Ajax applications with perl backend - how to?

Asked 13 years, 6 months ago Modified 10 years, 6 months ago Viewed 7k times

15

There are already questions about the Perl+AJAX, like here, here or here and several others. They're more than 2 years old and I was hoping for some new stuff.

The questions are:

- What is the most preferred method today making AJAX apps with a Perl backend?
- Are there some *active* and commonly used Perl modules that help build AJAX based applications?

Something, for the usual workflow:

```
if clicked this button (or changed this field.. etc),
POST these data to the server,
read the JSON answer,
and update this/these DIV(s) in a DOM... etc.
```

This question is can be classified as vague, but I'm **really lost** and need help with this: what is the most common way making AJAX apps in the Perl world, TODAY.

Looking for a helper module that help me build the browser-side javascript.

I found these:

- https://metacpan.org/pod/OpenThought
- https://metacpan.org/pod/HTML::AjaxTags
- https://metacpan.org/pod/CGI::Ajax (but this is CGI based and IMHO not the best for Plack world)

These modules *have not been updated* for several years. Are they stable and **in use**? Or are they deprecated and is there some better way? (for the modern Perl technologies - like Plack).

UPDATE

As I read answers, I'm thinking that the main problem is probably in my English. I don't know how to express myself right.

I know Perl. Maybe I'm not an expert, but I wrote several thousand lines of code. I know Dancer, and already write some apps in Mojo...::Lite. Know JSON{::XS} and I know how AJAX works.

Now (for some reason) I prefer using Mason2, with Mason::Plugin::RouterSimple and several other CPAN modules and Moose. Catalyst, Jifty are too big for my needs.

Back to the question:

My favorite JS framework is jQuery, I'm using it in several projects for modal windows, or accordions, tabs etc.

BUT

My main problem is exactly in Sismetic's answer. I don't want to write JavaScript. Don't like it. (Don't know it very well, and hate every language where I must write something like: `var arr = new Array();` instead of `my @arr`)

So, looking for a solution, how I can **minimize** (or in the ideal world - totally eliminate) the need of writing the JavaScript code. Don't want write into my templates

```
$('.clickableButton').click(function(e) {
      .... etc... etc..
)}
```

but something like:

```
 $ajax->make_button( -onchange=>$url, -updatedom=>'#thisdiv",
some_special_button_description_in_perl );
 $tohead .= $ajax->gen_libs();
 $tohtml .= $ajax->gen_html();
 $jsdocready .= $ajax->gen_jsinitcode();
```

An in my templates will output only `$tohead` in the head part (so include jQuery), `$tohtml` will come into body, and `$jsdocready` will come into the end of body as JavaScript init code.

Offcourse, the above is an very stupid example, but hopefully shows what I mean. Simply: The ideal solution was (probably does not exists) is totally eliminate the need writing JavaScript, only Perl code what will generate the needed JS.

Therefore I mentioned the above modules, especially https://metacpan.org/pod/OpenThought, because these really help minimize writing JavaScript. The problem is - these have not updated for 2 years. ;(And unfortunately - I don't know any others.

javascript perl

Share

Improve this question

Follow

edited Jun 20, 2020 at 9:12

Community Bot

1 ● 1

asked May 30, 2011 at 12:32

kobame

5,856 ● 3 ● 34 ● 63

I can't help you with the modules, but your question brought up some responses that helped me - I've edited your question to correct your English a bit, maybe that helps you for future questions :) – Konerak Jun 10, 2011 at 13:12

Thanx. ;) My perl is definitely better than my english... – kobame Jun 10, 2011 at 20:13

6 Answers

Sorted by: Highest score (default)

23

+25

The reason you're not getting answers isn't just the vagueness of the question. The problem space is *very* wide and has a many angles of attack.

Let's clarify that the "x" in Ajax shouldn't be taken to mean XML anymore. JSON is obviously more natural and doesn't suffer from nearly as many problems so all my advice will point toward it.

What hobbs said already is right on. You do not want to mess with the server-side code much at all but adopt a framework. This is not because dealing with Ajax is hard in Perl; it's trivial. It's because the problem space gets messy quickly and you'll end up repeating your code in endless minor variations. So–

Perl/Server-side

Any of these will make you happy eventually. They all have a learning curve. There are other options but these are the Best™.

- Catalyst (Catalyst::View::JSON)
- Dancer (Dancer::Serializer::JSON)
- Mojolicious (prefer Mojolicious for HTML5/Websockets)

Deployment SHOULD be Plack/PSGI based.

