

# Exploiting Other Enterprises' Web Services

Léon Brocard  
YAPC::Taipei 2005



# Abstract

Web Services are the next wave of enterprise development. Large companies such as Google, Amazon and Yahoo! are solving the challenges of developing, deploying, and supporting complex services in the face of interoperability and scalability challenges. Seeing as they've gone to all this trouble, it's only fair to present how we can exploit all this hard work using a few lines of Perl.



# Abstract

**Web Services** are the next wave of **enterprise** development. Large companies such as Google, Amazon and Yahoo! are solving the challenges of developing, deploying, and supporting complex services in the face of interoperability and scalability challenges. Seeing as they've gone to all this trouble, it's only fair to present how we can **exploit** all this hard work using a few lines of **Perl**.



# Web service?

- Web services are loosely-coupled software components delivered over standard internet technologies
- Intended for use by programs
- Scraping is not a (really) web service



# Enterprise?

- An enterprise is an organisation that uses computers, but usually a large organisation
- We'll concentrate on enterprise web services for reliability



# Exploit?

- Use or manipulate to one's advantage
- In this case, for my website



# Perl?

- You probably know about Perl already
- A dynamic programming language
- Often used to glue libraries together
- Handy for web services
- CPAN saves the day

# Types of web service

- SOAP
  - W3C standard, evil specification
  - Intended for static languages, seems overkill for Perl
- REST
  - HTTP+XML
  - Simpler than SOAP, but wheel reinvention



# Are you the Keymaster?

- Some web services are open
- Most enterprise web services require you to sign up to receive a key, which you hand over upon every request

# Web services

amazon.com.

Google™

YAHOO!®

The word "Google" is written in its characteristic white, multi-colored font, centered over a background of green bamboo branches against a clear blue sky. The branches are thin and dark, with many long, narrow green leaves.

# Google

- Search Requests
- Cache Requests
- Spelling Requests
- All sorts of scrape-able data too



# Google limits

- 1,000 queries per day



# Google search

- I have 100k recipes on [www.astray.com](http://www.astray.com)
- I typed them all in
- Laziness is a virtue
- Line 2 is untrue
- DOS recipe program: MealMaster
- Easily parsable recipe files...
- So let's find them with Google!

# Find recipes

```
my $search = WWW::Search->new('Google', key => $key);
$search->native_query("mealmaster recipes");
my $count = 0;

while (my $result = $search->next_result()) {
    last if $count++ > 100; # limits!
    my $extor = HTML::SimpleLinkExtor->new();
    $extor->parse(get($result->url));
    foreach my $link ($extor->links) {
        next unless $link =~ /\.zip$/;
        my $uri = URI::WithBase->new($link, $result->url);
        print $uri->abs . "\n";
    }
}
```

# Google cache

```
use Net::Google::Cache;  
my $cache = Net::Google::Cache->new( {  
    key => $key  
} );  
$cache->url( "http://www.astray.com/" );  
print $cache->get();
```

# Google spelling

- I have 100k recipes on [www.astray.com](http://www.astray.com)
- You can search for recipes
- Apparently ~10% of all queries on search engines are misspelled - more like 30%!
- Accurate spelling correction is tricky, so exploit Google's hard work with Net::Google::Spelling
- Backup spelling correction (> API limit)

# Net::Google::Spelling

```
use Net::Google::Spelling;
my $spelling = Net::Google::Spelling->new({
    key => $key
});
$spelling->phrase("steak bernayes recipe");
print $spelling->suggest()."\n";
# steak bernaise recipe
```

# Action shot

**astray**

[Curriculum Vitæ](#) · [Journal](#) · [Perl Modules](#) · [Photos](#) · **[Recipes](#)**

No recipes found while searching for "steak bernayes". Did you mean: [steak bernaïse](#)

The title "Google Maps" is centered in a large, white, sans-serif font. It is overlaid on a background image of green bamboo leaves and thin branches against a clear blue sky.

# Google Maps

- Recently introduced
- Pretty maps (USA only)
- `WebService::GoogleMaps` does the nasty calculations for us
- ... but we need latitude, longitude

