

# LABINVENT - ANNEXES

**URL officielle de ce doc :** <https://tinyurl.com/labinvent-Annexes>

**Auteurs:** E. Pallier, E. Bourrec

**Version:** 30/11/2020

⇒ *Pour faire une copie Word, Libre/Open Office, ou PDF de ce doc : menu "Fichier/Télécharger..."*

*Cette documentation annexe contient une mine d'informations techniques qui s'ajoutent à la [documentation technique de référence](#).*

N'hésitez pas aussi à la mettre à jour vous-mêmes quand c'est nécessaire, ou bien à soumettre vos suggestions à :

*epallier AT irap POINT omp POINT eu*

## HISTORIQUE DES MISES A JOUR DE CETTE DOCUMENTATION

- 25/11/20 Mise à jour du chapitre "[Étiquettes](#)" pour ajouter la nouvelle étiqueteuse Dymo LabelManager 420P
- 19/10/20 Ajout nouveau chapitre "[Astuces \(howto\)](#)"
- 29/9/20 Mise à jour du chapitre "[Installations existantes](#)" pour ajouter IP2I
- 28/9/20 Ajout d'un [nouveau chapitre expliquant la gestion des notifications](#)
- 8/9/20 Ajout d'un exemple dans le howto "[Ajouter une table dans la BD](#)" (ajout de la nouvelle table "**projets**")
- 4/9/20 Mise à jour du chapitre "[Étiquettes](#)" suite à la mise en place de la nouvelle étiqueteuse Dymo MobileLabeler
- 24/7/20 Mise à jour du chapitre "[Installations existantes](#)" pour IAS
- 23/7/20 Ajout de CETTE page-ci
- 23/7/20 Modification/Ajout des chapitres suivants :
  - [Configuration des autorisations](#)
  - [Configuration des logos](#)
  - [Configuration des étiquettes](#)
  - [Configuration niveau DEV only](#)

## TABLE DES MATIÈRES

<b>1. Installation des pré-requis sur Mac OS 11 BigSur (avec HomeBrew)</b>	<b>5</b>
<b>2. Installation des pré-requis sur Mac OS X (avec HomeBrew)</b>	<b>6</b>
2.0.1. Cleanup BREW	6
2.0.2. APACHE	8
2.0.3. PHP	12
2.0.4. MYSQL => MARIADB	26
2.0.5. Apache Virtual Hosts	27
2.0.6. PHPMYADMIN	29
2.0.7. Installation de l'extension YAML	31
<b>3. Installation des pré-requis sur Windows 10</b>	<b>33</b>
3.1. Installation directement sur Windows 10 (TODO)	33
3.2. Installation sur une VM linux (avec VirtualBox)	33
<b>4. Installation et configuration du logiciel depuis l'IDE Eclipse</b>	<b>38</b>
<b>5. UPDATE DE LA BRANCHE MASTER (dev assermenté only)</b>	<b>50</b>
<b>6. Auto-génération du code (avec "bake")</b>	<b>50</b>
<b>7. METHODES ET CONCEPTS INTERESSANTS A UTILISER</b>	<b>51</b>
<b>8. Comment on a simulé un LDAP pour la connexion utilisateurs en mode LOCAL</b>	<b>54</b>
<b>9. Comment on a rendu le logiciel plus "RESPONSIVE" (avec bootstrap)</b>	<b>57</b>
9.1. Analyse de l'existant et choix d'une solution technique	57
9.2. Dépendances	58
9.3. Installation du plugin bootstrap-ui	58
9.4. Fichiers installés par bootstrap-ui	59
9.5. Informations générales	60
9.6. Principe d'utilisation	62
9.7. BUGFIXES	64

9.8. Revenir à la version NOT RESPONSIVE (pour debug)	65
<b>10. Roadmap (feuille de route) (TODO LIST)</b>	<b>66</b>
10.1. TODO LIST temporaire à migrer sur le Redmine (notes urgentes)	66
10.2. ANCIENNE ROADMAP	83
GENERAL	83
USERS	84
ETIQUETTES	86
MATERIELS	86
SUIVIS	89
EMPRUNTS	89
10.3. VERY OLD TODO LIST A VIRER BIENTOT (en italique ce qui a déjà été migré dans le tableau "TODO LIST" ci-dessus)	91
<b>11. Migrations de la BD</b>	<b>96</b>
11.1. Procédure proposée pour garder la BD à jour suite à nos modifs (pour qu'on ait tous la même version de la BD)	96
11.2. Présentation du concept	97
11.3. Exemples divers	99
11.4. Générer une Migration à partir d'une Base de Données Existante	100
11.5. Nom de Fichier des Migrations	101
11.6. Définition de Colonnes	101
11.7. Créer une Table	102
11.8. Ajouter des Colonnes à une Table Existante	103
11.9. Générer un diff entre deux états de base de données	104
11.10. Les Commandes	105
11.10.1. migrate : Appliquer les Migrations	105
11.10.2. status : Statuts de Migrations	105
11.10.3. seed : Remplir votre Base de Données (Seed)	106
11.11. Trucs et Astuces	110
11.11.1. Mettre à jour les Noms de Colonne et Utiliser les Objets Table	110
11.11.2. Migrations et déploiement¶¶	110
11.11.3. Renommer une table	110
11.11.4. Ne pas générer le fichier schema.lock	110
<b>12. Gestion du temps (DATE &amp; TIME) et localisation</b>	<b>112</b>

12.1. Contexte général (pour votre culture, mais vous pouvez zapper)	112
12.1.1. Main differences and improvements from Carbon to Chronos :	112
12.1.2. Testing and fixating time	113
12.1.3. Usage	114
12.1.3.1. Usage in frameworks	114
12.2. Et dans LabInvent ?	115
<b>13. Le framework CakePhp</b>	<b>117</b>
13.1. Partage de logique dans les modèles, les vues, et les contrôleurs	117
13.2. Le fichier Application (Application.php)	121
13.3. Mon CakePhp Cheat sheet	122
13.4. Erreurs sur le framework (TODO)	136
13.5. Mise à jour du framework CakePhp	136
13.5.1. Upgrade de la version 3.6 à la version 3.7 (16/1/19 pm)	136
13.5.2. Upgrade de la version 3.5 à la version 3.6 (16/1/19 matin)	138
13.6. PLUGINS	140
13.6.1. “Composer”, le gestionnaire de plugins	140
13.6.2. Plugins spécifiques à Cakephp	140
13.6.3. Les plugins installés pour LabInvent	142
13.6.3.1. Génération des QrCodes (plugin phpqrcode)	142
13.6.3.2. Génération des documents PDF	142
13.6.3.2.1. Le Plugin fpdf	142
13.6.3.2.2. Le plugin cakephp-dompdf pour générer des docs pdf	143
13.6.4. Mettre à jour ou Réinstaller tous les plugins (dossier vendor/)	143
13.6.5. Ajouter un plugin spécifique	144
13.6.6. Supprimer un plugin	146
<b>14. Historique du logiciel - Changements importants</b>	<b>147</b>

# 1. Installation des pré-requis sur Mac OS 11 BigSur (avec HomeBrew)

Cette installation a été faite sur la version Big Sur 11.1 le 30/12/20

- [Mise à jour de HomeBrew](#)
- [Apache setup avec plusieurs php](#)

## 2. Installation des pré-requis sur Mac OS X (avec HomeBrew)

Cette installation a été faite sur la version Mojave (10.14.2)

### 2.0.1. Cleanup BREW

Pour plus de détail :

<https://getgrav.org/blog/mac-os-mojave-apache-multiple-php-versions>

```
$ brew uninstall --force php56 php56-apcu php56-opcache php56-xdebug php56-yaml php56-intl
Ignoring bigdecimal-1.3.4 because its extensions are not built. Try: gem pristine bigdecimal --version 1.3.4
Uninstalling php56-intl... (7 files, 895.5KB)
```

```
$ brew uninstall --force php72 php72-apcu php72-opcache php72-xdebug php72-yaml php72-intl
Uninstalling php... (522 files, 76.4MB)
Uninstalling php72-xdebug... (12 files, 679.3KB)
Uninstalling php72-intl... (7 files, 474.5KB)
```

```
$ brew cleanup
Pruned 1 symbolic links from /usr/local
```

```
$ brew list | grep php
phpmyadmin
```

```
$ brew uninstall --force phpmyadmin
Uninstalling phpmyadmin... (2,434 files, 32.3MB)
```

```
$ brew list | grep php
RIEN
```

```
$ rm -Rf /usr/local/etc/php/*
```

```
$ brew --version
```

```
Homebrew 1.9.2
```

```
Homebrew/homebrew-core (git revision d4872; last commit 2019-01-21)
```

```
$ brew doctor
```

```
Your system is ready to brew.
```

```
$ brew update
```

```
$ brew upgrade
```

```
$ brew install openldap libiconv
```

```
Warning: openldap 2.4.47 is already installed and up-to-date
```

```
To reinstall 2.4.47, run `brew reinstall openldap`
```

```
==> Downloading https://homebrew.bintray.com/bottles/libiconv-1.15.mojave.bottle.tar.gz
```

```
##### 100.0%
```

```
==> Pouring libiconv-1.15.mojave.bottle.tar.gz
```

```
==> Caveats
```

```
libiconv is keg-only, which means it was not symlinked into /usr/local,  
because macOS already provides this software and installing another version in  
parallel can cause all kinds of trouble.
```

```
If you need to have libiconv first in your PATH run:
```

```
echo 'export PATH="/usr/local/opt/libiconv/bin:$PATH"' >> ~/.bash_profile
```

```
For compilers to find libiconv you may need to set:
```

```
export LDFLAGS="-L/usr/local/opt/libiconv/lib"
```

```
export CPPFLAGS="-I/usr/local/opt/libiconv/include"
```

```
==> Summary
```

```
 /usr/local/Cellar/libiconv/1.15: 30 files, 2.4MB
```



## 2.0.2. APACHE

\*\*\*\*\*

### 1.1 - Apache INSTALL

\*\*\*\*\*

The latest macOS 10.14 **Mojave comes with Apache 2.4 pre-installed**, however, it is no longer a simple task to use this version with Homebrew because Apple has removed some required scripts in this release. **However, the solution is to install Apache 2.4 via Homebrew** and then configure it to run on the standard ports (80/443).

