Diem7 - Diem & Symfony2

Discussions on how to implement Diem in Sf2

Abstract:

In Symfony2, many symfony1 concepts have changed. Also, there are other things possible that needs discussion between core & community developers.
This draft is a place for me, Stéphane Erard, to put them down paper so we have a starting point to discuss on.

Please write your name (login) and writting color here
Kye Etherton
Nikola Svitlica (a.k.a. TheCelavi joined in)
Maksim Borisov
Anton Stoychev
Jarek Rencz

We better comment the text

1. FrontEndControllers : index.php & admin.php

In sf1, we have two distinct applications, namely front & admin, which have frontend controllers in web/ in the files index.php and dev.php for front application, and admin.php & admin_dev.php for admin application.

In Sf2, Application is just a way to register Bundles.
But some are saying that ONE APP FIT THEM ALL.
I don't really agree.

1. The Bundles Option

We make everything a Bundle. We see Bundles as we see Modules in sf1.
AppKernel.php will register ALL Front & Admin Bundles.

Having Front & Admin within the very same app/ might have pros & cons :

1. No more problem in generating urls for the "other end"
2. Securing admin application is made using "firewall" concept of Sf2
3. … ?

The directory structure would be similar to what you have when you install.

I have made three bundles, FrontBundle, CoreBundle & AdminBundle.
FrontBundle declares a route with /
AdminBundle declares a route with /admin
Then in my app/config/routing.yml I declare

```
DiemAdminBundle:
   resource: "@DiemAdminBundle/Resources/config/routing.yml"
   prefix:   /

DiemFrontBundle:
   resource: "@DiemFrontBundle/Resources/config/routing.yml"
   prefix:   /
```

You HAVE to declare the DiemAdminBundle route *before* the front one because its a "FIFO" implementation : first route matching is used.
And /admin/something matches the route {slug} of DiemFrontBundle, so we won't get into the admin bundle thing. Terrible option! This will just cause huge amounts of extra processing time.

Another thing considering this, on high-volumes for frontend, is that we will always check for the Admin whereas we know it is for front. This tells me to consider using the Application option. But perhaps I'm looking too high... Would need bench etc.

Also, see http://blog.liip.ch/archive/2011/05/19/symfony2-bundle-structure-a-use-case.html

2. The Application Option

We make two distinct applications, so we create two distinct directories.
We would then stick to actual way of doing.

- diem7/
  - front/
    - FrontAppKernel.php (AppKernel.php)
    - autoload.php
  - admin/
    - AdminAppKernel.php (AppKernel.php)
    - autoload.php
  - web/
    - index.php
    - dev.php
    - admin.php
    - admin_dev.php
  - src/
  - vendor/
  - ...

Instead of just having "app/", we can create two directories, "admin/" & "front/", then make the Frontend controllers in web/ (those app.php files).
At first try, it is doable !
One thing, is that we would have to do some extra coding about configurations.
This can be done during installation process.
front/ & admin/ applications will have their own config/ files, thus they can use different databases, different things for anything, but it makes us duplicate things when we want the same thing over admin & front apps.
It's not a big deal anyway.

2. Take over Symfony2

Most of us, see and are amazed by how Diem overloads symfony code.

Yeah, it just amazed me.

I think we will need to take over Sf2 in the pretty same way.

For example, seeing app/autoload.php, we can see that there are ways to make it taken over by Diem. Simply by making it within a Diem Bundle, and using this file as an include.

This section needs more, *really* more oversights.

See http://www.martinsikora.com/symfony2-and-dependency-injection#how-to-write-my-own-service

About Doctrine & Paged objects

Overload the doctrine managers (ORM + ODM) to add Diem7 behaviors (page management for page: true objects).

3. Having Diem developers taking over the admin generators

As of now, some AdminGenerators exist in the Sf2 world.
We need some of us to see what they are capable of, and take part in the development phase of these Bundles (because, yes, these are Bundles too !).

