Technical Documentation Smashing

<?xml version="1.0"?><DocumentBlank xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"> <GraphicCharterDefinitionId>0</GraphicCharterDefinitionId> <TemplateBaseTypeId>0</TemplateBaseTypeId> <CompanyId>1</CompanyId> <ConfidentialId>3</ConfidentialId> <ConfidentialDescription>Internal Use Only</ConfidentialDescription> <CountryId>0</CountryId> <PageSizeId>1</PageSizeId> <PageOrientationId>1</PageOrientationId> <PrePrintedStationary>false</PrePrintedStationary> <Project>Pole MCS</Project> <Reference>20180327-160306-CE</Reference> <TemplateType>1</TemplateType> <CultureId>en-GB</CultureId> <LanguageId>1</LanguageId> <Customer>Sopra Steria</Customer> <DocumentDate>2018-05-30T16:04:47.9638861+02:00</DocumentDate> <FirstPageHeader /> <FirstPageSubHeader /> <FirstPageTitle\_Blank /> <FirstPageSubtitle\_Blank /> <Saved>false</Saved> <IsValid>true</IsValid> <FirstPageCover>false</FirstPageCover> <IsNew>false</IsNew> <Title>Technical Documentation Smashing</Title></DocumentBlank>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  | Internal Use Only | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | | |
|  |  |  | |  |

1. Installation

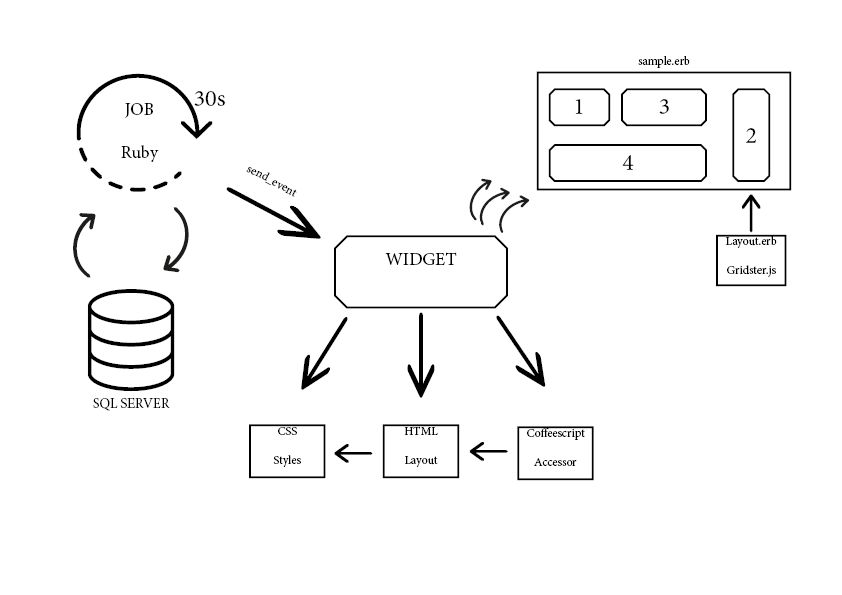
(External document )

2. Explication

Smashing is a Sinatra based framework, which is himself a Ruby framework lighter than the most famous, Ruby on Rails.

Smashing lets you build beautiful dashboards on TV’s screens.

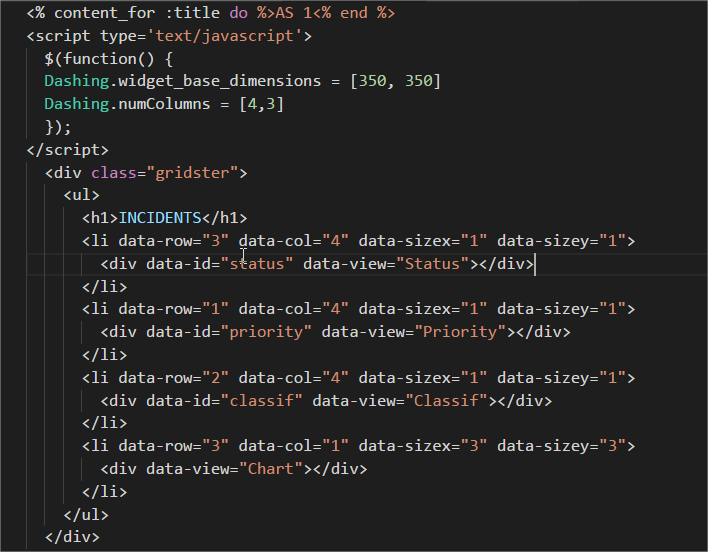
Once you have done the installation, whatever you decided to use a folder who already exist, or install a new smashing. You will have at disposition a directory setup as follows:

* Assets — All your images, fonts, and JS/Coffeescript libraries.
* Dashboards — One .erb file for each dashboard that contains the layout for the widgets.
* Jobs — Your ruby jobs for fetching data (e.g for calling third party APIs like twitter).
* Lib — Optional ruby files to help out your jobs.
* Public — Static files that you want to serve. A good place for a favicon or a custom 404 page.
* Widgets — All the HTML/CSS/Coffeescript for individual widgets.