**If you already have the built-in Apache running, it will need to be shutdown first**, and any auto-loading scripts removed

```
$ sudo apachectl stop
```

```
Password:
```

```
httpd: Syntax error on line 191 of /usr/local/etc/httpd/httpd.conf: Cannot load /usr/local/lib/libphp5.so into server: dlopen(/usr/local/lib/libphp5.so, 10):  
image not found
```

```
==> J'ai Commenté ces lignes dans httpd.conf :
```

```
#LoadModule php5_module /usr/local/lib/libphp5.so  
#LoadModule php7_module /usr/local/lib/libphp7.so
```

```
$ sudo apachectl stop
```

```
(2)No such file or directory: AH02291: Cannot access directory '/usr/local/opt/httpd/logs/' for error log of vhost defined at  
/usr/local/etc/httpd/extra/httpd-vhosts.conf:56  
AH00014: Configuration check failed
```

```
==> J'ai Commenté ces lignes dans httpd-vhosts.conf :
```

```
#####ErrorLog logs/labinvent2.error_log  
#####CustomLog logs/labinvent2.access_log combined
```

```
$ sudo apachectl stop
httpd (pid 6600?) not running
```

```
$ sudo launchctl unload -w /System/Library/LaunchDaemons/org.apache.httpd.plist 2>/dev/null
```

Now we need to install the new version provided by Brew:

```
$ brew install httpd
```

```
Warning: httpd 2.4.37_1 is already installed and up-to-date
To reinstall 2.4.37_1, run `brew reinstall httpd`
```

```
$ brew reinstall httpd
```

```
==> Reinstalling httpd
```

```
==> Downloading https://homebrew.bintray.com/bottles/httpd-2.4.37_1.mojave.bottle.tar.gz
```

```
Already downloaded:
```

```
/Users/epallier/Library/Caches/Homebrew/downloads/4459c998a0822638fecfe41352bbfd0755d17a65e188bff0b3adbffc61dd5e4c--httpd-2.4.37_1.mojave.bottle.tar.gz
```

```
==> Pouring httpd-2.4.37_1.mojave.bottle.tar.gz
```

```
==> Caveats
```

```
DocumentRoot is /usr/local/var/www.
```

The default ports have been set in `/usr/local/etc/httpd/httpd.conf` to 8080 and in `/usr/local/etc/httpd/extra/httpd-ssl.conf` to 8443 so that httpd can run without sudo.


To have launchd start httpd now and restart at login:

```
brew services start httpd
```

Or, if you don't want/need a background service you can just run:

```
apachectl start
```

```
==> Summary
```

```
 /usr/local/Cellar/httpd/2.4.37_1: 1,648 files, 26.9MB
```

**\$ sudo brew services start httpd**

To ensure the server is up:

**\$ ps -aef | grep httpd**

```
0 91454      1  0 10:59  ??      0:00.07 /usr/local/opt/httpd/bin/httpd -D FOREGROUND
70 91455 91454  0 10:59  ??      0:00.00 /usr/local/opt/httpd/bin/httpd -D FOREGROUND
70 91456 91454  0 10:59  ??      0:00.00 /usr/local/opt/httpd/bin/httpd -D FOREGROUND
70 91457 91454  0 10:59  ??      0:00.00 /usr/local/opt/httpd/bin/httpd -D FOREGROUND
70 91458 91454  0 10:59  ??      0:00.00 /usr/local/opt/httpd/bin/httpd -D FOREGROUND
70 91459 91454  0 10:59  ??      0:00.00 /usr/local/opt/httpd/bin/httpd -D FOREGROUND
70 91460 91454  0 10:59  ??      0:00.00 /usr/local/opt/httpd/bin/httpd -D FOREGROUND
501 91463  453  0 11:01  ttys002 0:00.00 grep httpd
```

=> se connecter à localhost[:8080]

Try to restart Apache with:

**\$ sudo apachectl -k restart**

You can watch the Apache error log in a new Terminal tab/window during a restart to see if anything is invalid or causing a problem:

**\$ tail -f /usr/local/var/log/httpd/error\_log**

Apache is controlled via the apachectl command so some useful commands to use are:

**\$ sudo apachectl start**

**\$ sudo apachectl stop**

**\$ sudo apachectl -k restart**

The -k will force a restart immediately rather than asking politely to restart when apache is good and ready

\*\*\*\*\*

1.2 - Apache CONFIG

\*\*\*\*\*

In the latest version of Brew, you have to manually set the listen port from the default of 8080 to 80, so we will need to edit Apache's configuration file.

**vi /usr/local/etc/httpd/httpd.conf**

**Change « Listen 8080 » to « Listen 80 »**

(pas fait:)

Next we'll configure it to use the to change the document root for Apache. This is the folder where Apache looks to serve file from. By default, the document root is configured as `/usr/local/var/www`. As this is a development machine, let's assume we want to change the document root to point to a folder in our own home directory.

Search for the term `DocumentRoot`, and you should see the following line:

```
DocumentRoot "/usr/local/var/www"
```

Change this to point to your user directory where `your_user` is the name of your user account:

```
DocumentRoot /Users/your_user/Sites
```

You also need to change the `<Directory>` tag reference right below the `DocumentRoot` line. This should also be changed to point to your new document root also:

```
<Directory /Users/your_user/Sites>
```

In that same `<Directory>` block you will find an `AllowOverride` setting, this should be changed as follows:

```
# AllowOverride controls what directives may be placed in .htaccess files.
```

```
# It can be "All", "None", or any combination of the keywords:
```

```
# AllowOverride FileInfo AuthConfig Limit
```

```
#
```

```
AllowOverride All
```

Also we should now enable `mod_rewrite` which is commented out by default. Search for `mod_rewrite.so` and uncomment the line by removing the leading `#`:

```
LoadModule rewrite_module lib/httpd/modules/mod_rewrite.so
```

User & Group (pas fait)

Now we have the Apache configuration pointing to a `Sites` folder in our home directory. One problem still exists, however. By default, apache runs as the user `daemon` and group `daemon`. This will cause permission problems when trying to access files in our home directory. About a third of the way down the `httpd.conf` file there are two settings to set the User and Group Apache will run under. Change these to match your user account (replace `your_user` with your real username), with a group of `staff`:

```
User your_user
```

```
Group staff
```

(je l'ai pas fait, j'ai laissé comme c'est:

```
User _www
```

Group \_www  
)

Servername

Apache likes to have a server name in the configuration, but this is disabled by default, so search for:

```
#ServerName www.example.com:8080
```

and replace it with:

```
ServerName localhost
```

Sites Folder (pas fait)

Now, you need to create a Sites folder in the root of your home directory. You can do this in your terminal, or in Finder. In this new Sites folder create a simple index.html and put some dummy content in it like: `<h1>My User Web Root</h1>`.

```
$ mkdir ~/Sites
```

```
$ echo "<h1>My User Web Root</h1>" > ~/Sites/index.html
```

```
$ sudo apachectl -k restart
```

## 2.0.3. PHP

\*\*\*\*\*

### 2.1 - Php INSTALL

\*\*\*\*\*

Up until the end of March 2018, all PHP related brews were handled by Homebrew/php tap, but that has been deprecated, so now we use what's available in the Homebrew/core package. This should be a better maintained, but is a much less complete, set of packages.

Both PHP 5.6 and PHP 7.0 has been deprecated and removed from Brew because they are out of support, and while it's not recommended for production, there are legitimate reasons to test these unsupported versions in a development environment.

Remember only PHP 7.1 through 7.3 are officially supported by Brew so if you want to install PHP 5.6 or PHP 7.0 you will need to add this tap:

```
$ brew tap exolnet/homebrew-deprecated
```

Both PHP 5.6 and PHP 7.0 has been deprecated and removed from Brew because they are out of support, and while it's not recommended for production, there are legitimate reasons to test these unsupported versions in a development environment.

Remember only PHP 7.1 through 7.3 are officially supported by Brew so if you want to install PHP 5.6 or PHP 7.0 you will need to add this tap:

```
$ brew tap exolnet/homebrew-deprecated
```

```
==> Tapping exolnet/deprecated
```

```
Cloning into '/usr/local/Homebrew/Library/Taps/exolnet/homebrew-deprecated'...
```

```
remote: Enumerating objects: 11, done.
```

```
remote: Counting objects: 100% (11/11), done.
```

```
remote: Compressing objects: 100% (11/11), done.
```

```
remote: Total 11 (delta 1), reused 7 (delta 0), pack-reused 0
```

```
Unpacking objects: 100% (11/11), done.
```

```
Tapped 2 formulae (42 files, 78.3KB).
```

We will proceed by installing various versions of PHP and using a simple script to switch between them as we need. Feel free to exclude any versions you don't want to install.

```
$ brew install php@5.6
```

```
(pas fait)$ brew install php@7.0
```

```
$ brew install php@7.1
```

```
$ brew install php@7.2
```

```
$ brew install php@7.3
```

COOL : You no longer have to unlink each version between installing PHP versions as they are not linked by default

```
$ brew install php@5.6
```

```
==> Installing php@5.6 from exolnet/deprecated
```

```
==> Installing dependencies for exolnet/deprecated/php@5.6: mhash and mcrypt
```

```
==> Installing exolnet/deprecated/php@5.6 dependency: mhash
```

```
==> Downloading https://homebrew.bintray.com/bottles/mhash-0.9.9.9.mojave.bottle.tar.gz
```

```
##### 100.0%
```

```
==> Pouring mhash-0.9.9.9.mojave.bottle.tar.gz
```

```
🍺 /usr/local/Cellar/mhash/0.9.9.9: 20 files, 455.4KB
```

```
==> Installing exolnet/deprecated/php@5.6 dependency: mcrypt
==> Downloading https://homebrew.bintray.com/bottles/mcrypt-2.6.8.mojave.bottle.1.tar.gz
##### 100.0%
==> Pouring mcrypt-2.6.8.mojave.bottle.1.tar.gz
🍺 /usr/local/Cellar/mcrypt/2.6.8: 19 files, 397.8KB
==> Installing exolnet/deprecated/php@5.6
==> Downloading https://dl.bintray.com/exolnet/bottles-deprecated/php@5.6-5.6.40.mojave.bottle.tar.gz
##### 100.0%
==> Pouring php@5.6-5.6.40.mojave.bottle.tar.gz
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set php_ini /usr/local/etc/php/5.6/php.ini system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set php_dir /usr/local/share/pear@5.6 system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set doc_dir /usr/local/share/pear@5.6/doc system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set ext_dir /usr/local/lib/php/pecl/20131226 system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set bin_dir /usr/local/opt/php@5.6/bin system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set data_dir /usr/local/share/pear@5.6/data system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set cfg_dir /usr/local/share/pear@5.6/cfg system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set www_dir /usr/local/share/pear@5.6/htdocs system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set man_dir /usr/local/share/man system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set test_dir /usr/local/share/pear@5.6/test system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear config-set php_bin /usr/local/opt/php@5.6/bin/php system
==> /usr/local/Cellar/php@5.6/5.6.40/bin/pear update-channels
==> Caveats
```

To enable PHP in Apache add the following to httpd.conf and restart Apache:

```
LoadModule php5_module /usr/local/opt/php@5.6/lib/httpd/modules/libphp5.so
```

```
<FilesMatch \.php$>
SetHandler application/x-httpd-php
</FilesMatch>
```

Finally, check DirectoryIndex includes index.php  
DirectoryIndex index.php index.html

The php.ini and php-fpm.ini file can be found in:  
/usr/local/etc/php/5.6/

php@5.6 is keg-only, which means it was not symlinked into /usr/local, because this is an alternate version of another formula.

If you need to have php@5.6 first in your PATH run:

```
echo 'export PATH="/usr/local/opt/php@5.6/bin:$PATH"' >> ~/.bash_profile
echo 'export PATH="/usr/local/opt/php@5.6/sbin:$PATH"' >> ~/.bash_profile
```

For compilers to find php@5.6 you may need to set:

```
export LDFLAGS="-L/usr/local/opt/php@5.6/lib"
export CPPFLAGS="-I/usr/local/opt/php@5.6/include"
```


To have launchd start exolnet/deprecated/php@5.6 now and restart at login:

```
brew services start exolnet/deprecated/php@5.6
```

Or, if you don't want/need a background service you can just run:

```
php-fpm
```

==> Summary

 /usr/local/Cellar/php@5.6/5.6.40: 494 files, 60.5MB

==> Caveats

==> php@5.6

To enable PHP in Apache add the following to httpd.conf and restart Apache:

```
LoadModule php5_module /usr/local/opt/php@5.6/lib/httpd/modules/libphp5.so
```

```
<FilesMatch \.php$>
```

```
SetHandler application/x-httpd-php
```

```
</FilesMatch>
```

Finally, check DirectoryIndex includes index.php

```
DirectoryIndex index.php index.html
```

The php.ini and php-fpm.ini file can be found in:

```
/usr/local/etc/php/5.6/
```

php@5.6 is keg-only, which means it was not symlinked into /usr/local, because this is an alternate version of another formula.



