

JSON-P

JSON-P

abusing Javascript for remote data access

Licensed under Creative Commons Attribution 2.5 License



1
2
3
4

1
2
3
4

JSON what?

1
2
3
4

JSON what?

Why pee for?

1

JSON what?

2

Why pee for?

3

Generating JSON

4

1

JSON what?

2

Why pee for?

3

Generating JSON

4

Returning JSON-P

web connections

<http://connections.webdirections.org>



Web Connections 06

http://connections.webdirections.org/

WEB CONNECTIONS

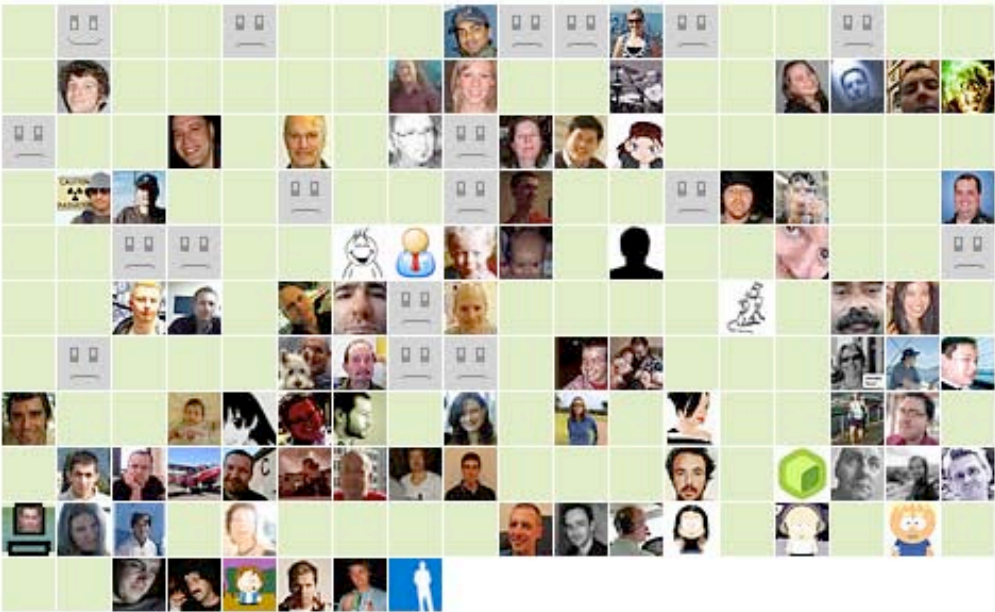
Home People Events Tags Login Sign Up

TAGS


EXPLORE ALL TAGS

Mac WE05 Web Directions Web Standards Group accessibility ajax CSS design html javascript likes beer melbourne photography php standards usability wd06 web standards wsg xhtml

FACES



MAP



POWERED BY Google Map data ©2006 MapData Sciences Pty Ltd, PSMA - [Terms of Use](#)

maps

tagging

feeds

social networking

JSON

HIJAX

resources

respond_to

flickr + technorati integration

restful authentication

```
>> Person.count
=> 188
>> Tag.count
=> 969
>> Tagging.count
=> 2412
>> Relationship.count
=> 3177
```



JSON what?

“Apart from being an
obviously cool named
technology”

— Jason Crane

J

S

O

N

JavaScript Object Notation

Describing data in Javascript

```
var person = new Object();  
person.nick_name = "Jez";  
person.name = "Jeremy Bogan";  
person.tags = ["SegPub", "coffee addict", "likes beer", "yellow"];  
person.latitude = -33.8974920438506;  
person.longitude = 151.177368164062;
```

Object Literal Syntax

```
var person = {  
  nick_name: "Jez",  
  name: "Jeremy Bogan",  
  tags: ["SegPub", "coffee addict", "likes beer", "yellow"],  
  latitude: -33.8974920438506,  
  longitude: 151.177368164062  
}
```

JSON is a **Subset** of Javascript's
Object Literal Syntax

JSON is a **Data** format

Object Literal Syntax

```
var person = {  
  nick_name: "Jez",  
  name: "Jeremy Bogan",  
  tags: ["SegPub", "coffee addict", "likes beer", "yellow"],  
  latitude: -33.8974920438506,  
  longitude: 151.177368164062  
}
```

JSON Syntax

```
var person = {  
  "nick_name": "Jez",  
  "name": "Jeremy Bogan",  
  "tags": ["SegPub", "coffee addict", "likes beer", "yellow"],  
  "latitude": -33.8974920438506,  
  "longitude": 151.177368164062  
}
```



```
eval("var data = (" + request.responseText + ")");
```



why pee for?

