Smoke

Smoke

Ben Schwarz

Chapter 1

Introduction

Smoke is a DSL written in Ruby. Its primary goal is to be a very simple interface to query many online sources, chop them up into usable pieces and use them in ways that the source provider may have never imagined.

This document is aimed to serve as a guide for usage to get you started.

THE CONCEPT

The concept comes from using Yahoo Pipes to make web based mash ups: Get a list of tv shows for my torrent client, compile a recipe book or make tools to give me a list of albums that artists in my music library are about to be released.

Chapter 2

Diving straight in

Say you wanted to get your twitter stream, doesn't sound too hard? It isn't!

The first thing that you'll need to do is discover your source. Smoke supports YQL, XML, JSON and RSS / Atom.

```
require 'smoke'
Smoke.data(:twitter) do
  url "http://twitter.com/users/show.json?screen_name=benschwarz"
end
```

Now that the source has been defined you can use it like this:

```
Smoke.twitter.output
```

This will give you an array of hashes, each hash will be a tweet.

Right now you're probably sitting there smugly thinking to yourself "yeah, I could do that myself pretty easily."

Well, you smug bastard, there is some more we can do. Perhaps you'd like to open an interface to pull twitter feeds for multiple users? Maybe you'd like to strip out tweets that were directed at someone?

```
Smoke.data(:twitter) do
  prepare do
    url "http://twitter.com/users/show.json?screen_name=#{username}"
  end

  discard :text, /@\w/
end
```

Now that you've suggested that a username is required for the query. You need to send through a username.

```
Smoke.twitter.username(:benschwarz).output
```

The smoke DSL is pretty easy to pickup. All methods are well documented using rdoc, so exploring the API should be easy.

Cool stuff smoke does

OUTPUT FORMATS

Not only does Smoke query a web service and automagically translate Json or XML to an array of ruby hashes, it will also output in Json or Yaml.

```
Smoke.twitter.output(:yaml)
```

SMART REQUESTS

All requests ask the web service on the other end to gzip its data before it delivers it back to smoke. This means that less data is transferred - You can compress plain text pretty successfully.

Not all services support / implement gzip, but for those who do, its a clear win.