If you need to have php@5.6 first in your PATH run:

```
echo 'export PATH="/usr/local/opt/php@5.6/bin:$PATH"' >> ~/.bash_profile
echo 'export PATH="/usr/local/opt/php@5.6/sbin:$PATH"' >> ~/.bash_profile
```

For compilers to find php@5.6 you may need to set:

```
export LDFLAGS="-L/usr/local/opt/php@5.6/lib"
export CPPFLAGS="-I/usr/local/opt/php@5.6/include"
```

To have launchd start exolnet/deprecated/php@5.6 now and restart at login:

```
brew services start exolnet/deprecated/php@5.6
```

Or, if you don't want/need a background service you can just run:

```
php-fpm
```

```
$ brew install php@7.1
```

```
==> Downloading https://homebrew.bintray.com/bottles/php@7.1-7.1.25.mojave.bottle.tar.gz
```

```
##### 100.0%
```

```
==> Pouring php@7.1-7.1.25.mojave.bottle.tar.gz
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set php_ini /usr/local/etc/php/7.1/php.ini system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set php_dir /usr/local/share/pear@7.1 system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set doc_dir /usr/local/share/pear@7.1/doc system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set ext_dir /usr/local/lib/php/pecl/20160303 system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set bin_dir /usr/local/opt/php@7.1/bin system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set data_dir /usr/local/share/pear@7.1/data system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set cfg_dir /usr/local/share/pear@7.1/cfg system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set www_dir /usr/local/share/pear@7.1/htdocs system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set man_dir /usr/local/share/man system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set test_dir /usr/local/share/pear@7.1/test system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear config-set php_bin /usr/local/opt/php@7.1/bin/php system
```

```
==> /usr/local/Cellar/php@7.1/7.1.25/bin/pear update-channels
```

```
==> Caveats
```

To enable PHP in Apache add the following to httpd.conf and restart Apache:

```
LoadModule php7_module /usr/local/opt/php@7.1/lib/httpd/modules/libphp7.so
```

```
<FilesMatch \.php$>
SetHandler application/x-httpd-php
</FilesMatch>
```


Finally, check DirectoryIndex includes index.php  
DirectoryIndex index.php index.html

The php.ini and php-fpm.ini file can be found in:  
/usr/local/etc/php/7.1/

php@7.1 is keg-only, which means it was not symlinked into /usr/local,  
because this is an alternate version of another formula.

If you need to have php@7.1 first in your PATH run:  
echo 'export PATH="/usr/local/opt/php@7.1/bin:\$PATH"' >> ~/.bash\_profile  
echo 'export PATH="/usr/local/opt/php@7.1/sbin:\$PATH"' >> ~/.bash\_profile

For compilers to find php@7.1 you may need to set:  
export LDFLAGS="-L/usr/local/opt/php@7.1/lib"  
export CPPFLAGS="-I/usr/local/opt/php@7.1/include"

To have launchd start php@7.1 now and restart at login:  
brew services start php@7.1  
Or, if you don't want/need a background service you can just run:  
php-fpm  
==> Summary  
 /usr/local/Cellar/php@7.1/7.1.25: 513 files, 63.2MB

```
$ brew install php@7.2
==> Downloading https://homebrew.bintray.com/bottles/php@7.2-7.2.14.mojave.bottle.tar.gz
```

```
##### 100.0%
==> Pouring php@7.2-7.2.14.mojave.bottle.tar.gz
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set php_ini /usr/local/etc/php/7.2/php.ini system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set php_dir /usr/local/share/pear@7.2 system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set doc_dir /usr/local/share/pear@7.2/doc system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set ext_dir /usr/local/lib/php/pecl/20170718 system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set bin_dir /usr/local/opt/php@7.2/bin system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set data_dir /usr/local/share/pear@7.2/data system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set cfg_dir /usr/local/share/pear@7.2/cfg system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set www_dir /usr/local/share/pear@7.2/htdocs system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set man_dir /usr/local/share/man system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set test_dir /usr/local/share/pear@7.2/test system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear config-set php_bin /usr/local/opt/php@7.2/bin/php system
==> /usr/local/Cellar/php@7.2/7.2.14/bin/pear update-channels
==> Caveats
```

To enable PHP in Apache add the following to httpd.conf and restart Apache:

```
LoadModule php7_module /usr/local/opt/php@7.2/lib/httpd/modules/libphp7.so
```

```
<FilesMatch \.php$>
SetHandler application/x-httpd-php
</FilesMatch>
```

Finally, check DirectoryIndex includes index.php  
 DirectoryIndex index.php index.html

The php.ini and php-fpm.ini file can be found in:  
 /usr/local/etc/php/7.2/

php@7.2 is keg-only, which means it was not symlinked into /usr/local,  
 because this is an alternate version of another formula.


If you need to have php@7.2 first in your PATH run:  
 echo 'export PATH="/usr/local/opt/php@7.2/bin:\$PATH"' >> ~/.bash\_profile  
 echo 'export PATH="/usr/local/opt/php@7.2/sbin:\$PATH"' >> ~/.bash\_profile

For compilers to find php@7.2 you may need to set:  
export LDFLAGS="-L/usr/local/opt/php@7.2/lib"  
export CPPFLAGS="-I/usr/local/opt/php@7.2/include"

To have launchd start php@7.2 now and restart at login:  
brew services start php@7.2

Or, if you don't want/need a background service you can just run:

php-fpm  
==> Summary

 /usr/local/Cellar/php@7.2/7.2.14: 514 files, 75.0MB

### \$ brew install php@7.3

==> Downloading <https://homebrew.bintray.com/bottles/php-7.3.1.mojave.bottle.tar.gz>

Already downloaded:

/Users/epallier/Library/Caches/Homebrew/downloads/012e37ae0f9b4dc0cbd5faa3e3fab9b71347dde138d38947819fe332b11fff0--php-7.3.1.mojave.bottle.tar.gz

==> Pouring php-7.3.1.mojave.bottle.tar.gz

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set php\_ini /usr/local/etc/php/7.3/php.ini system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set php\_dir /usr/local/share/pear system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set doc\_dir /usr/local/share/pear/doc system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set ext\_dir /usr/local/lib/php/pecl/20180731 system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set bin\_dir /usr/local/opt/php/bin system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set data\_dir /usr/local/share/pear/data system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set cfg\_dir /usr/local/share/pear/cfg system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set www\_dir /usr/local/share/pear/htdocs system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set man\_dir /usr/local/share/man system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set test\_dir /usr/local/share/pear/test system

==> /usr/local/Cellar/php/7.3.1/bin/pear config-set php\_bin /usr/local/opt/php/bin/php system

==> /usr/local/Cellar/php/7.3.1/bin/pear update-channels

==> Caveats

To enable PHP in Apache add the following to httpd.conf and restart Apache:

```
LoadModule php7_module /usr/local/opt/php/lib/httpd/modules/libphp7.so
```

```
<FilesMatch \.php$>
```

```
SetHandler application/x-httpd-php
```

</FilesMatch>

Finally, check DirectoryIndex includes index.php  
DirectoryIndex index.php index.html

The php.ini and php-fpm.ini file can be found in:  
/usr/local/etc/php/7.3/


To have launchd start php now and restart at login:

```
brew services start php
```

Or, if you don't want/need a background service you can just run:

```
php-fpm
```

==> Summary

 /usr/local/Cellar/php/7.3.1: 521 files, 76.4MB

Also, you may have the need to tweak configuration settings of PHP to your needs. A common thing to change is the memory setting, or the date.timezone configuration. The php.ini files for each version of PHP are located in the following directories:

```
/usr/local/etc/php/5.6/php.ini
```

```
/usr/local/etc/php/7.0/php.ini
```

```
/usr/local/etc/php/7.1/php.ini
```

```
/usr/local/etc/php/7.2/php.ini
```

```
/usr/local/etc/php/7.3/php.ini
```

Recopie de mes anciens php.ini :

```
cp -p /tmp/oldbrew_php/5.6/php.ini /usr/local/etc/php/5.6/
```

Pour toutes les versions 7, j'ai laissé le php.ini par défaut intact, sans modif

Let's switch back to the first PHP version now:

```
$ brew unlink php@7.3 && brew link --force --overwrite php@5.6
```

```
Unlinking /usr/local/Cellar/php/7.3.1... 24 symlinks removed
```

```
Linking /usr/local/Cellar/php@5.6/5.6.40... 25 symlinks created
```

If you need to have this software first in your PATH instead consider running:  
echo 'export PATH="/usr/local/opt/php@5.6/bin:\$PATH"' >> ~/.bash\_profile  
echo 'export PATH="/usr/local/opt/php@5.6/sbin:\$PATH"' >> ~/.bash\_profile

At this point, I strongly recommend closing ALL your terminal tabs and windows. This will mean opening a new terminal to continue with the next step. This is strongly recommended because some really strange path issues can arise with existing terminals (trust me, I have seen it!).

### **\$ php -v**

```
PHP 5.6.40 (cli) (built: Jan 16 2019 14:53:29)
Copyright (c) 1997-2016 The PHP Group
Zend Engine v2.6.0, Copyright (c) 1998-2016 Zend Technologies
    with Zend OPcache v7.0.6-dev, Copyright (c) 1999-2016, by Zend Technologies
```

\*\*\*\*\*

## **2.2 - Apache PHP Setup - Part 1**

\*\*\*\*\*

If you have been following this guide correctly, the last entry should be your mod\_rewrite module:

```
LoadModule rewrite_module lib/httpd/modules/mod_rewrite.so
```

Below this add the following libphp modules:

```
LoadModule php5_module /usr/local/opt/php@5.6/lib/httpd/modules/libphp5.so
#LoadModule php7_module /usr/local/opt/php@7.0/lib/httpd/modules/libphp7.so
#LoadModule php7_module /usr/local/opt/php@7.1/lib/httpd/modules/libphp7.so
#LoadModule php7_module /usr/local/opt/php@7.2/lib/httpd/modules/libphp7.so
#LoadModule php7_module /usr/local/opt/php@7.3/lib/httpd/modules/libphp7.so
```

Also you must set the Directory Indexes for PHP explicitly, so search for this block:

```
<IfModule dir_module>
    DirectoryIndex index.html
</IfModule>
```

and replace it with this:

```
<IfModule dir_module>
```

```
    DirectoryIndex index.php index.html
</IfModule>
```

```
<FilesMatch \.php$>
    SetHandler application/x-httpd-php
</FilesMatch>
```

```
$ sudo apachectl -k stop
$ sudo apachectl start
```

Simply create a file called info.php in your Sites/ folder you created earlier with this one-liner.

```
echo "<?php phpinfo();" > /usr/local/var/www
(ou echo "<?php phpinfo();" > ~/Sites/info.php)
```

Point your browser to <http://localhost/info.php> and you should see a shiny PHP information page:

You can test the other PHP versions by commenting the LoadModule ... php@5.6 ... entry and uncommenting one of the other ones. Then simply restart apache and reload the same page.