# WWW::Gazetteer

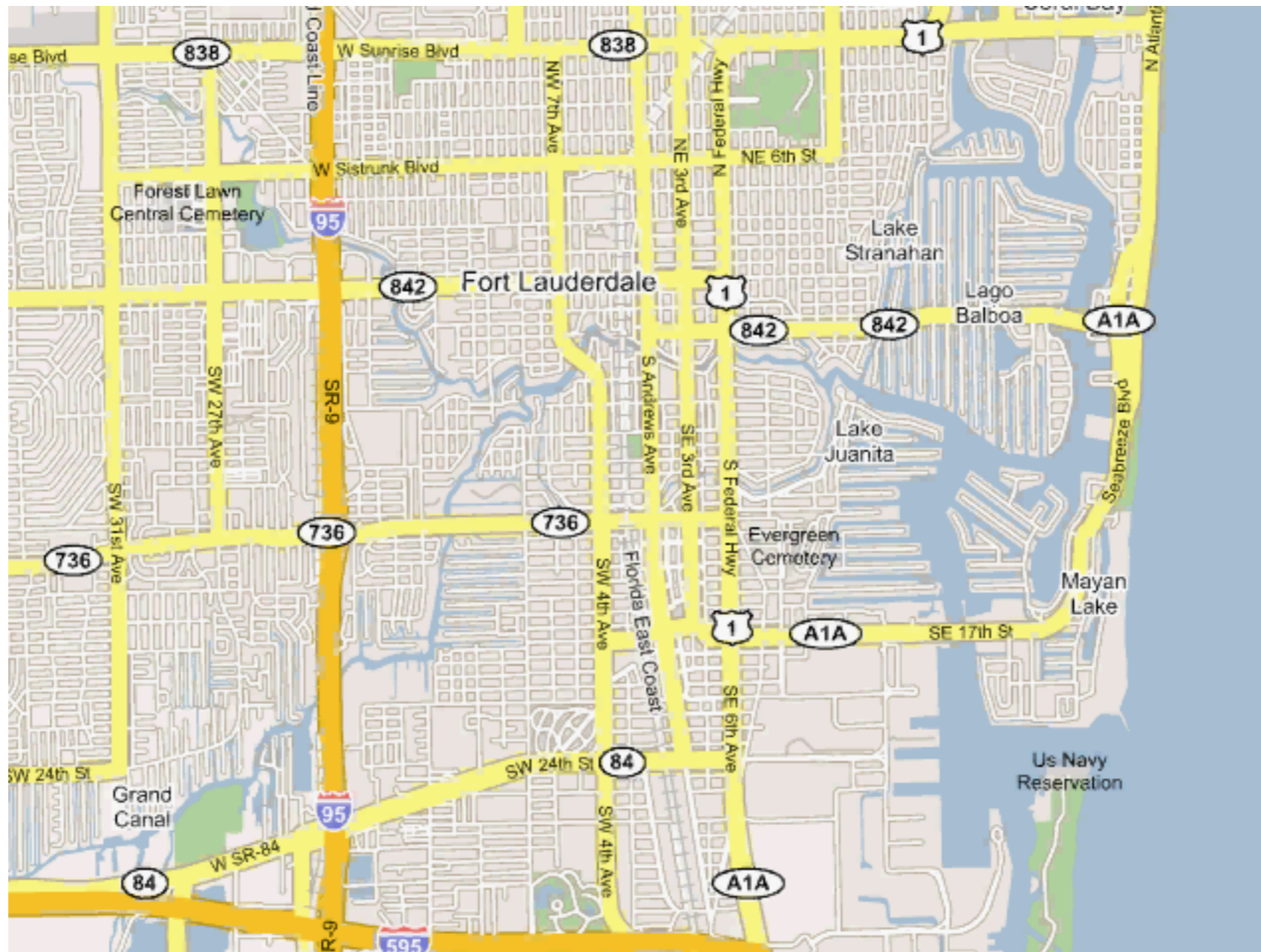
**ACUTE**

```
my $g = WWW::Gazetteer->new( 'heavensabove' );  
my $ftl = $g->find( 'Fort Lauderdale', 'US' )->[0];  
my( $longitude, $latitude ) = ( $ftl->{longitude},  
$ftl->{latitude} );
```

# Zoom



# Waave





# Yahoo

- Web search
- Image search
- Video search
- News search
- Local search
- Spell
- Related, Y!Q...

A photograph of bamboo branches with green leaves against a clear blue sky, serving as a background for the title.

# Recipe news

- I have a lot of recipes
- What distinctive service do they have?
- Aha, **news**!

# Recipe news

```
use Yahoo::Search AppId => 'AcmeNews';

my @links = Yahoo::Search->Results(
    News => 'recipes',
    Sort => 'date',
);

foreach my $link (@links) {
    print $link->Title . " (" .
        $link->SourceName . ") \n";
}
```

# Recipe news output

Authentically Sichuanese A pair of mainland chefs showcase the sophisticated cuisine of southwestern China (Bangkok Post)

Bowling-alley romance, British sitcom inspire diners (Pioneer Press)

SILVIA DODSON, ST. CLEMENT SHORTCAKE (Tampa Bay's 10)

‘Nasi briyani’ Johor style (@Metro KL)

Family Resource Fair scheduled for Saturday (Albany Democrat-Herald)

In the Raw: Cook Without Your Oven (ThirdAge)

I. Embarrass Yourself (ThirdAge)

[ Dining Index | North Bay | Metroactive Home | Archives ] (Metro Santa Cruz)

Zeslij swiathose swoja: Send your light upon us (Canyon Courier)

Eat to Your Heart's Content (Le Mars Daily Sentinel)



# Yahoo limits

- No API key needed
- 5,000 queries per day per IP

The word "Amazon" is written in a large, white, sans-serif font, centered horizontally. It is overlaid on a background image of green bamboo leaves and branches against a clear blue sky.

