Export PDF

Adding export=pdf to your HTTP request will cause QuickDRY to try to render your page to a PDF. There are a few things that must be done to enable PDF support for your page.

The first thing you need a master page that will set up the PDF’s available tools like Font Awesome and Google Charts.

The master page provided is httpdocs/masterpages/pdf.php

It is essentially default.php without the navigation or controls included. We’re not going to want our menu included on PDFs and we’re not going to be popping up notices to the user.

Normally your master page is defined in the Init or DoInit methods. This is still used on the page so it can be displayed in simple HTML in a browser for previewing.

However, the master page will need to be overwritten in the ExportToPDF or DoExportToPDF method of your page model. The framework will check for MasterPage again before it renders the PDF.

I’ve created a define(‘MASTERPAGE\_PDF’,’pdf’) to make it easy to reference.

You will also need to create the DoExportToPDF or Export to PDF function for your page you are enabled PDF exports on.

**public static function** DoExportToPDF()  
{  
 **self**::*$MasterPage* = ***MASTERPAGE\_PDF***;  
 **self**::*$PDFPageOrientation* = ***PDF\_PAGE\_ORIENTATION\_LANDSCAPE***;  
}

There are also a couple defines now available to set the orientation of the page. These are defined in QuickDRY/web/BasePage.php

*define*(**'PDF\_PAGE\_ORIENTATION\_LANDSCAPE'**, **'landscape'**);  
*define*(**'PDF\_PAGE\_ORIENTATION\_PORTRAIT'**, **'portrait'**);

$PDFFileName is used to set the resulting filename that the report is saved as.

$PDFPostRedirect is used when you want the server to create the file but not send it to the user. This is a legacy feature that needs some additional work. The intent is that you can have the server create a file and then it is available to download by the user on another page. Currently, the file is created and sent to the user and has to be recreated if the user want to download it again. Fully implementing this would require a file handler to store the information about where the PDF is physically stored into a database. For now it is an exercise for the user of QuickDRY to implement this for their custom project.

Note that QuickDRY uses the WebKit tool to convert HTML to a PDF. As such, this is an area that the framework could be extended to more fully implement the options available. But what is there is sufficient for your standard reporting needs.