DAN WEBB.NET

Unobtrusive Ajax With Rails

Dan Webb (dan@danwebb.net)

Overview

- A bit of history
- What is unobtrusive scripting?
- The UJS Plugin
- Casestudy
- ~~Ranting~~ Upcoming UJS Features

DAN WEBB.NET

The dark days of DHTML

DAN WEBB.NET

This is best viewed in 800 x 600 resolution. We strongly suggest that you have "**Internet Explorer 4.0**". Can be viewed with Internet Explorer 3.0 and Netscape Navigator 3.0.
If your computer is not 800 x 600 res.
You would not be able to view this site properly.

DAN WEBB.NET

This site is best viewed with 4.0 and up versions of Netscape Navigator/Communicator or Microsoft Internet Explorer. Some downloadable documents can only be viewed with Adobe® Acrobat® Reader. Click on any of the Icon's to download the latest software.

DAN WEBB.NET

Dan Webb

on-line CV

This site has been optimised for IE4+ and Netscape 4+ with 800x600 resolution.

1. Flash site

2. HTML site

Then web standards arrived

The Client-side Cake

What web standards did for us

- More maintainable
- More accessible
- Leaner pages
- Platform independent (print, mobile...)
- 'Future-proof' as well as backwards compatible

DAN WEBB.NET

JavaScript got a bad name

It deserved it

DAN WEBB.NET

Web 2.0

DAN WEBB.NET

Folksonomies

DAN WEBB.NET

Massive text boxes

DAN WEBB.NET

Social software

DAN WEBB.NET

and Ajax...

DAN WEBB.NET

JavaScript is trendy again!

Browser support is much better

DAN WEBB.NET

We learnt our lessons from the DHTML days

What did we learn?

DAN WEBB.NET

Unobtrusive DOM Scripting

It's an approach to browser UI design

It's about separating content and style from behaviour

DAN WEBB.NET

Behaviour: A new layer for the client-side cake

It's enhancing a working application

so it degrades gracefully when things don't work

It's not just about putting JavaScript in a different file

DAN WEBB.NET

It's not rocket science

DAN WEBB.NET

It's not the 'Rails Way'

An example:

```
link_to_remote
```

```
<%= link_to_remote 'View description',
                   :controller => 'product',
                   :action => 'desc',
                   :id => @product.id %>
```

```
<a href="#" onclick="new Ajax.Request('/product/
desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>
```

```
# better...

<a href="/product/desc/1" onclick="new
Ajax.Request(this.href, {asynchronous:true,
evalScripts:true}); return false;">View
description</a>
```

It's not possible to do that with link_to_remote

```
<% @products.each do |product| %>

<%= link_to_remote 'View description',
                   :controller => 'product',
                   :action => 'desc',
                   :id => @product.id %>

<% end %>
```

```
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>

<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true});
return false;">View description</a>
```

```
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
<a href="#" onclick="new Ajax.Request('/product/desc/1', {asynchronous:true, evalScripts:true}); return false;">View description</a>
```

DAN WEBB.NET

Getting started

The UJS Plugin

- A plugin to aid unobtrusive scripting with Rails
- Allows you to define behaviours via CSS selectors
- Keeps script in an external, cacheable files
- www.ujs4rails.com - check it out!

DAN WEBB.NET

Remote Links

```
<ul id="outline">
  <% @items.each do |item| %>
  <li><%= link_to :action => 'more',
                  :id => @item.id %></li>
  <% end %>
</ul>
```

```
<ul id="outline">

   <% @items.each do |item| %>

   <li><%= link_to :action => 'more',
                   :id => @item.id %></li>

   <% end %>

</ul>

<% apply_behaviour '#outline a:click',
   'new Ajax.Request(this.href); return false;' %>
```

```
<ul id="outline">

   <% @items.each do |item| %>

   <li><%= link_to :action => 'more',
                   :id => @item.id %></li>

   <% end %>

</ul>

<% apply_behaviour '#outline a', make_remote_link %>
```

What about links with side-effects?