We hard-coded Apache to use PHP 5.6, but we really want to be able to switch between versions. Luckily, some industrious individuals have already done the hard work for us and written a very handy little PHP switcher script.

We will install the sphp script into brew's standard /usr/local/bin:

```
$ curl -L https://gist.githubusercontent.com/rhukster/f4c04f1bf59e0b74e335ee5d186a98e2/raw > /usr/local/bin/sphp
$ chmod +x /usr/local/bin/sphp
```

Homebrew should have added its preferred /usr/local/bin and /usr/local/sbin to your path as part of its installation process. Quickly test this by typing:

```
$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

After you have completed these steps, you should be able to switch your PHP version by using the command sphp followed by a two digit value for the PHP version:

```
$ sphp 7.1
Switching to php@7.1
```

Switching your shell

Unlinking /usr/local/Cellar/php@5.6/5.6.40... 25 symlinks removed

Unlinking /usr/local/Cellar/php@7.1/7.1.25... 0 symlinks removed

Unlinking /usr/local/Cellar/php@7.2/7.2.14... 0 symlinks removed

Unlinking /usr/local/Cellar/php/7.3.1... 0 symlinks removed

Linking /usr/local/Cellar/php@7.1/7.1.25... 25 symlinks created

If you need to have this software first in your PATH instead consider running:

```
echo 'export PATH="/usr/local/opt/php@7.1/bin:$PATH"' >> ~/.bash_profile
```

```
echo 'export PATH="/usr/local/opt/php@7.1/sbin:$PATH"' >> ~/.bash_profile
```

You will need sudo power from now on

Switching your apache conf

Restarting apache

PHP 7.1.25 (cli) (built: Dec 7 2018 08:20:45) ( NTS )

Copyright (c) 1997-2018 The PHP Group

Zend Engine v3.1.0, Copyright (c) 1998-2018 Zend Technologies

with Zend OPcache v7.1.25, Copyright (c) 1999-2018, by Zend Technologies

All done!

You will need to switch to each of your installed PHP versions and run update again to get updates for each PHP version and ensure you are running the version of PHP you intend.

Je l'ai fait pour 5.6 et 7.3 :

```
$ sphp 5.6
```

```
$ brew update
```

This will spit out a list of available updates, and any deleted formulas.

To upgrade the packages simply type:

```
$ brew upgrade
```

**\$ sphp 7.3**

Switching to php@7.3

Switching your shell

Unlinking /usr/local/Cellar/php@5.6/5.6.40... 25 symlinks removed



```
Unlinking /usr/local/Cellar/php@7.1/7.1.25... 0 symlinks removed
Unlinking /usr/local/Cellar/php@7.2/7.2.14... 0 symlinks removed
Unlinking /usr/local/Cellar/php/7.3.1... 0 symlinks removed
Linking /usr/local/Cellar/php/7.3.1... 24 symlinks created
You will need sudo power from now on
Switching your apache conf
Restarting apache
```

```
PHP 7.3.1 (cli) (built: Jan 10 2019 13:15:37) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.1, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.3.1, Copyright (c) 1999-2018, by Zend Technologies
$ brew update
$ brew upgrade
```

You can see the specific versions of a PHP package by typing:

```
$ brew info php@7.3
```

```
php: stable 7.3.1 (bottled)
General-purpose scripting language
https://secure.php.net/
/usr/local/Cellar/php/7.3.1 (521 files, 76.4MB) *
  Poured from bottle on 2019-01-22 at 12:25:39
From: https://github.com/Homebrew/homebrew-core/blob/master/Formula/php.rb
==> Dependencies
Build: httpd ✓, pkg-config ✓
Required: apr ✓, apr-util ✓, argon2 ✓, aspell ✓, autoconf ✓, curl-openssl ✓, freetds ✓, freetype ✓, gettext ✓, glib ✓, gmp ✓, icu4c ✓, jpeg ✓,
libpng ✓, libpq ✓, libsodium ✓, libzip ✓, openldap ✓, openssl ✓, pcre2 ✓, sqlite ✓, tidy-html5 ✓, unixodbc ✓, webp ✓
==> Caveats
To enable PHP in Apache add the following to httpd.conf and restart Apache:
  LoadModule php7_module /usr/local/opt/php/lib/httpd/modules/libphp7.so
```

```
<FilesMatch \.php$>
```

```
SetHandler application/x-httpd-php
</FilesMatch>
```

Finally, check DirectoryIndex includes index.php  
DirectoryIndex index.php index.html

The php.ini and php-fpm.ini file can be found in:  
/usr/local/etc/php/7.3/

To have launchd start php now and restart at login:  
brew services start php

Or, if you don't want/need a background service you can just run:  
php-fpm

==> Analytics

install: 37,504 (30 days), 123,299 (90 days), 417,557 (365 days)

install\_on\_request: 33,685 (30 days), 111,773 (90 days), 376,762 (365 days)

build\_error: 0 (30 days)

```
$ brew services list
```

```
Name Status User Plist
httpd started root /Library/LaunchDaemons/homebrew.mxcl.httpd.plist
mysql started epallier /Users/epallier/Library/LaunchAgents/homebrew.mxcl.mysql.plist
php stopped
php@5.6 stopped
php@7.1 stopped
php@7.2 stopped
```

```
$ mysql --version
```

```
mysql Ver 8.0.13 for osx10.14 on x86_64 (Homebrew)
```

```
*****
```

## Part 2: macOS 10.14 Mojave Web Development Environment

\*\*\*\*\*

<https://getgrav.org/blog/mac-os-mojave-apache-mysql-vhost-apc>

\*\*\*\*\*

## 2.0.4. MYSQL => MARIADB

\*\*\*\*\*

(mysql ne semble plus marcher sur mon mac, donc j'installe mariadb)

In the original guide, we used the Oracle MySQL installation package. However, we now have switched to MariaDB which is a drop-in replacement for MySQL and is easily installed and updated with Brew. Detailed information on the HomeBrew installation process can be found on the [mariadb.org](http://mariadb.org) site but the essentials are as follows.

```
$ brew uninstall --force mysql
```

```
$ brew doctor
```

```
$ brew update
```

```
$ brew install mariadb
```

```
==> Downloading https://homebrew.bintray.com/bottles/mariadb-10.3.12.mojave.bottle.tar.gz
```

```
##### 100.0%
```

```
==> Pouring mariadb-10.3.12.mojave.bottle.tar.gz
```

```
==> Caveats
```

```
A "/etc/my.cnf" from another install may interfere with a Homebrew-built server starting up correctly.
```

MySQL is configured to only allow connections from localhost by default

To connect:

```
mysql -uroot
```

To have launchd start mariadb now and restart at login:

```
brew services start mariadb
```

Or, if you don't want/need a background service you can just run:

```
mysql.server start
```

==> Summary

 /usr/local/Cellar/mariadb/10.3.12: 658 files, 174.4MB

macp1219:install epallier\$ brew services start mariadb

==> Successfully started `mariadb` (label: homebrew.mxcl.mariadb)

macp1219:install epallier\$ mysql -uroot

After a successful installation, you can start the server to ensure it autostarts in the future with:

**\$ brew services start mariadb**

You should get some positive feedback on that action:

==> Successfully started `mariadb` (label: homebrew.mxcl.mariadb)

(PAS FAIT)

[Download SequelPro](#) and install it. (it's awesome and free!). You should be able to automatically create a new connection via the Socket option without changing any settings.

(PAS FAIT, je garde "no password")

It is advisable to change the **MySQL server password** and secure your installation. The simplest way to do this is to use the provided script:

`$ /usr/local/bin/mysql_secure_installation`

Just answer the questions and fill them in as is appropriate for your environment.

If you need to stop the server, you can use the simple command:

**\$ brew services stop mariadb**

## 2.0.5. Apache Virtual Hosts

Apache already comes preconfigured to support this behavior but it is not enabled. First you will need to uncomment the following lines in your `/usr/local/etc/httpd/httpd.conf` file:

```
LoadModule vhost_alias_module lib/httpd/modules/mod_vhost_alias.so
```

and:

```
# Virtual hosts
```

```
Include /usr/local/etc/httpd/extra/httpd-vhosts.conf
```

Then you can edit this referenced file and configure it to your needs:

```
$ vi /usr/local/etc/httpd/extra/httpd-vhosts.conf
```

When you set up virtual hosts, you will lose your older document root, so you will need to add back support for that first as a virtual host.

```
<VirtualHost *:80>
```

```
    DocumentRoot "/Users/your_user/Sites"
```

```
    ServerName localhost
```

```
</VirtualHost>
```

```
<VirtualHost *:80>
```

```
    DocumentRoot "/Users/your_user/Sites/grav-admin"
```

```
    ServerName grav-admin.test
```

```
</VirtualHost>
```

Dnsmasq (PAS FAIT car j'utilise plutot /etc/hosts)

We used to recommend using .dev domain name, but since Chrome 63 forces all .dev domains to use SSL, this guide has been updated to use .test In the example virtualhost we setup above, we defined a ServerName of grav-admin.test. This by default will not resolve to your local machine, but it's often very useful to be able to setup various virtual hosts for development purposes. You can do this by manually adding entries to /etc/hosts every time, or you can install and configure Dnsmasq to automatically handle wildcard \*.test names and forward all of them to localhost (127.0.0.1).

```
$ cat /etc/hosts
```

```
##
```

```
# Host Database
```

```
#
```

```
# localhost is used to configure the loopback interface
```

```
# when the system is booting. Do not change this entry.
```

```
##
```

```
127.0.0.1 localhost
```

```
255.255.255.255 broadcasthost
```

```
::1 localhost
```

```
127.0.0.1 labinvent.test
```

```
$ ping labinvent.test
```

APC Cache (pas fait)

Caching in PHP is a big part of the performance equation. There are two types of caching typically available, and both have a big impact on speed and performance.

The first type of cache is called an opcode cache, and this is what takes your PHP script and compiles it for faster execution. This alone can typically result in a 3X speed increase!.

The second type of cache is a user cache, and this is a data-store that PHP can use to quickly store and retrieve data from. These typically run in memory which means they are transient, but very fast.

All PHP packages now come pre-built with Zend OPcache by default, but you can still install APCu Cache as a data store.

Xdebug (pas fait)

One of the most important aspects of any kind of development is the ability to debug and fix your code. PHP comes with limited support to dump variables or log to a file, but for more complex situations you need something more powerful.

Xdebug provides is a debugging and profiling extension for PHP that provides an HTML-friendly output for the `var_dump()` method that improves the readability of the default version. It also provides other useful dumping methods as well as displaying stack traces. One of the best features however, is the ability to remote debug your code. This means you can set breakpoints, and step through your PHP code inspecting as you go. Full documentation on Xdebug contains extensive information about all the functionality available. Let's switch back to PHP 5.6 and get started.

NOTE: The brew installation actually creates configuration files in `/usr/local/etc/php/5.6/conf.d`, `/usr/local/etc/php/7.0/conf.d`, `/usr/local/etc/php/7.1/conf.d`, `/usr/local/etc/php/7.2/conf.d`, and `/usr/local/etc/php/7.3/conf.d` respectively. If you want to uninstall a PHP extension, simply rename the `.ini` file to `.ini.bak` and restart apache. Alternatively, you can simply use brew to uninstall it, and reinstall it again when you need it.

## 2.0.6. PHPMYADMIN