Client-side Scripting

Security Sandbox



mysite.com

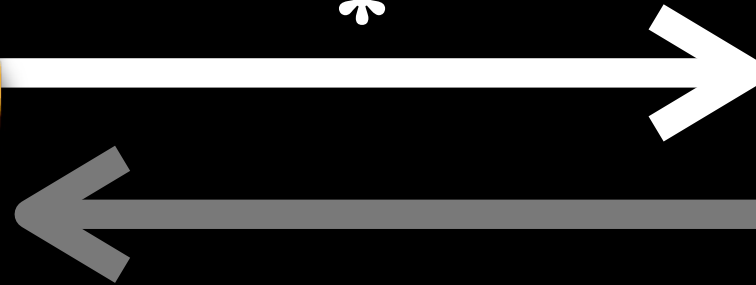


mysite.com



mysite.com

*



mysite.com



mysite.com



flickr.com



mysite.com

Restricted



flickr.com



mysite.com

Restricted



images
stylesheets
javascripts



flickr.com

TOOLMANTIM

[articles](#) [gigs](#) [photos](#) [who am i?](#) [subscribe](#)

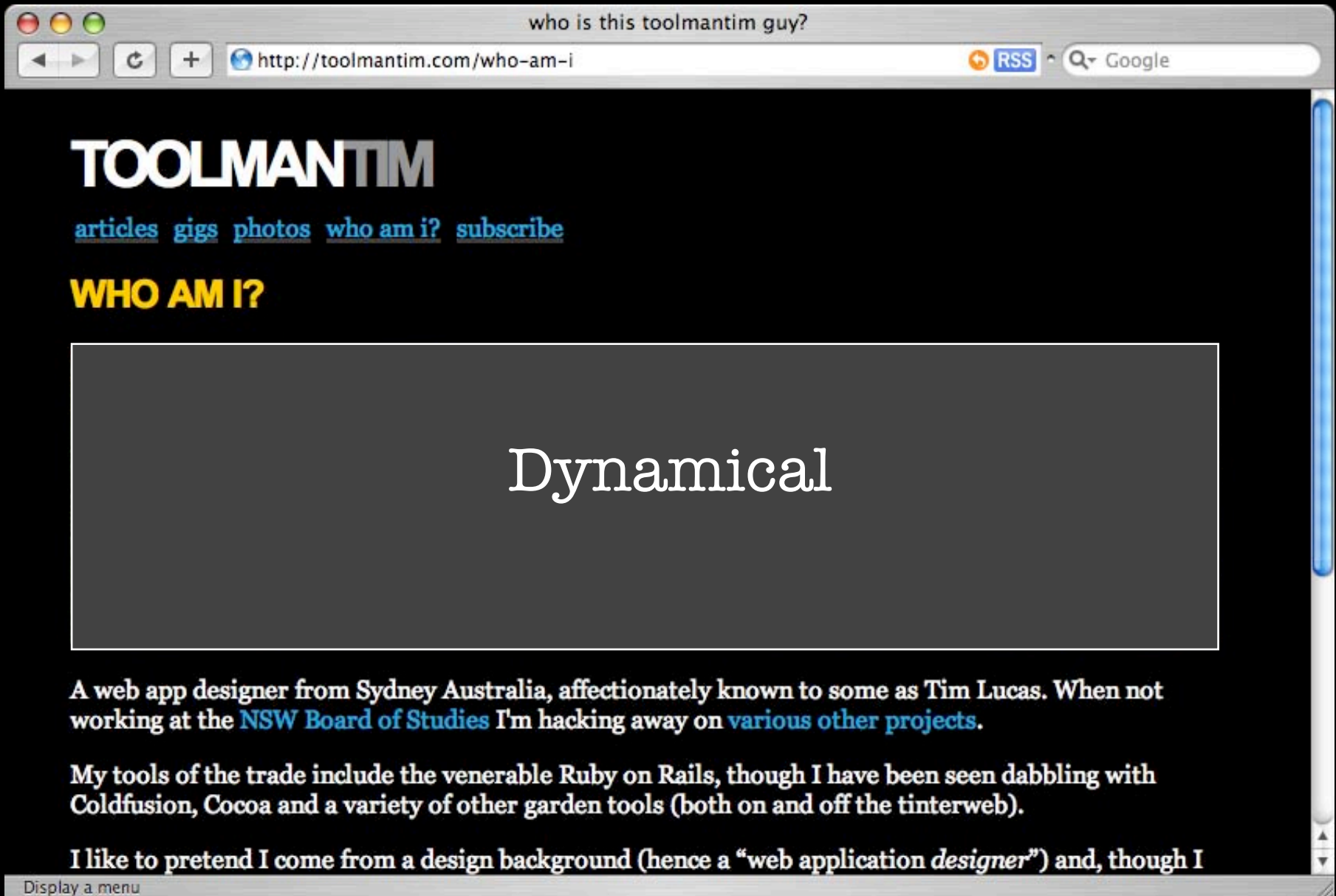
WHO AM I?



A web app designer from Sydney Australia, affectionately known to some as Tim Lucas. When not working at the [NSW Board of Studies](#) I'm hacking away on [various other projects](#).

My tools of the trade include the venerable Ruby on Rails, though I have been seen dabbling with Coldfusion, Cocoa and a variety of other garden tools (both on and off the tinterweb).