Also, I'd like to say this:
There are AdminGenerators and Admins.
AdminGenerator is what we know from sf1 while AdminBundle is a concrete bundle with extension points on which you can hook to extend behavior etc.

cedriclombardot.github.com/AdmingeneratorGeneratorBundle/
https://github.com/CodeMeme/AdminBundle
https://github.com/sonata-project/SonataAdminBundle (<= best choice ? doc here http://rabaix.net/AdminBundle/html/index.html)

http://sonata-project.org/bundles/admin/master/doc/reference/architecture.html
(doc for AdminBundle)

Here's a demo of SonataAdminBundle integrated with all addon bundles:
https://github.com/marcosQuesada/Symfony2--Sonata-Full---FOSUserBundle

4. About the Admin application

How to handle the routes ?

We can handle routes the same way we do in Diem 5.
Routes represent a way to execute an action (a controller in sf2).
In my opinion, we better not lock it to a 3 level menu, but simply make it lose.

Thus, routes on admin would be like routes on front : doctrine slugs.

In fact, I’m wondering if the admin & the front shouldn’t be similar applications.
We then would be able to construct pages as we wish, then drop widgets in containers, mark them secure for given ACLs, etc.

The way Diem 5 handles admin menu & modules is really great, but needs overload when it doesnt fits your needs. Here we would make it faconable to any need on the first strike.

We’d have a menu.yml config file (in app/, in bundles, wherever u want, like modules.yml).
This menu.yml file would describe menu structure and controllers, if any.
Entries might not necessarily be linked to controllers, let say I want a menu entry to add some JS event handlers to it.

5. How to take over Diem7 ?

Sf2 makes heavy use of DIC, and when you use Diem in some extreme uses (in a development perspective, saying you really overload it without touching its core classes*)

As in Diem 5, any thing might be implemented as a service.
Bundled Classes from Diem7 core|admin|front should consider offering extension points to be overloaded by inheritance. We could consider it on-demand, by making an RFC in mailing-list then applying a patch or such (which is the strategy employed by Symfony2 developers, as pretty lot of methods are private and will be made protected -for inheritance overload- on-demand with *valid* arguments. The other ways are to provide extension points (events) or delegation mechanisms (specific classes-for-methods as service)).

Consider the sfFormDoctrine and its ->save() method calling ->doSave(), for example.

* in sf1, you can overload core classes by copy/paste them in your project-level or within your plugins, thanks to the sfAutoload class which generates a php file with an array of files. It will parse all .class.php files following PEAR naming convention.

So, let say there is a class in dmFrontPlugin/lib/ that you want to overload, you can do so at project-level, at application-level or even at plugin-level, until your plugin is registered after Diem plugins, which is always the case. - This is for Kye :)

6. How to overload Bundles ?

http://sonata-project.org/bundles/easy-extends/master/doc/reference/why.html

7. How to handle Data ?

Diem does a very great job at using Doctrine.
In Doctrine2, there are many, many ways of doing things.
There is ORM, of course, but also ODM, which is known to be much much faster than any ORM thing, as it is a NoSql thing, thus using key-store or graph-based next-gen databases such as CouchDB or MongoDB.

Also, I would really appreciate, and in fact it is a kind of mandatory feature for me, at least at first seek, to not lock-in developers into any ORM-ODM-MongoDB-CouchDB dependency.

The same way we choose which sql server to use in Diem install process, we would be able to choose which data storing infrastructure we want to choose : ORM(Mysql, mssql, pgsql, etc, all the stuff handled by Doctrine ORM) XOR ODM (CouchDB, MongoDB, etc, all the stuff handled by Doctrine ODM).

update
Doctrine2 handles objects as POPO, so the only thing to consider here is how to hook into Doctrine2 to load ORM or ODM (as specified by user in app/config) definition files.

clarifications following TheCelavi comment
Diem has one class named DmPage.
We can provide both ORM & ODM YAML files for this concept.
When using ORM, DmPage will be an Entity.
When using ODM, DmPage will be a Document.
We should/must let the end-developer choose which one he prefers.
This is what I meant in my **update**

Things to consider about Data :

- Pages
- Widget Content
- I18N & L10N
- Data Environments (Prod, Staging, Validation, etc)

    - I’m not talking about website environment with admin_dev.php, but more about *data environment*, where users can see different data depending on which data env they are in (and if they have the right to change that).
    - Implementing this in Diem 5 might not be that hard with Doctrine behaviors and some tricks.
- Rights management

In the actual PHP world, there is a PHPCR project, namely PHP Content Repository, which aims at providing an API at manipulating Contents.
We might be keen at being part of it.
Using this PHPCR would let us abstract away all these problems from Diem7, and make Diem7 an interface to PHPCR functionalities.