Example Dashboard:

Here's our AS dashboard with 4 widgets. With a name of AS.erb, it becomes accessible at

“localhost:XXXX/AS” in your navigator once you start smashing.



The first line is for the name of the tab in Chrome.

The line in the script, defines the dimension of each widget.

Each widget is represented by a **div** element needing data-id and data-view attributes. The wrapping **<li>** tags are used for layout.

You can consider your Dashboard as a grid, in the **<li>** you have 4 informations, the “data-row” and “data-col” are the layout of the widget in the grid.

Example: data-row=”1”, data-col=”1” will be the first element at the top left of your dashboard.

You can’t put two widgets at the same layout.

The data-sizex and data-sizey are the size of the widget, X the width and Y the height, both integer.

In the **div** element:

**“data-id”:** Sets the widget ID which will be used when pushing data to the widget. Two widgets can have the same widget id, allowing you to have the same widget in multiple dashboards. Push data to that id, and each instance will be updated. The name for the data-id will be the name of your “send\_event” in the job ruby. But we will see that later.

**“data-view”:** Specifies the type of widget that will be used.

Anatomy of a widget:

* an HTML file used for layout and bindings
* a SCSS file for styles
* a coffeescript file which allows you to handle incoming data & functionality

These informations are true for the initial widgets provided by smashing that you can find at this link:

<https://github.com/Smashing/smashing/wiki/Additional-Widgets>

These widgets are really simple to use, you just need to copy-paste the code and create a new widget folder and you can display it on your dashboard.

But, for the Gantt Part and the AS part, we just created our widgets with some libraries pre-existing.

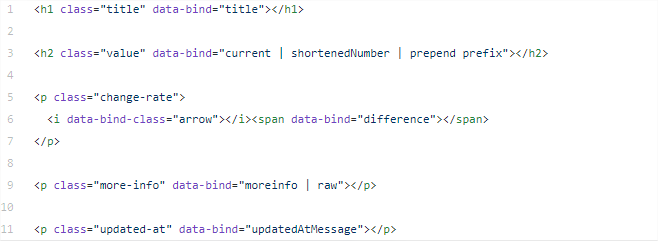
It’s not really different, just that the HTML file is far longer because he contains the JS that we use from the library that we put into the folder “ASSET”.

HTML:

Here is an example of a really simple widget incorporate in Smashing, the widget number.

The HTML is really short and the piping '|' characters in some of the data-bind's above are Batman filters (from the batman.js library), they let you easily format the representation of data.

The data is binded from the Coffeescript who catch the data.

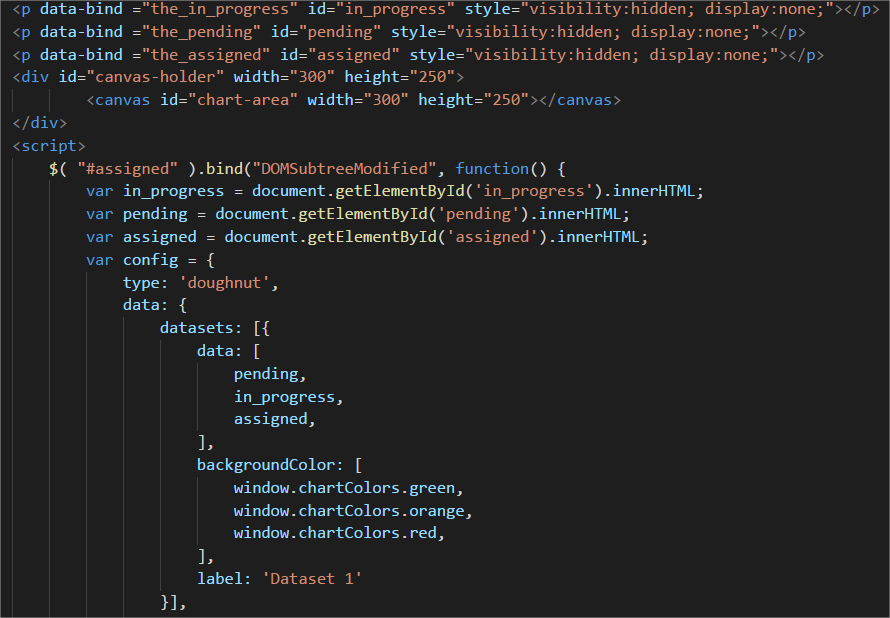


Here is a HTML file more specific that we use to create the AS part. He is far different from the one above. We bind all the element pushed by the coffee that we hide. And we put all the treatment of data in a JS Script.