I like to pretend I come from a design background (hence a "web application *designer*") and, though I



```
var flickrRequest = null;
var flickrAPI = "http://api.flickr.com/services/rest/?method=";

window.addEventListener("load", initFlickrImages, false);

function initFlickrImages()
{
    flickrRequest = new XMLHttpRequest();
    req.onreadystatechange = processFlickrResponse;
    req.open("GET", flickrAPI + "flickr.photos.search&tags=timlucas&per_page=3", true);
}

function processFlickrResponse()
{
    if (flickrRequest.readyState == 4 && flickrRequest.status == 200)
    {
        // Process XML and insert images into the DOM
    }
}
```



```
var flickrRequest = null;
var flickrAPI = "http://api.flickr.com/services/rest/?method=flickr.photos.search";

window.addEventListener("load", initFlickrImages, false);

function initFlickrImages()
{
    flickrRequest = new XMLHttpRequest();
    req.onreadystatechange = processFlickrResponse;
    req.open("GET", flickrAPI + "flickr.photos.search&tags=timlucas&page=3", true);
}

function processFlickrResponse()
{
    if (flickrRequest.readyState == 4 && flickrRequest.status == 200)
    {
        // Process XML and insert images into the DOM
    }
}
```

images
stylesheets
javascripts

Abuse the script tag

```
<head>
  <script type="text/javascript" src="/javascripts/
flickrImages.js" />
  <script type="text/javascript" src="http://api.flickr.com/
services/rest/?
method=flickr.photos.search&tags=timlucas&per_page=3&format=json&
jsoncallback=processFlickrResponse" />
</head>
```



```
processFlickrResponse({  
    "stat": "ok",  
    "photos": [  
        // ...  
    ]  
});
```

flickrImages.js

```
window.addEventListener("load", insertFlickrImages, false);
```

```
var flickrImages = null;
```

```
function processFlickrResponse(rsp)
```

```
{
```

```
  if (rsp.stat !== "ok")
```

```
    return;
```

```
  flickrImages = rsp.photos;
```

```
}
```

```
function insertFlickrImages()
```

```
{
```

```
  if (flickrImages == null)
```

```
    return;
```

```
  // Insert flickr images via the DOM
```

```
}
```

JSON with Padding

介護新時代



第63回 全国老人福祉施設大会
平成18年 11月6日(月)～8日(水)
神戸市 神戸国際会議場

平成18年度 全国老人福祉施設研究会議
平成18年 10月16日(月)・17日(火)
札幌市 北海道立総合体育センター(きたえる)



全国老協

3 generating JSON

```
>> 1.to_json
=> "1"
>> "1".to_json
=> "\"1\""
>> [1].to_json
=> "[1]"
>> { :a => :hash }.to_json
=> "{\"a\": \"hash\"}"
>> Person.find(:first).to_json
=> "{\"incoming_relationships\": [{\"related_person_id\": 2,
\"person_id\": 3, \"relationships\": [\"met\"], \"id\": 1}],
\"nick_name\": \"themaninblue\", \"photo\": null, \"latitude\":
-42.0329743324414, \"tags\": [], \"longitude\": 146.42578125,
\"name\": \"Cameron Adams\", \"id\": 2, \"outgoing_relationships
\": []}"
```

ActiveRecord

```
class Person
  #..
  def to_json
    returning({}) do |json|
      %w(id name latitude longitude nick_name
outgoing_relationships incoming_relationships photo).each do |attr|
        json[attr.to_sym] = self.send(attr)
      end
      json[:tags] = tags.collect(&:name)
    end.to_json
  end
end
```



returning JSON-P

ActionController

```
class PeopleController < ApplicationController
  before_filter :find_person

  def show
    respond_to do |format|
      format.html
      format.js { render_json @person.to_json }
    end
  end

  protected
  def find_person
    @person ||= Person.find(params[:id])
  end
end
```

render_json

```
class ApplicationController < ActionController::Base
  protected
  def render_json(json, render_options={})
    callback, variable = params[:callback], params[:variable]
    response = begin
      if callback && variable
        "var #{variable} = #{json};\n#{callback}(#{variable});"
      elsif variable
        "var #{variable} = #{json};"
      elsif callback
        "#{callback}(#{json});"
      else
        json
      end
    end
    render({:content_type => :js, :text => response}.merge(render_options))
  end
end
```

Eating your own JSON

```
<%  
  menu_section (current_persons_profile? ? 'profile' : 'people')  
  depends_on_stylesheets 'profile'  
  depends_on_javascripts 'map'  
  depends_on_tag_data formatted_person_path(:id => @person, :format => :js, :variable => 'tagData')  
  @page_title = h(@person.name)  
%>
```

```
<script type="text/javascript" charset="utf-8" src="<%= formatted_person_path(:id =>  
@person, :format => :js, :variable => "tagData") %>></script>
```

```
<script src="/people/1.js?variable=tagData" type="text/javascript"></script>
```

json.org
wikipedia
sitepoint article
yahoo APIs

Acknowledgements



Nemo's great Uncle

<http://flickr.com/photos/maynard/109612815/>



Snapperwolf

<http://flickr.com/photos/snapperwolf/153030817/>

Thankyou!

Tim Lucas

t.lucas@toolmantim.com