Another topic is how to handle Data in the sense of object properties from within Domain Model ?
This is where Diem is best.
Let say I have a Car model with a Name and a Description. I want data environment for them. How to go ?

8. Pages - Section proposal

These are just some thoth's/feedbacks with issues that I have stumbled upon:

- Lifetime span -> visible from date to date, or to date, or from date...
- If page is hidden (i.e. not public, not available to the visitors) all sub-pages should not be reachable to the visitors as well.
- This is also for indexing/searching, if page is hidden/not available for indexing/searching, all sub pages should be omitted from that too
- Page versioning - possibility of undo/redo

What have left for us is to think about mobile versions of the pages - mobile devices are more popular now days and what have to be discussed is whether we will implement some logic that will support this too. Is there going to be 1 on 1 translation of the pages for mobile devices, some automated process, or it is required to create yet another web site.

What ever decision is made - this should not be top priority if its going to take considerable amount of time.

9. Widgets, Areas, Zones, Templating

- Only using Twig
- Using a grid system or such

    - Possibility to switch ? Implementing something like is done with assets in Diem 5
    - use_layout_system('960'); ?
        - would include JS/CSS
        - themes depending on layout systems or not ? (needs CSS insights on this)
    - http://960.gs/
    - http://stacklayout.com/
    - http://www.gridsystemgenerator.com/
    - http://www.tripwiremagazine.com/2009/06/css-grid-layout.html
    -
- Two concepts only, Widget and Container
    - Containers can be nested, whereas Widgets cannot.
    - Containers would be similar to Zones, they can be placed by front editors
    - So we can provide a Layouting editor
- Being able to declare Containers & their Widgets from templates (Twig)
- Widgets & Containers would have properties on their own (wrapping tag, classes)
- Being able to reuse widgets (shared widgets -thanks to Nikola)
    - reuse content of widgets
    - reuse properties of widget
    - saying it all, it would be a "singleton" (Apostrophe does it, but its not dynamic in the sense a front editor cannot add a singleton widget-slot, it's only declareable from within php templates)
- Being able to attach behaviors (js) to Widgets & Containers (Thanks Nikola)
    - Programmatically & via Widget editing
    - Possibility to add values, stored as an array or json
    - Let say you have a widget echoing a bunch of <li> of <img>, you want to use this Gallery behavior to make it a gallery, with the paginable option.
- Being able to make Containers & Widgets as Data are, about Data Environments.
- Page/Area/Zone/Widget - Lifetime span -> visible from date to date, or to date, or from date...
- Widget undo/redo?

10. Javascript ? CSS ? HTML ?

HTML5, CSS3, CoffeeScript, LESS|SASS.
Using assetic.

JS code would be written in CoffeeScript and in natural JS
CSS3 would be written using Less or Sass or natural

jQuery + jQuery UI as standard library for handling DOM and UI.

To reconsider JS and CSS dependency injector for loading required libraries (per example, if $('div').accordion() is called - it would be neat to load in that point of time required JS and CSS files for display) => issue, XHR will keep session alive, and if developer does not want that -> well, dependency injector will be a problem. Voting? :)

Possible solution - usage of CDNs, different domain, no session keep alive?

