

- Solution 2: Automating real browser.
- Example: Lynx and Expect.
- Advantages:
  - Cookies, forms, redirection: automatic.
  - Understandable – normal browser.
- Disadvantages:
  - Harder to control (errors, page loads).
  - Deal with browser's UI idiosyncrasies.

Implementations

- Solution 3: Shell script.
- Example: Curl and shell.
- Advantages:
  - Very easy for simple tasks.
- Disadvantages:
  - Hard for anything else.

Implementations

- Solution 4: meta-language for describing common interaction types (login, etc.)
- Example: Kamajii.
- Solution 5: Recording real user sessions, replaying with modified parameters.

Libwww-perl

Example 1:

Find latest known price of an American stock, given ticker symbol.

```
$ quote GM
```

49.21

```
$ quote '^DJI'
```

10,598

```
$ quote XYZ
```

quote: XYZ is not a valid ticker symbol.

Libwww-perl

- Start by manually browsing the site.
- Assessing what login forms need to be filled, whether cookies are in use, etc.
- In this example, we're in luck: for GM quote, only need to fetch http://finance.yahoo.com/q?s=GM
- Few Libwww-perl features needed.

Libwww-perl

- *Check arguments:*

```
if($#ARGV!=0){
  print STDERR "Usage: $0 <symbol>\n";
  exit(1);
}
```

- *Libwww-perl is OO, implementing classes for requests, responses, cookie jar, etc.*

```
use LWP::UserAgent;
my $ua = new LWP::UserAgent;
```

Libwww-perl

- *Make request:*

```
my $request =
  HTTP::Request->new('GET',
  "http://finance.yahoo.com/q?s=$ARGV[0]");
my $res = $ua->request($request);
```

- *Check for successful response:*

```
if(!$res->is_success){
  print STDERR "Can't get $ARGV[0] from".
   “Yahoo:\n".$res->status_line."\n";
  exit(2);
}
```

Libwww-perl

- *While developing,* print $res->content;
- *We end up with:*

```
if($res->content =~ /not a valid ticker symbol/){
    print "$ARGV[0] is not a valid ticker symbol.\n";
    exit(3);
} elsif($res->content =~
  /(Last Trade|Index Value):(<[^>]*>)*([0-9][0-9.,]*)/){
  print "$3\n";
} else {
  print "unexpected content in $ARGV[0] page.\n";
  print STDERR $response->content;
  exit(3);
}
```

Libwww-perl

- Note: we parsed HTML with Perl. HTML::Parser (et al.) also available.
- Libwww-perl has good manual pages
  - Start with LWP(3)
  - For each class: LWP::UserAgent, HTTP::Request, HTTP::Response.

Libwww-perl

Example 2:
SendSMS, simplified, using ICQ Web interface (Cellcom and Pelephone)

- Usage: sendsms *num message*
- *Modules:*

```
use LWP::UserAgent;
use URI::Escape;
use HTTP::Cookies;
```

Libwww-perl

- *Argument parsing:*

```
die "Usage: $0 phonenum message\n" if
($#ARGV+1 != 2);
my $phonenum=$ARGV[0];
$phonenum =~ s/[ ()-]//go;
my $message=$ARGV[1];
```

- *To be configured:*

```
my $user = '123456';
my $password = 'paSwOrD';
```

Libwww-perl

- *User Agent object:*
  ```
  my $ua = new LWP::UserAgent;
  $ua->agent("Mozilla/4.73 [en] (Win95; I)");
  $ua->env_proxy();
  ```
- Going to http://web.icq.com/sms/ we see a login form.
- Submitting the form is an HTTP request of type POST, “url-encoded”:

Libwww-perl

```
$req = new HTTP::Request POST=>
  "http://web.icq.com/newlogin/1,,,00.html";
$req->content_type('application/x-www-form-
urlencoded');
$req->content(
"karma_user_login=".uri_escape($user, '^A-Za-z0-9')."&".
"karma_user_passwd=".uri_escape($password, '^A-Za-z0-9')."&".
"lang=eng&karma_product_id=21&karma_success_url=http%3A%2F%
2Fweb.icq.com%2Fsms%2Finbox%2F%3Fdsfp%3D0&karma_fail_url=%
2Flogin%2Flogin_page%3Fkarma_product_css%3Dicq2go%
26karma_success_url%3Dhttp%253A%252F%252Fweb%252Eicq%
252Ecom%252Fsms%252Finbox%252F%253Fdsfp%253D0%
26karma_forget%3D%26karma_service%3D&karma_service=");

$res = $ua->request($req);
```

Libwww-perl

- *On success, we see redirection:*

```
if($res->code!=301 ||
   $res->header('location') !~ m@/sms/inbox/@){
      print STDERR "Failed login to ICQ\n";
      exit 1;
}
```

- *Remember cookies to send later:*

```
my $cookie_jar = HTTP::Cookies->new;
$cookie_jar->extract_cookies($res);
```

- “Detective work” continues (show source, sniffer, LiveHeaders, etc.)

Libwww-perl

- *Fill message-sending form. Use cookies*.

```
$req = new HTTP::Request POST =>
"http://web.icq.com/sms/send_msg_tx/1,,,00.html";
$req->content_type('application/x-www-form-urlencoded');
$req->content("country=972&prefix=%
2B972&uSend=1&charcount=".(160-length
($message))."&".
  "carrier=".substr($phonenum,1,2)."&".
  "tophone=".substr($phonenum,3)."&".
  "msg=".uri_escape($message, '^A-Za-z0-9'));
$cookie_jar->add_cookie_header($req);
$res = $ua->request($req);
```

Libwww-perl

- *Finally, check success:*

```
if($res->code!=301 ||
 $res->header('location') !~ m@^/sms/thanks/@){
    print STDERR "Failed to send message\n";
    exit 1;
}
print STDERR "Sent successfully.\n";
```