# Amazon

- E-Commerce Service
- Alexa Web Information Service
- Simple Queue Service



# E-Commerce Service

- Detailed product information on all Amazon.com products
- Access to Amazon.com product images
- All customer reviews associated with a product
- Extended search
- Remote shopping cart



# E-Commerce Service

- Amazon wish list search
- Precise response groups
- Multi-operation and batch interfaces
- Detailed error messages
- Built-in help functionality



# E-Commerce Service

- CPAN modules:
  - Net::Amazon



# Find me a recipe book

- So I have lots of recipes
- Some of them list an ISBN
- Some of them list a book (freeform)
- I'd like to make \$\$\$\$ from Amazon Associates

# Find me a cookbook

```
my $ua = Net::Amazon->new(token => $AMAZON, cache => $cache,
    strict => 1, max_pages => 1);
my $response = $amazon->search(
    blended => $source, # title or ISBN
);
if ($response->is_success()) {
    my @properties = grep { $_->Catalog eq 'Book' }
        $response->properties;
    my $first = $properties[0];
    return unless $first;
    my @nodes = $first->browse_nodes;
    my $nodes = "@nodes";
    return unless $nodes =~ /(cooking|cookbooks|food|gourmet)/i;
    my $asin = $first->Asin;
    return $asin;
}
```

# Cookbook in action

Astray Recipes: Chocolate & cherry loaf

http://www.astray.com/recipes/?show=Chocolat

## Chocolate & cherry loaf

Categories: [Brdmachine](#) [Breads3](#)  
Yield: 1 Loaf

- 1/3 c Light cream
- 2 tb Light cream
- 2 Eggs
- 3/4 ts Orange rind -- grated
- 1/4 c Sugar
- 1 ts Salt
- 3 c Bread flour
- 2 1/2 ts Yeast
- 3/4 c Semisweet chocolate, chopped
- 3/4 c Dried cherries (4 oz)
- 1/4 c Orange juice

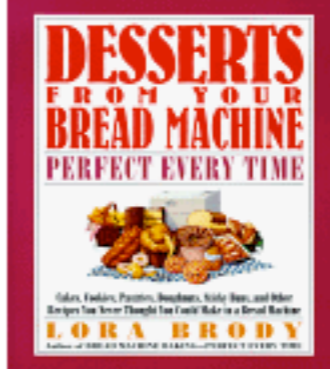
**PREPARED CHERRY MIXTURE** To prepare cherry mixture, soak cherries in orange juice overnight. Place all the ingredients except cherry mixture and chocolate in the machine, program for basic white bread and press start. After 33 minutes have elapsed, add the cherries with any remaining liquid and the chocolate.

Recipe from: Desserts from Your Bread Machine

[Bread machine recipe tips](#)  
[Chocolate cherry bread \(machine\) - large loaf](#)  
[Chocolate cherry bread \(machine\) - regular loaf](#)  
[Oatmeal-wheat germ bread](#)  
[Potato bread \(zojirushi\)](#)  
[Sugar free white bread](#)

**Related book**

Desserts from Your Bread Machine: Perfect Every Time : Cakes, Cookies, Pastries, Doughnuts, Sticky Buns, and Other Recipes You Never Thought You Cou ([amazon.com](#), [amazon.co.uk](#))



Done

Adblock

# Alexa Web Information Service

- URL information
- Browse category
- Web search
- Crawl meta data
- Web Map

# Alexa Web Information Service

- CPAN module:
  - none!
  - wait, that was accurate 12 hours ago
  - Net::Amazon::WIS
  - No, I can't find a good use for it on my website either...



# Simple Queue Service

- Basic queue operations
- Specifically designed for distribution applications
- Implemented to the highest standards for performance and reliability

# Simple Queue Service

- CPAN module:
  - `Net::Amazon::SimpleQueue`

**ACME**

# Simple Queue Service

```
my $sq = Net::Amazon::SimpleQueue->new( $sub_id );  
my $queue_id = $sq->create_queue(  
    name => $queue_name );  
$sq->enqueue(  
    name => $queue_name,  
    body => "here is the actual data" );  
my $entry = $sq->read( name => $queue_name );  
$sq->dequeue(  
    name => $queue_name,  
    entry_id => $entry->{id} );
```



# Queues

- If I had a cluster of distributed machines, I might use `Amazon::Simple::Queue`
- Donations welcome

# More Amazon modules

- CPAN modules:
  - Business::Associates
  - WWW::Amazon::Wishlist
  - WWW::Scraper::ISBN::AmazonUK\_Driver
  - WWW::Scraper::ISBN::AmazonUS\_Driver



# Amazon limits

- No more than one query per second



# Links

- <http://www.google.com/apis/>
- <http://developer.yahoo.net/>
- <http://www.amazon.com/gp/aws/landing.html>



# Summary

- For my website, I exploit enterprise web services in Perl
- Perl makes it easy



# Demo!

- Let's show it all off!