I suggest picking one of 3 frameworks: Ember, Knockout or Angular as a foundation of admin application/bundle. jQuery UI is nice but from my own experience I'd say it is not enough for modern application. Since I have some experience with Angular i propose choosing it as main JS framework. I provides both excellent testability and great modularity. This is what is needed for for frontend editing (one widget = one angular element directive with services to make them talk to each other and to frontend editing forms). It shouldn't be obligatory to use Angular in front with editing mode off. Angular is provided with jqLite but works well with full stack jQuery.
We should decide how to build front part of a project: namely - if we will be using grunt.js and if so, are there any cons of using nodeJS. I think we definitely should at least to run tests using Karma. It is to be decided where should be assets built: using assetic (in sf2) or using some grunt plugins. (It connects directly to topic of deployment but sadly I don't have much time to write about it today). Please comment this general idea.

11. CDN

We must be able to define CDNs to deliver medias.
Kye is able to help us on this
Mostly ready for 5.5 anyway. Just needs to set up rsync's for the folders.

12. Subdomains

We must be able to define and link to subdomains.
The Page concept must be linked to a domain, be it root or sub.
So, somewhere in a config/ directory on root, we should be able to declare subdomains
I don't think it should be needed to make it dynamic via Doctrine (make it a Doc or an Entity might not be that relevant, except if we want to somehow link them with other things).

Thanks to Kye
Subdomains can be dynamics, finally. Let say I want "user1.mydomain.com".
So subdomains will be POPO classes accessible via a Sf2 service. You'll be able to define the class to use for subdomains, still it will need to implement an interface (to keep type).

http://www.craftitonline.com/2011/08/symfony2-locale-on-subdomains-not-on-the-url-path/
http://stackoverflow.com/questions/5366234/symfony2-routing-route-subdomains

a. Subdomains for languages ?

We might be able to handle subdomains for languages.
If we want to redirect user to DiemFrontBundle_homepage with parameter culture=fr, it then should redirect me to fr.mysite.com if it is configured to do so.
Also, mysite.com/fr would redirect me to fr.mysite.com ? Make this behavior configurable ?
Would do another DNS lookup, but once, not every.
There are a lot of cases that use subdomains.... Ex.: cities separation, product-category: tv.shop.com; note.shop.com...
Subdomains should be the hardcoded instance of the page... I want to be able to redirect user from one subdomain to another.
1 case) I have 4 domains as satellite, and need to redirect with moved premanently code to one-main domain
2 case) I have 3 domains, that logacally split my site... but I want to manage all of them from one place. http://forum.diem-project.org/viewtopic.php?f=24&t=372 here was some discussion on that case

update
Ok for subdomains for anyuse. Remember Diem is a CMF, it provides tools & code sugar to help you realize what is needed to do.
If we keep the Pages bar on front, it will show only *current domain* pages.
All pages from all domains will be shown on another admin module.
I think that a DmPage should be accessible from multiple domains too, though, thus the domain will be a variable passed by to views.

13. Test

We would use Mink, Behat and other JS-proxied stuffs to unit test, functional test and browser test Diem7

See https://gist.github.com/1206414

14. Images

We will use Imagine library

15. Auth

16. Security

Firewall from Sf2

17. Bundles

A list of Bundles that are useful to see/use/check or are inspiring. We should put anchors/links to these Bundles into their relevant section

