Lazy Programmers Guide to html tables in Dancer2

Jason Lewis

September 21, 2016

Contents

About me

Jason Lewis
  Owner of a wholesale distribution business in Australia
  Jack of all trades...

contact

- email: jason@dickson.st
- IRC: k-man
- Twitter: @jasonblewis

Why am I doing this?

Business reporting done in Crystal Reports

- Replace Crystal Reports with something less proprietary.
- make reports accessable on the web

Crystal Reports

Existing system was built in Crystal Reports

- Pros:

    - very quick to build reports
    - quite easy to customise and make the reports just so

- Cons:

    - proprietary
    - distribution of reports difficult
    - not easy to put the reports on the web
    - win32 only

Why Dancer2?

I looked around
    Dancer has an awesome community.

Hand coded html table in Template

Route handler

```
get ’/demo01’ => sub {

  my $sth = database->prepare(
’select * from invoices’,
      );
  $sth->execute() or die $sth->errstr;

  my $fields = $sth->{NAME};
  my $invoices = $sth->fetchall_arrayref({});

  template ’demo01’,
    { title => ’demo01’,
      fields => $fields,
      invoices => $invoices,
    };
};
```

View

```
<table width="80%" id="example-table">
...
```

```
    <tbody>
[% FOREACH invoice IN invoices %]
    <tr>
[% FOREACH field IN fields %]
    <td>[% invoice.$field %]</td>
[% END %]
    </tr>
[% END  %]
    </tbody>
</table>
```

Demo 1

`http://localhost:5000/demo01`

Pros and Cons

- pros:
  - simple to write
- Cons:
  - resulting web page very static
  - tables with many rows cumbersome

add DataTables to the mix

hand coded table + DataTables

What is DataTables?

- JavaScript framework for manipulating html tables
- adds about about 136k to your page (depending on components)
- `https://datatables.net/`

Include the CSS and JavaScript for DataTables

in the main.tt layout add:

```
<!-- DataTables -->
<link rel="stylesheet"
      href="[% request.uri_base %]/css/jquery.dataTables.min.css">
<script type="text/javascript"
src="javascripts/jquery.dataTables.min.js">
</script>
```

call the DataTable enabler after document ready

add JavaScript DataTable call to the view:

```
<script type="text/javascript">
 $(document).ready(function(){
     $(’#example-datatable’).DataTable();
 });
</script>
```

#example-datatable is the CSS id of the table you want to make fancy

add #example CSS id to table

```
<table width="80%">
    <thead>
<tr>
    [% FOREACH field IN fields %]
<th>[% field %]</th>
    [% END  %]
</tr>
    </thead>
    ...
</table>
```

add #example CSS id to table

```
<table width="80%" id="example-datatable">
    <thead>
<tr>
    [% FOREACH field IN fields %]
```

```
<th>[% field %]</th>
    [% END  %]
</tr>
    </thead>
    ...
</table>
```

Demo 02

http://localhost:5000/demo02

Pros and Cons of hand coded html tables with DataTables

- pros:
  - very configurabe, you can generate the HTML table just how you like it.
  - easy to give rows and columns custom css IDs and classes
- cons:
  - not very reusable, you have to hand code each each report
  - changes to data structure may require updates to the view

using JSON with DataTables

- build the HTML table headings in javascript
- pass in a URL that returns JSON to DataTables
- DataTables retrieves the data and fills the table.

build the table header in JavaScript

insert the #tableDiv

```
$( document ).ready( function( $ ) {
    $.ajax({
"url": ’[% json_data_url %]’,
"success": function(json) {
    var tableHeaders = ’’;
    $.each(json.columns, function(i, val){
```

```
tableHeaders += "<th>" + val.data + "</th>";
    });

    $("#tableDiv").html(
      '<table id="displayTable"      \
class="display compact"      \
cellspacing="0"><thead><tr>'
      + tableHeaders + '</tr></thead></table>');
    $('#displayTable').DataTable(json);
},
"dataType": "json"
    });
});
```

build a json route

Include the columns you want to render and the results from the query

```
get '/api/demo03' => sub {
# return query as JSON
  my $sth = database->prepare(
'select * from invoices',
      );
  $sth->execute() or die $sth->errstr;

  my $invoices = $sth->fetchall_arrayref({});

  send_as JSON => { columns => [
    { data => 'InvoiceId'},
    { data => 'InvoiceDate'},
    { data => 'CustomerId' },
    { data => 'BillingAddress'}
      ],
    data => $invoices,
  };
};
```

Demo 03

`http://localhost:5000/demo03`

Pros and Cons

- pros
  - very easy to reuse code
  - page response feels faster for the user
- cons
  - you need an API route to return the data
  - more difficult to customise your resulting html table
  - adding custom CSS IDs to rows requires writing javascript

styling the table

DataTables comes with some predefined CSS for example, classes for left and right alignment:

- dt-left
- dt-right

css classes

Add CSS classes to columns

```
...
send_as JSON => { columns => [
  { className => 'dt-right', data => 'InvoiceId',      },
  { className => 'dt-left',  data => 'InvoiceDate',    },
  { className => 'dt-right', data => 'CustomerId',     },
  { className => 'dt-left',  data => 'BillingAddress',
       title => 'Billing Address'}
    ],
  data => $invoices,
};
```

Demo 04

http://localhost:5000/demo04

Other columns properties

columns has many other properties that can be useful

- name: Descriptive name for the column
- title: Column title
- visible: enable or disable display of this column

problem with this approach

formatting creeping into the data view

table export options

Users are never satisfied

- Can I export it to Excel?
- DataTables makes that easy
- Buttons component.

CSS and JavaScript for DataTables Buttons

Install pdfmake

```
cd MyApp/public
bower install pdfmake
```

add the DataTables Buttons css

```
<link rel="stylesheet"
  type="text/css"
  href="https://cdn.datatables.net/buttons/1.1.1/css/buttons.dataTables.min.css">
```

add the JavaScript

```
<script src="/javascripts/buttons.html5.min.js"></script>
<script src="/javascripts/buttons.print.min.js"></script>
<script src='/bower_components/pdfmake/build/pdfmake.min.js'></script>
<script src='/bower_components/pdfmake/build/vfs_fonts.js'></script>
```

add buttons option to our javascript

Add this to our JavaScript from before

```
json.dom = 'Blfrtip'; // customise the table
json.buttons = ['copy',
'csv',
'excel',
{ extend: 'pdfHtml5',
  text: 'PDF',
  orientation: 'landscape',
  pageSize: 'A4',
  download: 'download',
  filename: '*',
  extension: 'pdf'
},
'print'];
```

Demo 05

`http://localhost:5000/demo05`

As yet unresolved challenges

- move formatting and options out of main route
- formatting dates like '2009-01-01 00:00:00'
- rounding floats to fixed decimal places

thanks for listening