

My experiences with Catalyst



My Perspective

Trainee/self-taught web developer of 12 years

Used to 'hand roll' a lot of stuff or use Mason + a few other CPAN modules ... pretty much frame-work free.

Why Catalyst?

I did not really know what a framework was, but I was looking for one!...

.. I was looking for a flexible CMS type thing we could all start using at work to meet our customers requirements.. I read a little about Catalyst, and was tempted to give it a go...



First Impressions

I was not blown away straight off ... having read only a bit about it I could not really see the point of it all... I can do all of this in Mason??

At the MKLUG, I asked a few questions and got a very, VERY positive and encouraging response... so I continued, with faith I was not going down the wrong path...

Facts like, BBC iPlayer is using it, calmed any worries I had about wasting time on this research .. obviously, production use of this thing called Catalyst is not a problem!



The best place to start..

For a good grounding in the different areas of Catalyst, I would suggest running through the **Catalyst::Manual::Tutorial**

.. next ...

DO NOT then go ahead and start your first production application, you will make important decisions that with a bit more experience you will probably make differently ... believe me, I KNOW!

DO read the **The Definitive Guide to Catalyst!**

This book is full of handy Perl tips (not just Catalyst). Most importantly though, it demonstrates good Catalyst practise...

.. now, with the knowledge from the book, do as much production stuff as you want ;)



What has Catalyst given me?

Structure – I do not have to think so hard about my files and how to lay them out. No more making up and sticking to rules. It's just, the way you do it.

Organisation – By its nature, your Catalyst apps will be tidy and organised, still requiring a bit of thought but the MVC layout goes a long way to sorting your code out.

Less Stress – Since I have been using Catalyst and the associated modules, I have a lot more confidence in the rigidity of my work.

Modern techniques – AJAX, Web services and more are easy with Catalyst.

Testing – Proper testing is a great way to keep track of your application and how updates have effected it ... slightly painful at first, but worth the extra typing.

Education – Using Catalyst has opened my eyes to world of OO Perl. Now there is so more much to learn.



What are some cool advantages?

Catalyst on its own is very light weight (if you just want the framework) but typically you will want to integrate with other CPAN modules... here are some I really liked using.

Catalyst::Plugin::Authentication - Session

Everyone wants an easy to use, flexible and quick to configure authentication system and session... here it is!

HTML::FormFu

Configuration of forms for output and validation...

- Groupings
- TT
- Model Integration
- Date Input types
- Validation (Call backs, this or this, etc. etc.)

Catalyst::Plugin::RequireSSL

Might be a bit old now, but I found it to be exactly what I wanted... now I don't even think about HTTP/S



.. and some more cool advantages?

DBIx::Class

Now this is whole different beast, and nothing to do with Catalyst, but I was introduced to it via Catalyst and thought it should be mentioned.

This is scary at first and frustrating... **I just want to do a freekin query!** ... but once over the first few hurdles I am using it all the time and LOVE all the advantages, although I know I have a lot more to understand yet.

One of first things I thought was cool ... I had done a pretty complex page for a client, who then said “love it, perfect ... but can we have it across multiple pages?”

I thought, nightmare!... all that lovely code messed up but with DBIx::Class and Data::Page, my problem was solved in about 20 characters.

Another cool thing is after spending time putting my schema into .. err.. Schema... I found I could do some cool things..

```
$customer->update( { name => 'bill' } )
```

```
$customer->add_groups(@groups)
```

```
$customer->search_related('groups', { name => { like => 'a' } } )->count
```



.. if you're interested...

```
## MODEL ## DBIX::Class::Schema

__PACKAGE__->load_components("InflateColumn::DateTime", "Core");
__PACKAGE__->table("customers.ta_customers");
__PACKAGE__->add_columns(
    "customer_id",
    {
        data_type      => "integer",
        default_value  => undef,
        is_nullable    => 0,
        is_numeric     => 1,
        size           => undef,
        sequence       => 'customers.sq_customer_id',
        is_auto_increment => 1,
        auto_nextval   => 1,
    },
    "date_registered",
    {
        data_type => "timestamp",
    },
    .. etc, etc...
)
```

CONTROLLER

```
# .. setup some vars (perhaps from the QS)
my $letter_list = [A .. G];
my $page = ( $c->params->{page} ? $c->params->{page} : 1 );

# ... running a not standard sql search (returning ResultSet)....
$c->stash->{foundcustomers} = $c->model("DB::Customers")->search(
    {
        custtype      => ['theatre', 'venue'],
        website_suppress => 0,
        'upper( substring(addr_town, 1, 1) )' => { in => $letter_list }
    },
    {
        order_by => ['company_name'],
        rows      => 40,
        page      => $page
    }
);
```

VIEW ## TemplateToolkit

```
[% customers_pager = foundcustomers->pager %]

# .. want to display how many pages got returned??..
[% customers_pager.total_entries -%] Results:

# .. use TT to display all available pages...
[% FOREACH pageno = [customers_pager.first_page .. customers_pager.last_page] -%]
    <a href="[% c.request.uri_with( page => pageno ) %]">
        Page: [% pageno %]
    </a>
[% END -%]

# .. display the customers details (the date in format we ask for)...
[% WHILE (cust = foundcustomers.next) %]
    [% cust.company_name -%]
    Registered: [% cust.date_registered.dmy('/') -%]
[% END -%]
```



There's SO much more...

I have only used Catalyst for one large (ish) project and therefore have only dipped my toe in the water, but I know there is loads more to learn and to do with it.

If you are in a similar position to me I can not shout loud enough about how great **Catalyst** is and how it will improve your life