Take the time to really learn the core of Perl's Ajax handling: JSON(::XS) so you know what the views in the various frameworks do under the covers.

JavaScript/Client-side

This is essentially an embarrassment of riches at this point.

- jQuery
  - Many Perl hackers like this kit; it seems to hit the same sweet spot Perl does. I adore jQuery.
- Dojo
  - I'm not a fan — they had the worst documentation possible in early versions and broke compatibility while deleting what little docs used to exist — but it's well-liked in its current version.
- MochiKit
- MooTools
- YUI
- ExtJS
  - This, now distant, fork from YUI is the 800lb gorilla of client-side JS. I personally dislike it because of its lack of graceful degradation, et cetera, but it is highly regarded and very sharp looking out of the box.

I personally dislike and can't recommend prototype and though I've never used it I also chose not to put script.aculo.us on the list.

There are *plenty* of other amazing specialty kits out there too; e.g., Modernizr. When you are looking into JS, consider how important standards compliance and forward-looking features like CSS3, HTML5, extended event handling like multi-touch are to what you'll do. Good luck and have fun.

Update: Possibly of further interest

- Jemplate – templating in JavaScript
  - Catalyst::View::Jemplate
- Jifty – YAWF

Share

Improve this answer

Follow

edited May 23, 2017 at 12:18

Community Bot
1 ● 1

answered Jun 1, 2011 at 18:53

Ashley
4,335 ● 2 ● 20 ● 28

1 +1 for taking the time to tackle this exhaustive write-up of the current practices – daxim Jun 3, 2011 at 15:07

1 Very nice and as daxim told - exhaustive write up of common perl web frameworks and JS frameworks. Thanx for the answer. My problem is somewhere between - how to avoid writing JS and generate it by perl. – kobame Jun 3, 2011 at 22:08

1 @kobame, yeah, and I think I should have included some things like Jemplate and other templating approaches. I do agree with @hobbs here again that writing Perl that writes JavaScript is generally just going to make you sad. Consider too that you want your client-side to be agnostic enough that you could use a Ruby/Python/PHP/Java service instead of Perl to get the same data without any alteration. The cleaner the separation, the more flexibility and maintainability you end up with. – Ashley Jun 3, 2011 at 22:14

1 Hm.. probably youre right. So, probably need take two stage approach. In the 1st stage generate the final template, so some standalone script what will generates the final templates with JS, and after it the main app is simple Dancer, Mojo or Mason JSON replies. But i don;t give up yet and trying find a solution how to not use JS - so a solution for the 1st stage. :) Thanx a lot for your interest and help. – kobame Jun 3, 2011 at 22:31

1 WOW! Jemplate looks at the 1st look iterestant. Going to learn. :) – kobame Jun 3, 2011 at 22:35

8

Looking for "ajax" isn't really what you need. Just use a web framework of your choice that has good facilities for serialization, working with Accept headers, etc. For example Catalyst and Catalyst::Action::REST, or Dancer. Don't write Perl that writes Javascript, it will only make you sad.

Share Improve this answer Follow

answered Jun 1, 2011 at 16:45
hobbs
239k ● 20 ● 222 ● 300

Can you show me, how for example in Dancer you will make an ajax page? Probably will write all needed JS code into templates by hand (and this is exactly what i want avoid), and after will use `get,post '/ajax'` route to return the wanted JSON reply. But this is a terrible solution. Writing page content by hand is what i want avoid. ANyway, thanx for the reply, because helping me express for what i looking. :) :) – kobame Jun 3, 2011 at 22:13

4

I use CGI::Application as my base framework and CGI::Application::Plugin::JSON to return JSON data to jQuery.

Share Improve this answer Follow

answered May 31, 2011 at 15:02
Raoul
3,889 ● 3 ● 25 ● 30

1

If you want to generate HTML code with a Perl module, I would recommend CGI.pm:

```
...
use strict;
use warnings;
#CGI is shipped along perl, I think
use CGI;

my $CGI = CGI->new();

my $return_string = '';
#From CGI documentation on CPAN
#http://search.cpan.org/~markstos/CGI.pm-3.55/lib/CGI.pm
$return_string .= $CGI->header;
$return_string .= $CGI->start_html('hello world');
$return_string .= $CGI->h1('hello world');
$return_string .= $CGI->button(-name    => 'button_name',
                               -value   => 'Click Me!',
                               #Javascript event if needed
                               -onClick => "do_something()"
                              );
$return_string .= $CGI->end_html;
print $return_string;
```