To access at the elements binded in the script, we simply use some “document.getElementById(‘’id)”

Note that at the line “**$( "#assigned" ).bind("DOMSubtreeModified", function() {”**

The element in the parenthesis (here assigned) need to be the last element that we binded above.



All the script part is taken from the individual

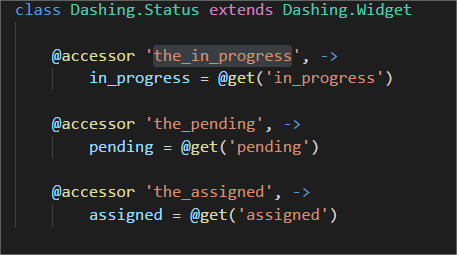
COFFEESCRIPT

The coffeescript is a little language that compiles into JavaScript, you can find everything about it at this link: <http://coffeescript.org/>

It permit you to take the data from the job ruby with some “accessor”. You have full of example of coffee file who fetch the data from the ruby.

This is one example:

It is the coffeescript for the HTML that we present before. There is 3 accessors, who fetch the data from the “send\_event” of the ruby file. Just create a new variable and @get() the name of the data in our send\_event.



Class Dashing.**Status** extends Dashing.Widget

“Status” will be the data-view in the ERB file.

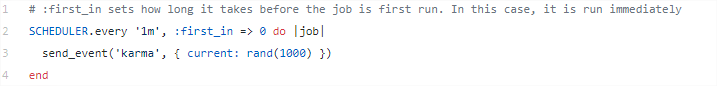
IMPORTANT NOTE:

The indentation is really important in coffeescript, and there is a lot of errors with that. To make you the life easier, just copy-paste your code at the same link as before (<http://coffeescript.org/>) at the tab “TRY COFFEESCRIPT”.

It will show you what is the conversion in Javascript and tell you immediately if there is some error of indentation.

JOBS

Smashing uses rufus-scheduler to schedule jobs. You can make a new job with dashing generate job sample\_job, which will create a file in the jobs directory called sample\_job.rb.

This

The job will run every minute, and will send a random number to ALL widgets that have data-id set to 'karma'.

This is the easier job that you can ever find.

You send data using the following method:

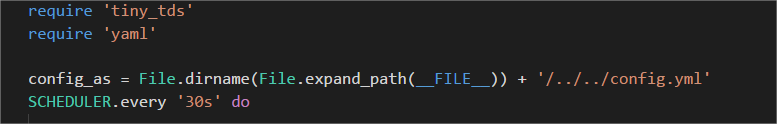
send\_event(widget\_id, json\_formatted\_data)

For the widget “Status”

The variable “in\_progress, pending, assigned” are both numbers, these data is fetched by the coffee and binded in the HTML, in the AS.erb, the data-id is Status.



Jobs are where you put stuff such as fetching metrics from a database, or calling a third party API like Twitter. Since the data fetch is happening in only one place, it means that all instances of widgets are in sync. Server Sent Events are used in order to stream data to the dashboards.



Before the Scheduling you need to insert the “gem” that you will need.

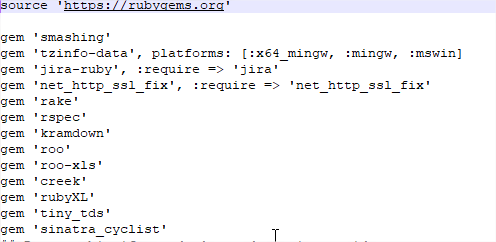
Ruby libraries are contained into a format called "gem"

Here tiny\_tds is for fetching the data base SQL Server. Yaml is for read the config file “yml” where the variable are stacked

Every Gem that you will need, is necessary in the Gemfile, at the root of your Smashing directory.

If you want to use a new one you need to enter as follows “gem ‘name\_gem’ ”, then run the command “bundle install” in your console.

You can find every gem that you want at this link: <https://rubygems.org/>



Note that if you modify a job then Smashing needs to be restarted to pick up the changes.

If you want to only execute the job, without smashing, the method is to put in comments the Scheduler and the send event, and run it in the console with “ruby nameofyourfile.rb”. Of course after moving in the folder of your job.

SQL Server

Download SqlServer Management Studio at this link:

<https://docs.microsoft.com/fr-fr/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>

It must be useful for you to do the request before doing any job ruby.

Start every morning

Search: Edit Group Policy

* Computer Config
  + Windows Setting
    - Scripts (startup/Shutdown)