DAN WEBB.NET

Links should never have side effects

Use a button

```
<%= button_to 'Remove from basket',
              :method => :delete %>
```

```
<form class="button-to" action="/basket/2" method="post">
  <input type="hidden" name="_method" value="delete" />
  <input type="submit" value="Remove from basket" />
</form>
```

Then attach the behaviour

```
<%= button_to 'Remove from basket',
              :method => :delete %>

<% apply_behaviour 'form.button-to',
                   make_remote_form %>
```

DAN WEBB.NET

Hijacking forms

```
<%= form_tag :url => entries_url, :id => 'comment' %>
  <%= text_field :name %>
  <%= text_field :email %>
  <%= text_area :comment %>
  <%= submit_tag 'Post comment' %>
<%= end_form_tag %>
```

```
<%= form_tag :url => entries_url, :id => 'comment' %>
  <%= text_field :name %>
  <%= text_field :email %>
  <%= text_area :comment %>
  <%= submit_tag 'Post comment' %>
<%= end_form_tag %>


<% apply_behaviour '#comment:submit',
   'new Ajax.Request(this.action, { parameters :
   Form.serialize(this)}); return false;' %>
```

```
<%= form_tag :url => entries_url, :id => 'comment' %>
  <%= text_field :name %>
  <%= text_field :email %>
  <%= text_area :comment %>
  <%= submit_tag 'Post comment' %>
<%= end_form_tag %>

<% apply_behaviour '#comment', make_remote_form %>
```

DAN WEBB.NET

A Case Study

Sneakr.com: A Web 2.0, Ajax Trainer Shop

routes.rb

```
map.resource :products
map.resource :basket, :controller => 'basket',
                      :collection => {:clear => :post}
```

The Product Controller

```
class ProductController < ApplicationController

  def index # show all products
    @products = Product.find :all
  end

  def show # show the details of a product
    @product = Product.find params[:id]
  end

end
```

index.rhtml

```
<div id="catalogue">
  <%= render :partial => 'products' %>
</div>

<div id="basket">
  <%= render :partial => 'basket' %>
</div>
```

_products.rhtml

```
<ul>

  <% @products.each do |product| %>

    <li id="<%= product.id %>_prod">
    <%= link_to product.name, product_url(product) %>
    <%= product.description %>
    </li>

  <% end %>

</ul>
```

show.rhtml

```
<h1><%= @product.name %></h1>

<p><%= image_tag @product.photo.public_filename %></p>

<p><%= @product.description %></p>

<p><%= button_to 'Add To Basket', basket_url
(@product), :method => :put %></p>
```

The Basket Controller

```
class BasketController < ApplicationController

  def update

    @product = Product.find params[:id]
    @basket << @product

    redirect_to products_url

  end

end
```

DAN WEBB.NET

We're done!

DAN WEBB.NET

now to add the Ajax

```
<% apply_behaviours do

    on '#catalogue li',
       make_draggable(:revert => true)

    on '#basket', make_drop_receiving(
      :url => basket_url,
      :with => "'_method=put&id=' +
        encodeURIComponent(element.id)"

    )

end %>
```

```
<% apply_behaviours do

    on '#catalogue li',
       make_draggable(:revert => true)

    on '#basket', make_drop_receiving(
      :url => basket_url,
      :with => "'_method=put&id=' +
        encodeURIComponent(element.id)"

    )

end %>
```

So how do we deal with this on the server-side?

DAN WEBB.NET

respond_to

```
class BasketController < ApplicationController

  def update

    @product = Product.find params[:id]
    @basket << @product

    redirect_to products_url

  end

end
```

```
class BasketController < ApplicationController

  def update

    @product = Product.find params[:id]
    @basket << @product

    respond_to do |type|
      type.html { redirect_to products_url }
      type.js
    end

  end

end
```

update.rjs

```
page.replace_html 'basket',
                  :partial => 'basket'
```

Take a look for yourself...

http://www.danwebb.net/railsconf2006/ujs_shopping.zip

We have no control over the environment our JavaScript runs in

DAN WEBB.NET

Code defensively

The path to enlightenment

- Write a working application using semantic HTML
- Style it with CSS
- Write JavaScript that 'hijacks' the page elements to enhance the UI
- Learn JavaScript and DOM Scripting

Further Reading

- The JavaScript articles on A List Apart (alistapart.com)
- Unobtrusive Scripting by Christian Heilmann (onlinetools.org)
- Jeremy Keith's presentations, book and articles (domscripting.com)
- Google it!

Upcoming in UJS

- ★ Improved testing (custom assertions)
- ★ Improved debugging
- ★ More behaviour helpers
- ★ More tutorials on ujs4rails.com

And finally...

```
<%= apply_behaviour @products, make_draggable %>
```

```
<% div_for @product, :behavior => make_draggable do %>

<h2><%= @product.name %></h2>
<p><%= @product.description %></p>

<% end %>
```

DAN WEBB.NET

Questions?