```
$ brew install phpmyadmin
```

```
Updating Homebrew...
```

```
==> Auto-updated Homebrew!
```

```
Updated 1 tap (homebrew/core).
```

```
==> Updated Formulae
```

```
graph-tool
```

```
librsvg
```

```
siril
```

==> Downloading <https://files.phpmyadmin.net/phpMyAdmin/4.8.4/phpMyAdmin-4.8.4-all-languages.tar.gz>

Already downloaded:

/Users/epallier/Library/Caches/Homebrew/downloads/d0266b8cf3a58a0c144a4b2fe8723d6fe8e854e1b6c81f6e4667236ee5875e76--phpMyAdmin-4.8.4-all-languages.tar.gz

==> Caveats


To enable phpMyAdmin in Apache, add the following to httpd.conf and restart Apache:

```
Alias /phpmyadmin /usr/local/share/phpmyadmin
<Directory /usr/local/share/phpmyadmin/>
Options Indexes FollowSymLinks MultiViews
AllowOverride All
<IfModule mod_authz_core.c>
Require all granted
</IfModule>
<IfModule !mod_authz_core.c>
Order allow,deny
Allow from all
</IfModule>
</Directory>
```

Then open <http://localhost/phpmyadmin>

The configuration file is **/usr/local/etc/phpmyadmin.config.inc.php**

==> Summary

 /usr/local/Cellar/phpmyadmin/4.8.4: 2,434 files, 32.3MB, built in 7 seconds

**Config dans /usr/local/etc/phpmyadmin.config.inc.php pour connexion avec “root” sans pass :**

```
//$cfg['Servers'][$i]['host'] = 'localhost';
```

```
$cfg['Servers'][$i]['host'] = '127.0.0.1';
```

```
---
```

```
//$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

```
$cfg['Servers'][$i]['AllowNoPassword'] = true;
```

```
---
```

```
$cfg['Servers'][$i]['controluser'] = 'root';
```

```
$cfg['Servers'][$i]['controlpass'] = '';
```

## 2.0.7. Installation de l'extension YAML

```
$ php -v  
=> 7.3
```

```
$ brew install libyaml  
$ pecl install yaml  
=> ça plante
```

Enlever l'extension du fichier php.ini et la mettre dans conf.d/ :

```
$ vi /usr/local/etc/php/7.3/php.ini  
;extension="yaml.so"  
$ vi /usr/local/etc/php/7.3/conf.d/ext-yaml.ini  
[yaml]  
;extension=/usr/local/Cellar/php/7.3.1/pecl/20180731/yaml.so  
extension=yaml.so
```

J'ai supprimé un vieux lien qui ne servait à rien et gênait l'installation qui voulait créer un dossier pecl/ à la place :

```
$ rm /usr/local/Cellar/php/7.3.1/pecl
```

Je recommence l'install :

```
$ pecl install yaml  
=> OK
```

```
$ php -v  
=> l'extension n'est pas trouvée
```

Il manque pour ça un lien vers pecl/ :

```
$ cd /usr/local/lib/php  
$ ln -s ../../Cellar/php/7.3.1/pecl/
```



```
$ php -v  
=> OK
```

Pour que ça marche depuis le web :

```
$ sudo apachectl -k restart
```

Pour l'installer sur une autre version de php, par exemple 5.6 :

```
$ sphp 5.6  
$ pecl uninstall -r yam1  
$ pecl install yam1
```

```
$ cd /usr/local/etc/php/7.3/conf.d/  
cp ext-yam1.ini ../../5.6/conf.d  
cp ext-yam1.ini ../../7.2/conf.d
```

# 3. Installation des pré-requis sur Windows 10

## 3.1. Installation directement sur Windows 10 (TODO)

TODO:

Installer WAMP server, ou bien XAMPP.

Il faudra aussi convertir le script d'installation bash install/installation.sh en PHP.

**Un volontaire ?**

## 3.2. Installation sur une VM linux (avec VirtualBox)

*(updated 17/1/20 - EP)*

*(testé le 17/1/20 sur Mac OS X 10.14 avec VB 6.1.2 et une VM Ubuntu 18.04.3 LTS)*

(

- Autre option possible : installer linux en dual-boot :

<https://lecrabeinfo.net/installer-ubuntu-18-04-lts-dual-boot-windows-10.html#etape-4-installer-ubuntu-1804-lts>

- Autre VM disponible, CentOS:

<https://pixelabs.fr/machine-virtuelle-centos-7-virtualbox/>

)

Dans le détail, suivre ce tuto:

<https://lecrabeinfo.net/virtualbox-installer-windows-linux-dans-une-machine-virtuelle.html>

(autre méthode possible :

<https://openclassrooms.com/fr/courses/43538-reprenez-le-contrôle-a-l'aide-de-linux/37630-installez-linux-dans-une-machine-virtuelle>)

En résumé (avec quelques remarques pertinentes absentes du tuto) :

- **Télécharger une image iso Linux (ici Ubuntu 18 LTS)**

<https://ubuntu.com/#download>

- **Installer VirtualBox (VB)**

<https://www.virtualbox.org/wiki/Downloads>

- **Ajouter le pack extension de VirtualBox pour bénéficier du plein écran et du copier-coller**

Télécharger ce pack (on trouve le lien de téléchargement sur la même page)

Démarrer VB

Cliquer sur Paramètres

Cliquer sur Extensions

Cliquer sur “+” et sélectionner le fichier téléchargé, puis sur Installation

- **Créer une machine virtuelle (VM) à partir de VB**

Cliquez sur le bouton **Nouvelle**

Donnez un **nom** à la machine virtuelle (ex. : ubuntu18), puis sélectionnez le **type** “linux” et la **version** “ubuntu” du système d’exploitation.

Sélectionnez la **quantité de mémoire vive** à attribuer la VM

Si vous avez assez de mémoire vive et pour plus de confort, je vous conseille de choisir **2048 Mo** (2 Go). Faites attention à laisser assez de mémoire vive à la machine hôte.

Ajoutez un **disque virtuel**. Il sera enregistré sous la forme d'un fichier disque (ex. : « ubuntu18.vdi ») sur votre PC. VirtualBox affiche la **taille recommandée** pour le disque virtuel en fonction du système d'exploitation choisi précédemment. Sélectionnez **Créer un disque dur virtuel maintenant** puis cliquez sur **Créer**.

Type de fichier pour le disque virtuel : si vous n'utilisez que VirtualBox comme logiciel de virtualisation, laissez le choix sur VDI (sélectionnez un autre format si vous souhaitez une compatibilité avec un autre logiciel de virtualisation, VHD est le format utilisé par Windows Virtual PC, le format VMDK par VMWare).

**Type de stockage** sur le disque de votre PC : **Dynamiquement alloué** (le fichier .vdi représentant le disque virtuel grossira automatiquement en fonction de la quantité de données que vous enregistrerez dessus dans la machine virtuelle. Si vous choisissez une taille de disque de 50 Go, le fichier .vdi représentant le disque virtuel occupera 0 Go sur votre PC, puis 10 Go après l'installation de Windows, puis 11 Go après l'installation de plusieurs logiciels, etc).

Pour terminer, choisissez la **taille** du disque virtuel puis cliquez sur **Créer**.

*(pour moi, c'est créé dans ~/VirtualBox VMs/ubuntu18/ubuntu18.vdi)*

Pour démarrer la VM, sélectionnez-la puis cliquez sur le bouton **Démarrer**.

Disque de démarrage : pour pouvoir installer notre système d'exploitation dans la machine virtuelle, il faut amorcer celle-ci à partir d'une disque de démarrage, l'image ISO du système d'exploitation Ubuntu18 à installer, puis cliquez sur **Démarrer**.

Votre machine virtuelle démarre **à partir de l'image ISO**, il ne vous reste plus qu'à installer le système d'exploitation en suivant les instructions de l'installateur.

- **Installer Ubuntu dans la VM**

Cliquer sur "Install Ubuntu"

Cliquer sur "Français" et "Installer Ubuntu"

Cliquer sur "Installation normale" et "Installer un logiciel tiers pour le matériel graphique et wifi..."

Cliquer sur "Effacer le disque et installer Ubuntu"

J'ai nommé la machine "vm-labinvent-ub18", user "Etienne Pallier" (login="labinv", pass="labinv")

- **Configurer la VM**

**Réseau** : vérifier qu'il est bien configuré en **NAT** (c'est ce qui est fait par défaut)

**Pour le copier-coller** : Menu Périphériques → Presse-Papier Partagé, choisir Bidirectionnel

(ou Devices/Shared Clipboard/Bidirectional)

**Pour le Plein écran :**

- menu périphériques, insérer CD extension => fait le lien vers le package extension VirtualBox (valider par oui...)
- menu View/Virtual screen 1/ choisir une résolution
- Sur Win10 : CTRL-F pour passer en plein écran, CTRL-C pour revenir en petit écran (touche CTRL de droite)
- Sur Mac : Cmd-F et Cmd-C (touche Cmd de gauche)

- **Maintenant, il ne reste qu'à procéder à l'installation de LabInvent** comme si vous étiez sur un pc linux !

Aller au début de cette doc, dans le chapitre "Installation des pré-requis", dans la section "Installation sur linux", suivre l'exemple pour Ubuntu "[2.2.3.1. Exemple pour une distribution UBuntu 18.04.3 LTS \(fait en 2020 01\) \(stagiaire Jeanne Prugniel\)](#)".



# 4. Installation et configuration du logiciel depuis l'IDE Eclipse

0) Installer Eclipse (si nécessaire)

<https://www.eclipse.org/downloads/>

**0) Installer Eclipse (Si nécessaire), voire Java**

En effet, même si la version que vous allez installer est une version pour PHP, Eclipse a besoin de Java pour pouvoir s'exécuter.

Vérifiez que Java soit bien installé sur votre système :

```
java -version
```

Si ce n'est pas le cas, exécutez les lignes suivantes :

```
sudo apt-add-repository ppa:webupd8team/java
```

```
sudo apt-get update
```

```
sudo apt-get install oracle-java8-installer
```

Pour installer Eclipse, allez sur : <http://www.eclipse.org/downloads/packages/release/Neon/3>

Sélectionnez la version que vous désirez (Neon, Oxygen, Mars ...) puis sélectionnez "Eclipse for PHP Developers".

Téléchargez la version correspondant à votre système d'exploitation.