- http://symfony2bundles.org/stof/StofDoctrineExtensionsBundle
- http://symfony2bundles.org/knplabs/KnpPaginatorBundle
- http://symfony2bundles.org/FriendsOfSymfony/FOSJsRoutingBundle
- http://symfony2bundles.org/schmittjoh/JMSDebuggingBundle
- http://symfony2bundles.org/Elao/WebProfilerExtraBundle
- http://symfony2bundles.org/liip/LiipThemeBundle
- http://symfony2bundles.org/Exercise/FOQElasticaBundle
- http://symfony2bundles.org/antimattr/GoogleBundle
- http://symfony2bundles.org/symfony/DoctrineFixturesBundle
- http://symfony2bundles.org/nelmio/NelmioSecurityBundle
- http://symfony2bundles.org/Behat/BehatBundle
- http://symfony2bundles.org/knplabs/KnpMarkdownBundle
- http://symfony2bundles.org/Behat/MinkBundle
- http://symfony2bundles.org/knplabs/KnpTimeBundle
- http://symfony2bundles.org/schmittjoh/JMSDiExtraBundle
- http://symfony2bundles.org/BeSimple/BeSimpleI18nRoutingBundle
- http://symfony2bundles.org/schmittjoh/JMSI18nRoutingBundle
- http://symfony2bundles.org/jonaswouters/XhprofBundle
- http://symfony2bundles.org/symfony/DoctrineMigrationsBundle
- http://symfony2bundles.org/ornicar/FOQTyperBundle
- http://symfony2bundles.org/schmittjoh/JMSSerializerBundle
- http://symfony2bundles.org/knplabs/KnpConsoleAutocompleteBundle
- http://symfony2bundles.org/robzienert/CalendarBundle
- http://symfony2bundles.org/pablodip/DoctratorBundle
- http://symfony2bundles.org/BeSimple/BeSimpleSoapBundle
- http://symfony2bundles.org/dstendardi/AriadneBundle
- http://symfony2bundles.org/Exercise/HTMLPurifierBundle
- http://symfony2bundles.org/Bazinga/BazingaExposeTranslationBundle
- http://symfony2bundles.org/opensky/OpenSkyRuntimeConfigBundle
- http://symfony2bundles.org/knplabs/KnpGaufretteBundle
- http://symfony2bundles.org/excelwebzone/EWZRecaptchaBundle
- http://symfony2bundles.org/schmittjoh/JMSCommandBundle
- http://symfony2bundles.org/oaugustus/DirectBundle
- http://symfony2bundles.org/symfony-cmf/ChainRoutingBundle

- http://symfony2bundles.org/craue/CraueFormFlowBundle
- http://symfony2bundles.org/liip/LiipContainerWrapperBundle
- http://symfony2bundles.org/moreweb/ImagineBundle
- http://symfony2bundles.org/madiedinro/FirePHPBundle
- http://symfony2bundles.org/Problematic/AclManagerBundle
- http://symfony2bundles.org/Funkiton/InjectorBundle (see info)
- http://symfony2bundles.org/craue/TwigExtensionsBundle
- http://symfony2bundles.org/avalanche123/MicroKernelBundle (MicroDiem ?)
- http://symfony2bundles.org/nicodmf/HighlightBundle (for code editor)
- http://symfony2bundles.org/Oryzone/OryzoneBoilerplateBundle (html5)
- http://symfony2bundles.org/CodeMeme/CodeMemeDaemonBundle (daemon'izing)
- http://symfony2bundles.org/opensky/SitemapBundle
- http://symfony2bundles.org/ioalessio/IoFormBundle (form types)
- http://symfony2bundles.org/lvanderree/DbRoutingBundle
- http://symfony2bundles.org/samjarrett/DoctrineSluggableBundle
- http://symfony2bundles.org/symfony-cmf/DoctrinePHPCRBundle
- http://symfony2bundles.org/naholyr/ABBundle
- http://symfony2bundles.org/Fristi/SessionBundle
- https://github.com/naderman/PyrusBundle (PEAR2 package manager)
- http://symfony2bundles.org/knplabs/KnpTestSessionBundle
- http://symfony2bundles.org/knplabs/KnpMediaExposerBundle(CDN ?)
- http://symfony2bundles.org/21Net/DebianizeBundle
- http://symfony2bundles.org/dennisoehme/GOCLocaleBundle
- http://symfony2bundles.org/winzou/AutocompleteBundle
- http://symfony2bundles.org/zeopix/RssBundle
- http://symfony2bundles.org/poulikov/DoctrineMongoDBBundle
- http://symfony2bundles.org/choffmeister/ChartBundle
- http://symfony2bundles.org/lbotsch/SpecBundle
- http://symfony2bundles.org/rubensayshi/AlohaBundle
- http://symfony2bundles.org/Xaddax/SproutCoreBundle
- http://symfony2bundles.org/timeoutdigital/TOLocalizationBundle
- http://symfony2bundles.org/comfortablynumb/ApplicationServiceAbstractBundle
- http://symfony2bundles.org/cordoval/PhpSpecBundle
-

18. Git

We better adopt a branching model.

master : production-ready
devel : latest devel version
tagged : versions

It also helps in isolating code mods within specific branchs (which die when merged into devel or master -hotfixs)

http://nvie.com/posts/a-successful-git-branching-model/

19. Design of Admin and Front interface

I propose some aqua/graphite color based interface, dark and light themes. Dark for light colored websites and light for dark colored websites in order to have contrast.