Or(as I dont like the latter method) you could just write it on perl(generating it yourself manually):

```
use strict;
use warnings;

#Needed header, otherwise will return error
print "Content-type: text/html\n\n";

#return random number from 0 to 100
my $random_number = int(rand(101));

my $HTML_generated_string = qq|
<html>
 <head>
  <title>HTML generated manually with perl</title>
 </head>
 <body>
  <h1>Hello world</h1>
  Bla bla bla Heres the random number $random_number
 </body>
</html>
|;

print $HTML_generated_string;
```

Other than that, I dont know any extra modules to do it. I normally do it by hand, or write a template(with CGI::Application).

Share   edited Jun 8, 2011 at 14:02   answered Jun 3, 2011 at 18:22

Improve this answer

Follow

Sismetic
327 ● 5 ● 16

Thanx for the answer. Veri helpful, because i can exactly show - what i don't want. That is exactly that what i want avoid. Writing 50 lines of javascript code and 5 lines of perl. I rather want wrote 50 lines of perl and 0 lines of javascript. :) – kobame Jun 3, 2011 at 22:04

Man, I dont think you can use AJAX without JavaScript, besides it's not 50 lines, they're like 6, each of around 10 characters, the rest are comments. As @Eric Johnson says, you should rethink about using JavaScript, JQuery makes its implementation really useful and easy. Btw, every line that starts with // in JS is a comment. And <!-- on HTML, so just strip them, and you will see the code is really short. – Sismetic Jun 7, 2011 at 17:12

Ofc, in the browser you need javascript. My questions is how do not writing javascript, but got a generated one. Think about like: html(body(h1("This is a title"), p(strong({id="idstr", class="strcls"}, "String text")))) - this is valid perl code, and (by using a right module) WILL generate valid HTML or XHTML. And I don't needed write nothing in HTML. I hope this clear a little bit what i mean. – kobame Jun 8, 2011 at 12:16

Oh, ok, you can write that with the CGI.pm module – Sismetic Jun 8, 2011 at 13:48 ✎

One question, do you know Javascript? Seeing the link you gave @Eric Johnson, that framework is for people who do not know Javascript, perhaps this is the case. So you want an HTML and Javascript generator so you can use the AJAX feature, without knowing any Javascript? – Sismetic Jun 8, 2011 at 14:24

0

I don't think the thing you are asking for exists. You can't write AJAX apps without using JavaScript. If someone invented a way to do JavaScript without JavaScript it would be a major project and not one person's weekend CPAN module.

I suggest rethinking your view of JavaScript. JavaScript is an excellent language and JQuery is great as well. There is a lot of bad JavaScript code out there, but there is good JavaScript code as well. Just like there are good and bad practices in the Perl community. I'd suggest you keep an open mind and take another look at JavaScript and make sure you really know the foundations well. Sometimes my frustration with a language just means I haven't learned it very well.

There are some very cool things happening in the JavaScript world. I admit its chaotic and decentralized and there is a lot of different stuff to know about and thats exhausting but its also exciting that so much is happening.

You probably already know about these resources, but just in case: FireBug, Mozilla Developer Network JavaScript docs, and JavaScript Garden

Share Improve this answer Follow

answered Jun 7, 2011 at 1:59

Eric Johnson
17.9k ● 10 ● 54 ● 60

See for example jQuery Mobile. You can make mobile applications, (with AJAX calls, triggers, sliders, popups and so on) **WITHOUT** writing **any** javascript - only in PURE HTML. Because someone already wrote an framework (jQuery-mobile), what can do right things. So, here already **are** frameworks what can simplify creating AJAX applications. For example, ajaxtags.sourceforge.net/index.html is really powerful. My only concern is than its perl-port is probably abandoned. And the main point of my question is - are here not abandoned modules? – kobame Jun 8, 2011 at 12:10

-1

See **Dancer::Plugin::Ajax** on cpan
http://metacpan.org/pod/Dancer::Plugin::Ajax

Share

Improve this answer

Follow

edited Jun 12, 2014 at 8:39
szabgab
6,262 ● 11 ● 54 ● 64

answered Jun 1, 2014 at 18:05
ft4bgs54f156
1

Please post all important information in the answer itself, do not just post a link. – Baum mit Augen ♦ Jun 1, 2014 at 18:27