Placez-vous dans le dossier où vous voulez installer Eclipse avec l'archive précédemment téléchargée (renommez-la en eclipse.tar.gz), puis exécutez la commande suivante :

```
tar -zxvf eclipse.tar.gz
```

Si vous voulez Eclipse dans le menu des applications sous Ubuntu, il vous faudra créer le fichier eclipse.desktop :

```
gksudo gedit /usr/share/applications/eclipse.desktop
```

puis collez-y ce qui suit, en modifiant le chemin des lignes Exec et Icon au besoin :

```
[Desktop Entry]
```

```
Name=Eclipse
```

```
Type=Application
```

```
Exec=/opt/eclipse/eclipse
```

```
Terminal=false
```

```
Icon=/opt/eclipse/icon.xpm
```

```
Comment=Integrated Development Environment
```

```
NoDisplay=false
```

```
Categories=Development;IDE
```

```
Name[en]=eclipse.desktop
```



Puis donnez les droits à tous les utilisateurs sur ce fichier :

```
sudo chmod a+r /usr/share/applications/eclipse.desktop
```

## **1) Désactiver la vérification du certificat**

Window -> Preferences -> Team -> git -> configuration -> Add entry

Key = http.sslVerify

Value = false

## **2) Récupérer le projet**

**(Si le projet git n'existe pas déjà sur votre machine)**

File/Import project from git

Select repository source: Clone URI: <https://gitlab.irap.omp.eu/epallier/labinvent.git>

Directory:

- par défaut, il propose : /Users/epallier/git/labinvent

- mais on peut le mettre n'importe où ailleurs,

par exemple, on pourrait le mettre directement dans le repertoire web de Apache:

/Applications/XAMPP/xamppfiles/htdocs

(si on veut que le projet s'exécute directement dans le dossier web apache htdocs, mais ca n'est pas obligatoire...)

initial branch: master

remote name: origin

Import as general project

Project name: LABINVENT

**(si le projet git existe déjà sur votre machine)**

File/Import project from git

Existing local repository

Directory

ADD => Sélectionnez le dossier contenant le projet git => Finish => Next

Import existing Eclipse project

Sélectionner le projet => Search for nested project => finish

### **3) Configurer le projet**

a) S'assurer que le projet est bien reconnu comme un projet PHP (il doit y avoir un petit "P" sur le dossier racine du projet)

Si ça n'est pas le cas, vérifier que le fichier .project (à la racine) contient bien

<natures>

<nature>org.eclipse.php.core.PHPNature</nature>

</natures>

NB : Le fichier .project est normalement versionné et donc le projet labinvent devrait être reconnu automatiquement comme projet PHP

b) S'assurer que les fichiers de vue de cakephp ("\*.ctp") sont bien reconnus comme des fichiers PHP.

Pour tester cela, ouvrir le fichier de vue cakephp/app/View/Categories/get\_all.ctp

Si ce fichier s'ouvre comme un simple fichier texte, c'est qu'il n'est pas reconnu par Eclipse comme un fichier Php.

Il faut donc associer l'éditeur Php à l'extension de fichier "\*.ctp" :

- Preferences/General/Content types
- Dans la liste "Content types", ouvrir la section "Text", sélectionner PHP
- Ajouter l'extension "\*.ctp"

c) Vérifier la version de php utilisée (il serait préférable d'utiliser la même version que celle officiellement utilisée par le logiciel, c'est à dire php 5.6, mais attention, le serveur IRAP utilise toujours une version 5.3 pour inventirap) :

- Clic-droit sur le projet, Propriétés
- PHP

- Interpreter
- Enable project specific settings, PHP Version : "PHP 5.6"

d) S'assurer que le texte est bien encodé en UTF-8 par défaut :

clic-droit sur le dossier racine du projet (dans PHP Explorer), Properties, Resource : dans la zone "Text file encoding" cocher "Other" et sélectionner UTF-8

(

Il faudrait commiter ça mais je ne sais pas trop si c'est risqué ou pas.

Les fichiers concernés sont :

- .project (déjà versionné) : car il commence par la ligne ""
- mais c'est surtout celui-ci qui compte (actuellement ignoré de git) : .settings/org.eclipse.core.resources.prefs : car sa 2eme ligne est "encoding/<project>=UTF-8"

)

Les éléments suivants sont normalement DEJA ignorés par git, à vérifier :

- .settings/
- cakephp/app/tmp/ : tout sauf
- documents/

- cakephp/app/Config/ :
- database.php
- labinvent.php

---

\*\*\*\*\*

REMARQUES INTERRESSANTES (MAIS VOUS POUVEZ LES IGNORER)

// DEBUT DES REMARQUES

A la racine du projet, j'ai plusieurs éléments cachés de configuration Eclipse :

1) fichier .buildpath

Il est versionné puisque "svn status .buildpath" (depuis la console) ne donne rien

Il contient :

```
<?xml version="1.0" encoding="UTF-8"?>
<buildpath>
  <buildpathentry kind="con" path="org.eclipse.php.core.LANGUAGE"/>
  <buildpathentry kind="lib" path="docs/mockup/mockup_html.zip"/>
  <buildpathentry kind="src" path="cakephp"/>
</buildpath>
```

2) fichier .project

Il est déjà versionné

Il contient :

```
<?xml version="1.0" encoding="UTF-8"?>
<projectDescription>
```

```
<name>invirap</name>
<comment></comment>
<projects>
</projects>
<buildSpec>
  <buildCommand>
    <name>org.eclipse.wst.common.project.facet.core.builder</name>
    <arguments>
    </arguments>
  </buildCommand>
  <buildCommand>
    <name>org.eclipse.wst.validation.validationbuilder</name>
    <arguments>
    </arguments>
  </buildCommand>
  <buildCommand>
    <name>org.eclipse.dltk.core.scriptbuilder</name>
    <arguments>
    </arguments>
  </buildCommand>
</buildSpec>
<natures>
  <nature>org.eclipse.php.core.PHPNature</nature>
  <nature>org.eclipse.wst.common.project.facet.core.nature</nature>
</natures>
</projectDescription>
```

3) dossier .settings/ (exclus de svn)

Il contient 3 fichiers :

- org.eclipse.core.resources.prefs : bizarrement, il ne contient que quelques références seulement :

```
eclipse.preferences.version=1
```

```
encoding//cakephp/app/Controller/MaterielsController.php=UTF-8
```

```
encoding//cakephp/app/View/Elements/menu_view.ctp=UTF-8
```

```
encoding//cakephp/app/View/Layouts/default.ctp=UTF-8
```

```
encoding//cakephp/app/View/Materiels/index.ctp=UTF-8
```

```
encoding//cakephp/app/View/Materiels/scaffold.view.ctp=UTF-8
```

```
encoding//database/Upd_TableConstraints.sql=UTF-8
```

```
encoding//database/update/README.txt=UTF-8
```

```
encoding//docs/HOWTO.txt=UTF-8
```

```
encoding//install/HOWTO.txt=UTF-8
```

```
encoding/<project>=UTF-8
```

- org.eclipse.php.core.prefs

```
eclipse.preferences.version=1
```

```
include_path=0;/invirap\u00051;/invirap/docs/mockup/mockup_html.zip
```

- org.eclipse.wst.common.project.facet.core.xml : sans doute inutile ? (lié à "Faceted Project Validation Builder" dans Properties/Builders)

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<faceted-project>
```

```
  <fixed facet="php.core.component"/>
```

```
  <fixed facet="php.component"/>
```

```
  <installed facet="php.core.component" version="1"/>
```

```
  <installed facet="php.component" version="5.4"/>
```

```
</faceted-project>
```

```
// FIN DES REMARQUES
```



\*\*\*\*\*

#### 4) (TODO:) Set Code style

Window/Preferences : PHP / Editor

...

#### 5) (TODO:) virtualenv

Now, once the PHP5 virtual environment is installed (see above),  
set it in Eclipse as the project interpreter:

(cf <http://virtphp.org>)

...

#### 6) (TODO:) Test

## 7) Run

(EP 02/07/20)

### Créer une configuration d'exécution :

- Cliquer sur la racine du projet
- Run/Run configurations
- Indiquer où se trouve le fichier d'index de démarrage :
  - dans le panneau vertical gauche ouvrir la section "PHP Web Application", et cliquer sur "index"
  - section "PHP server" : cliquer sur "Configure..."
    - Base URL: <http://labinvent.devv> (remplacer par le nom de votre serveur virtuel, par exemple "<http://localhost:8080>" ou autre...)
  - section "File", cliquer sur "Browse" et sélectionner le fichier index.php qui est dans le dossier webroot (/labinvent2/webroot/index.php, remplacer labinvent2 par le nom de votre racine du projet)
  - section "URL" :
    - décocher "Auto Generate"
    - URL: <http://labinvent.devv/> puis /labinvent2/webroot/index.php
- **Cliquer sur Run pour démarrer sur la page de login**

## 5. UPDATE DE LA BRANCHE MASTER (dev assermenté only)

**Attention, cette opération ne doit être réalisée que par le responsable du projet LabInvent.**

Pour mettre à jour la branche "master" à partir de la branche "dev-IRAP" (ou "dev") :

Aller sur la branche master :

```
$ git checkout master
```

```
$ git branch
```

Mettre à jour la branche master :

```
$ git merge --no-ff dev-IRAP
```

```
$ git push
```

Revenir sur la branche dev-IRAP (ou dev)

```
$ git checkout dev-IRAP
```

## 6. Auto-génération du code (avec “bake”)

(fait début 2019)

Tentative d'auto-génération du code avec "cake bake", pour voir ce qui pourrait manquer au code actuel, pour y ajouter plus de cohérence et peut-être des nouveautés...

```
$ bin/cake bake model Materiels
```

```
Notice Error: Undefined index: Gestionnaires in vendor/cakephp/bake/src/View/Helper/DocBlockHelper.php, line 234
```

```
Notice Error: Undefined index: Photos in vendor/cakephp/bake/src/View/Helper/DocBlockHelper.php, line 234
```

```
Baking table class for Materiels...
```

```
Wrote src/Model/Table/MaterielsTable.php
```

```
Baking entity class for Materiel...
```

```
Wrote src/Model/Entity/Materiel.php
```

```
Baking test fixture for Materiels...
```

```
Wrote tests/Fixture/MaterielsFixture.php`
```

```
Baking test case for App\Model\Table\MaterielsTable ...
```

```
Wrote tests/TestCase/Model/Table/MaterielsTableTest.php`
```

## 7. METHODES ET CONCEPTS INTERESSANTS A UTILISER

Dans la vue Materiels/view, find, et index :

```
$materiel->has('designation') ? $materiel->designation : "
```

Dans le controleur AppController, définition de fonctions passées aux vues :

```
$displayElement = function ($nom, $valeur, $params = "") {
```

```
    $balise = ($params != "") ? '<td ' . $params . '>' : '<td>';
```

```
    // Ca c'est parce que sinon y'a au moins deux tests qui passent pas, a cause de l'espace dans la balise ...
```

```
    if ($valeur != "")
```

```

        echo '<tr><td><strong>' . $nom . ' </strong></td>' . $balise . $valeur . '</td></tr>';
    };
    $this->set('displayElement', $displayElement);

    $dateProchainControleVerif = function ($t) {
        $time = Time::now(); // On récupère la date et l'heure actuelles
        $today = new \DateTime((new date("$time->year-$time->month-$time->day"))->format('Y-m-d'));
        $time1 = new time($t);
        $dateTime1 = new \DateTime((new date("$time1->year-$time1->month-$time1->day"))->format('y-m-d'));
        $interval = ($today->diff($dateTime1));
        $strInterval = $interval->format('%a');
        return (int) $strInterval;
    };
    $this->set('dateProchainControleVerif', $dateProchainControleVerif);

```

⇒ utilisées dans la vue Suivis/view :

```

if ($dateProchainControleVerif($suivi->date_prochain_controle) < 0)
    $style = "color: red";
elseif ($dateProchainControleVerif($suivi->date_prochain_controle) <= 15)
    $style = "color: orange";
else $style = "color: green";

$displayElement(__('Materiel'), $suivi->has('materiel') ?
    $this->Html->link($suivi->materiel->designation, [
        'controller' => 'Materiels',
        'action' => 'view',
        $suivi->materiel->id

```

);  
D  
:  
"

## 8. Comment on a simulé un LDAP pour la connexion utilisateurs en mode LOCAL

Comment simuler un LDAP quand on est sur une machine de développement, et qu'on n'a pas de LDAP sous la main ?

Comment simuler un LDAP quand on est sur une machine de développement, et qu'on n'a pas de LDAP sous la main ?

1 - Créer une table fakeldap qui représentera le LDAP et contiendra quelques utilisateurs.