Beside Diem 5 original theme -> no one ever published new theme for Diem (maybe it was developed, community did not see it). This time, efforts should not be wasted on supporting this, Admin and Front interface in Admin mode should be “as is”, lets do it properly so everyone is content :)

- Page tree panel: http://www.jstree.com/ - with context menu support and other operations as well (per example - in Front, you did not have possibility to organize pages)
  - Add page
  - Edit page
  - Delete pege
  - Move page
  - Search
  - Dynamic load of page tree (in order to support sites with no of pages 100+)
- Media panel: everything supported from Admin should be supported here as well. I hated because for image upload i had to go to the Admin, upload images, then back to the Front again (per example for HTML content plugin - TinyMce)
- Media and Page tree could be jQuery dialogs - you can open them from toolbar and keyboards shortcuts for opening/closing.
- Front toolbar - redesing? http://www.wizzud.com/jqDock/ ?
- Add widget and Add behavior - Apple Mac OS X like grid + searching/filtering capability with http://razorjack.net/quicksand/
- Add/Edit Widgets, Behaviors - usage of icons instead of text in box?
- Interface theme (buttons etc...) Aristo? http://www.warfuric.com/taitems/demo.html
- Alert popup, question/conformation popup - little bit advanced system to be available to the developers of widgets/behaviors: http://okonet.ru/projects/modalbox/index.html
- Graphite icon set (there is better one, i m shore of it): http://www.webdesignerdepot.com/2010/07/200-exclusive-free-icons-reflection/

Panels

In old Diem it was not comfy to ADD WIDGET, huge number of widgets appear in small popup... Quick search filter can help only if you remember the widget name. I suggest to move all widgets, that can be added to a page, move them all to right panel. If we look at the right panel: there are three buttons there: “pages”, “media”, “widgets” (for example), click to the pages button open pages panel, and so on, everithyng in one place. We want to see as much height of page as we can, that’s why it’s wrong to open popup with smth. at bottom.

20. Performance

In the Diem5.4 we’ve got trouble with memory managment while page number getting over 1000, specifically in seo-synchronization task. The same thing with synchronization of media (it runs in every request to open media-bar). We should be carefull with writing sql-queries and getting results as Doctrine-objects, cause they eats a lot of memory. In the task, that should works with all records in the table we should use limit query with iteration. This method can guarantee limited memory usage.

For instant response generation (http://developer.yahoo.com/performance/rules.html) we can rebuild kernel modules to do the following:

1) only one sql request ~~to rule them all~~ to fetche page information, and generate <head>...</head> block of responce - it’s quick and anought to do <?php flush(); ?> and user-agent will download required css and other resources while the rest page <body> is being generated, doing a lot of sql requests, working hard...

2) only site-wide resources can be included as <script> and <link>, wigdet-wide css and js must be inline... Instead <?php use_stylesheet(...) ?> in widget template you write <?php inline_stylesheet() ?> and required css will be inlide page, this is important for big projects with great number of different widgets.

21. Widget resourses organisation

See p20 -> (2) (some lines above)

Also it’s a good practice to use unique css class for a widget:
Example:

```
@someModule/templates/_someWidget.php :
<?php inline_stylesheet(‘someModule/someWidget.css’) ?>
<div class=”someModule_someWidget”>...</div>

@someModule/someWidget.css :
.someModule_someWidget {
	backgound: red !important;
```

}

And scripts better be placed inline manual, don’t use functions to include smth., when I’m looking to html structure I want to see js in th esame place, to understand that’s happening. In the same time it’s good for pageload speedup.

22.
23.
24.
25. **ffezfezf**