```
CREATE TABLE `fakeldap` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `sn` varchar(45) DEFAULT NULL,  
  `givenname` varchar(45) DEFAULT NULL,  
  `uid` varchar(45) DEFAULT NULL,  
  `mail` varchar(45) DEFAULT NULL,  
  `userpassword` varchar(255) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
-- ALTER TABLE `fakeldapusers`  
-- MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=73;
```

(...TO BE CONTINUED...)

COMMENT CONTOURNER LE LDAP officiel

\*\*\*\*\*

**(ATTENTION: CONTENU OBSOLETE A METTRE A JOUR (EP))**

(cf Controller/UtilisateursController.php : methode login())

1) Si on veut tester le LDAP, on peut utiliser le LDAP de la Virtual Machine Upsilon (CentOS 6)

Dans ce cas, il faut ajouter dans la BD les utilisateurs spécifiques suivants :

# Ajout d'utilisateurs de TEST (LDAP upsilon)

```
INSERT INTO utilisateurs (nom, login, email, role, groupes_metier_id) VALUES
    ('Hillebrand Cedric', 'Cedric', 'Cedric.Hillebrand@irap.omp.eu', 'Super Administrateur', 1),
    ('Turner Daniel', 'Daniel', 'Daniel.Turner@irap.omp.eu', 'Administration', 1),
    ('Sky Gin', 'Gin', 'Gin.Sky@irap.omp.eu', 'Responsable', 1),
    ('Robert Henri', 'Henri', 'Henri.Robert', 'Utilisateur', 1);
```

2) Sinon, le plus simple est d'utiliser un FAUX (fake) LDAP interne à l'application

Dans ce cas, il faut décommenter la variable \$fakeLDAP dans le fichier config.php et ajouter dans la BD les utilisateurs spécifiques suivants :

# Ajout d'utilisateurs de TEST (hors LDAP)

```
INSERT INTO utilisateurs (nom, login, email, role, groupes_metier_id) VALUES
    ('Hubert SuperAdmin', 'superadmin', 'hubert.superadmin@irap.omp.eu', 'Super Administrateur', 1),
    ('Pierre Responsable', 'responsable', 'pierre.resp@irap.omp.eu', 'Responsable', 2),
    ('Jean Administration', 'admin', 'Jean.Administration@irap.omp.eu', 'Administration', 5),
    ('Jacques Utilisateur', 'user', 'Jacques.Utilisateur@irap.omp.eu', 'Utilisateur', 7);
```

Informations sur le LDAP :

Classe LdapAuthComponent (extends AuthComponent), définie dans : app/Controller/Component/LdapAuthComponent.php

Définition des rôles (ACL) : app/Model/Utilisateur.php :

```
private $acceptedRoles = array ('Utilisateur', 'Responsable', 'Administration', 'Super Administrateur');
public function getAuthenticationLevelFromRole($role) {
    if ($role == 'Utilisateur')
        return 1;
    elseif ($role == 'Responsable')
        return 2;
    elseif ($role == 'Administration')
        return 3;
    elseif ($role == 'Super Administrateur')
        return 4;
    return 0;
}
```



```
}
```

Lecture du fichier de config : app/Model/LdapConnection.php :  
private function checkConfiguration()

Workflow du LDAP :

- 1) Page d'accueil : j'entre mon login et pwd
- 2) clic sur bouton "Se Connecter" --> execute l'action Utilisateurs/login (dans app/Controller/UtilisateursController.php)

# 9. Comment on a rendu le logiciel plus “RESPONSIVE” (avec bootstrap)

*\bootstrap \responsive  
(updated 12/2/20 - EP)*

**Réponse rapide : En utilisant bootstrap.**

Cette évolution vers une version mobile a été réalisée au 1er trimestre 2020.

Ce chapitre décrit la démarche qui a été adoptée, les difficultés rencontrées, et les solutions trouvées.

## 9.1. Analyse de l'existant et choix d'une solution technique

Il y a environ 4 “plugins” (parfois “helpers”) bootstrap disponibles pour cakephp, dont 2 qui semblent vraiment obsolètes.

Obsolètes :

- <https://github.com/WebAndCow/CakePHP-BsHelpers>
- ...

Actifs :

- <https://holt59.github.io/cakephp3-bootstrap-helpers/> (et <https://github.com/Holt59/cakephp3-bootstrap-helpers>)
- <https://github.com/FriendsOfCake/bootstrap-ui/tree/develop>
- <https://twitter-bootstrap-plugin-for-cakephp-3.readthedocs.io/en/latest/readme/> qui se base sur **bootstrap-ui** ci-dessus

On a choisi "**bootstrap-ui**" car géré par une communauté d'utilisateurs FOC (contrairement à holt59 qui est seul à maintenir son projet bien qu'actif...), dans sa version de développement qui utilise bootstrap 4 (ça serait un peu bête d'utiliser bootstrap 3 quand bootstrap4 est proposé par défaut sur le site officiel) :

<https://github.com/FriendsOfCake/bootstrap-ui/tree/develop> (dernier commit novembre 2019 donc c'est actif...)

Bootstrap est proposé par défaut en version 4 (et déjà en 4.4) : <https://getbootstrap.com>

Doc :

- <https://getbootstrap.com/docs/4.4/getting-started/introduction/>
- <https://getbootstrap.com/docs/4.4/examples/>

Plus d'infos sur les "helpers" cakephp : <https://book.cakephp.org/3/en/views/helpers.html>

## 9.2. Dépendances

Apparemment (à confirmer), bootstrap utiliserait :

- jQuery 3.2+ : <https://jquery.com>
- Popper.js 1.x : <https://github.com/popperjs/popper-core> (j'ai téléchargé directement : <https://unpkg.com/@popperjs/core@2.0.3/dist/umd/popper.min.js>)
- Fontawesome 5.x (<https://fontawesome.com>)

## 9.3. Installation du plugin bootstrap-ui

<https://github.com/FriendsOfCake/bootstrap-ui/tree/develop>

### 1 - installation du plugin

```
$ cd labinvent/  
$ php composer.phar require friendsofcake/bootstrap-ui
```

=> a ajouté la ligne suivante dans composer.json :

```
"friendsofcake/bootstrap-ui": "^1.4"
```

=> a modifié composer.lock en conséquence

=> **bootstrap-ui 1.4** installé

### Warnings :

*Package aferrandini/phpqrcode is abandoned, you should avoid using it. Use endroid/qr-code instead.*

*Package phpunit/phpunit-mock-objects is abandoned, you should avoid using it. No replacement was suggested.*

Package zendframework/zend-diactoros is abandoned, you should avoid using it. Use laminas/laminas-diactoros instead.

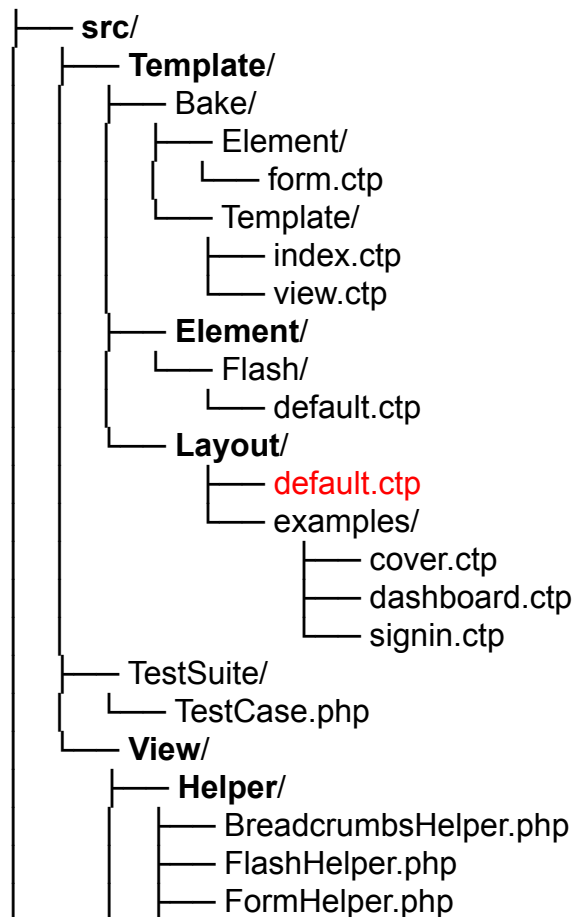
## 2 - Load

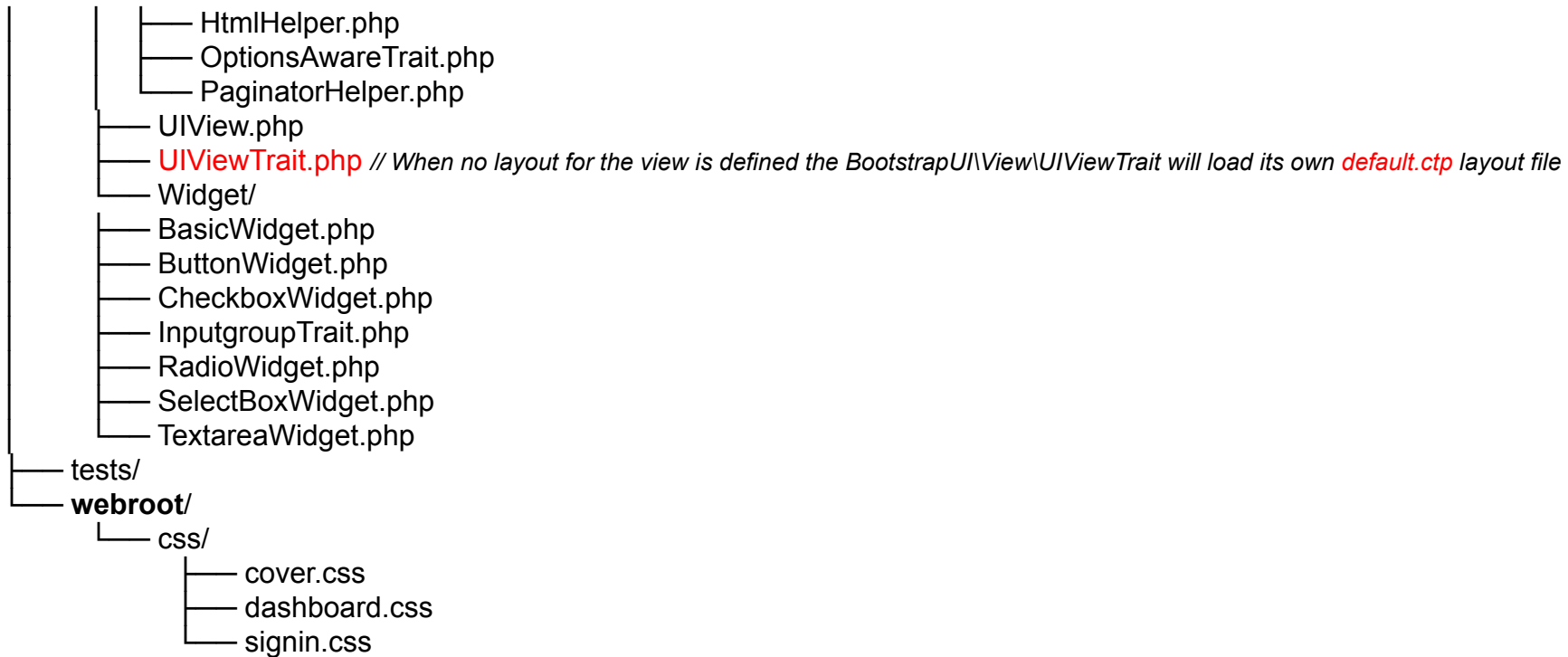
Add the following to your app's **config/bootstrap.php**:

```
Plugin::load('BootstrapUI');
```

## 9.4. Fichiers installés par bootstrap-ui

Contenu de vendor/friendsofcake/bootstrap-ui/





## 9.5. Informations générales

The `src\View\AppView.php` should look something like the following:

```

namespace App\View;
use BootstrapUI\View\UIView;
class AppView extends UIView
{

    /**
     * Initialization hook method.
     */
    public function initialize()
  
```

```
{
  //Don't forget to call the parent::initialize()
  parent::initialize();
}
}
```

or

```
namespace App\View;
use BootstrapUI\View\UIViewTrait;
use Cake\View\View;
class AppView extends View
{
  use UIViewTrait;

  /**
   * Initialization hook method.
   */
  public function initialize()
  {
    //render the initializeUI method from the UIViewTrait
    $this->initializeUI();
  }
}
```

## **Layout**

When no layout for the view is defined the **BootstrapUI\View\UIViewTrait** will load its own **default.ctp** layout file

## **Helper Usage**

(voir la section du même nom sur la page de bootstrap-ui pour avoir quelques exemples)

At the core of BootstrapUI is a collection of enhancements for CakePHP core helpers. These helpers replace the HTML templates used to render elements for the views. This allows you to create forms and components that use the Bootstrap styles.

The current list of enhanced helpers are:

- BootstrapUI\View\Helper\FlashHelper
- BootstrapUI\View\Helper\FormHelper
- BootstrapUI\View\Helper\HtmlHelper
- BootstrapUI\View\Helper\PaginatorHelper
- BootstrapUI\View\Helper\BreadcrumbsHelper

When the BootstrapUI\View\UIViewTrait is initialized it loads the above helpers with the same aliases as the CakePHP core helpers. That means that when you use `$this->Form->create()` in your views, the helper being used is from the BootstrapUI plugin.

⇒ voir dans src/Template/ (par exemple src/Template/Materiels/add.ctp)

## 9.6. Principe d'utilisation

On garde la page de layout (mise en page) par défaut de LabInvent, c'est à dire : **src/Template/Layout/default.ctp**

On l'a juste un peu adaptée pour qu'elle ajoute bootstrap.css, bootstrap.js, et jquery.js, et on en a profité aussi pour la nettoyer car c'était plutôt crade... Pour cela, on s'est inspiré de la page de layout par défaut de bootstrap-ui :

**vendor/friendsofcake/bootstrap-ui/src/Template/Layout/default.ctp**

Rappel : toutes les pages web du site utilisent le fichier **src/Template/Layout/default.ctp** comme template par défaut (car il fait la mise en page générale avec le menu à gauche, le titre en haut, le footer en bas...).

Ce template contient une section (<div>) nommée "**content**" (vers la ligne 120) :

```
<!-- Contenu principal de chaque page web -->
```

```
<div id="content">
```

```
<?=$this->Flash->render()?>
```

```
<?=$this->Flash->render('auth')?>
```

```
<?=$this->fetch('content')?>
```

```
</div> <!-- content -->
```

Cette section contient à son tour une ligne qui va charger le “bloc” nommé “**content**” :

```
<?=$this->fetch('content')?>
```

C'est à cet endroit qu'est copié tout le contenu de chaque page web, donc par exemple la page `src/Template/Materiels/index.ctp` pour la liste des matériels (voir dans cette doc sur le même sujet le chapitre [13.5. Structure générale d'une page web \(TEMPLATE\) = default.ctp](#))

Maintenant, concernant bootstrap. Attention, on ne touche plus au fichier `src/View/AppView.php`.

Si on veut voir la version du site AVANT bootstrap (c'est à dire la version d'origine), il suffit de faire ceci :

```
$ cd src/Template/Layout/
```

```
$ cp default_sans_bootstrap.ctp default.ctp
```

Pour revenir à la version AVEC bootstrap :

```
$ cp default_avec_bootstrap.ctp default.ctp
```

Comme ça on peut comparer et avancer tranquillement dans les modifs.

Maintenant, si on veut adapter une page web précise, par exemple la liste des matériels, il faut modifier le fichier suivant :

```
src/Template/Materiels/index.ctp
```

Pour adapter la page web de la vue d'un seul matériel :

```
src/Template/Materiels/view.ctp
```



Pour adapter la page web de l'ajout d'un matériel :

```
src/Template/Materiels/add.ctp
```

Pour adapter la page web de la modification d'un matériel :

```
src/Template/Materiels/edit.ctp
```

etc...

## 9.7. BUGFIXES

Ces bugs sont-ils des bugs de bootstrap ou de bootstrap-ui, ou bien sont-ils liés à une mauvaise installation de bootstrap-ui ? En effet, bootstrap-ui nécessitait normalement le gestionnaire de package npm (qui vient avec l'installation de node.js), or celui-ci n'a pas été installé car ne semblait pas indispensable (et ça ajouterait une dépendance contraignante et peut-être inutile au projet).

### **Bugfix 1 - Les messages Flash (erreur ou succès) sont devenus invisibles**

*(EP 3/3/20)*

Le fichier CSS `_transitions.scss` (qui est liée à bootstrap mais qui vient de je ne sais où car ce fichier est absent du projet...) ajoute cette règle qui empêche de voir les messages d'alerte Flash (`$this->Flash->error` ou `$this->Flash->success` par exemple) :

```
.fade:not(.show) {  
  opacity: 0;  
}
```

La feuille de style `webroot/css/inventirap.css` étant chargée APRÈS `bootstrap.css`, je peux ajouter une règle dans `inventirap.css` qui surcharge (écrase) la règle du même nom dans `bootstrap.css`

J'ai donc ajouté le code suivant à la fin du fichier inventirap.css :

```
.fade:not(.show) {  
  opacity: 1;  
}
```

## 9.8. Revenir à la version NOT RESPONSIVE (pour debug)

*(EP 4/3/20)*

Dans un but de déboguage, il est très simple de revenir à la version NOT RESPONSIVE, bien que ce retour ne soit quand même que partiel.

Il suffit de changer 2 fichiers :

### **Changer le fichier AppView :**

```
$ cd src/View/
```

```
$ cp AppView.sans_bootstrap.php AppView.php
```

### **Changer le layout par défaut :**

```
$ cd src/Template/Layout/
```

```
$ cp default_sans_bootstrap.ctp default.ctp
```

# 10. Roadmap (feuille de route) (TODO LIST)

\todo

**La Roadmap est désormais gérée sur une feuille de calcul partagée.  
Vous la trouverez [ICI](#).**

*ANCIENNE ROADMAP officielle gérée sur le REDMINE du laboratoire IRAP : [Suivre ce lien](#) (version en cours v2.14)*

## 10.1. TODO LIST temporaire à migrer sur le Redmine (notes urgentes)

### URGENT :

- ~~— UPDATE : “Version actuelle” puis “Nouvelle version :”~~
- ~~— LOGOs personnalisés~~
- ~~— DOC :~~
  - ~~— logos personnalisés~~
  - ~~— set authorizations~~
- ~~— Ajouter filtre par domaine~~
- ~~— script update BD : un seul script exécuté avec un argument : “exec\_sql \$nom\_du\_script.sql”~~
- ? Ajouter filtre par année, par personne...
  
- **Convertir les scripts shell en PHP :**
  - install/installation.sh => installation.php
  - database/update/db-update.sh script.sql => db-update.php script.sql
  - install/update.sh => update.php
  - ...
  
- **(TODO) :**

### (Lot 1)

- ~~— retour du champ recherche générique sur page materiels/index/ (comme avant), désactiver le champ de recherche par défaut dans le menu quand on est sur la page index~~
- ~~— Nouvelle page web "Statistiques mensuelles" (/pages/stats)
  - ~~— ajouter suivis et emprunts~~~~
- **bugfix test dates : MaterielsControllerTest::testMaterielDatesValidation**
- ~~— files\_to\_process.txt~~
- ~~— bugfix champ recherche général~~
- ~~— bugfix fournisseur~~
- ~~— bugfix tests~~
- stats plus jolies
- **Filtres directement sur colonnes du tableau materiels/index/ :**
  - statut
  - pagination
  - age
  - domaine
  - ...

### (Lot 2)

- ~~— Pas de mail envoyé quand VALIDATED !!~~
- **Durée garantie = from date achat ou livraison ?**
- ~~— DYMO étiquettes 19-24mm~~
- **Ajouter un lien URL cliquable dans le champ description**
- ~~— Gestionnaire de ref : ajouté automatiquement si user = admin et gestionnaire\_ref vide~~
- ~~— label des filtres~~
- **Saisir fiches pweb et pweb2 et hyp ?**
- 
- ~~— Lien page view vers projet + index~~
- 
- 
- **Fiche hyperion2**
- **install LabInvent sur VM et/ou pweb2**
- **Update VM Ubuntu**
- 
- 
-

### (Lot3)

- Suivis relance (CRAL)
- version SANS LDAP ok
- install from scratch ok

### (Lot4)

- cakephp 3.8 avec Application.php
- Update VM Ubuntu => mettre en ligne téléchargeable
- Update pweb2 ou/et VM chemcam : php7 + mysql récent

### (Lot5)

- Php7
- CakePhp4

### **AUTRES :**

- BUGIFX : En mode ldap\_used, un user est dans la table users mais pas dans le ldap, il peut donc se connecter, mais \$this->ldapAuth->user() est à null après connexion...
- Ajouter colonne from\_ldap dans table users pour distinguer les users qui viennent du ldap de ceux qui sont créés localement... (???)
- Connexion via LDAP marche pas si option "Optimisation accès LDAP" pas activée... car table users pas mise à jour !!!  
=> l'activer obligatoirement
- La TOUTE première mise à jour de la table users from le LDAP semble ne pas fonctionner : à l'IAS on est passé de 15 utilisateurs dans la table à plus de 300 après update, mais ça s'est fait longtemps après... et il a fallu insérer l'utilisateur "caminade" manuellement dans la table users pour qu'elle se mette à jour ("caminade" n'était pas dedans avant) => En fait, utiliser le mode ldap\_used SANS activer le cache ldap (option "Optimisation accès LDAP") semble poser pb !!

~~— bugfix emprunt en cours : <https://inventirap-test.irap.omp.eu/materiels/view/13166>~~

~~— pb update table users from ldap : pas à jour pour collègue Elodie~~

~~— Superadmin a tous les droits (depuis menu Tools)~~

~~— page acls : explication nom des actions~~

~~— Suivi (simplification) : saisie des groupes thématique et métier INUTILE => à virer car redondant~~

~~— Matériel : technique ou/et inventorable => trop compliqué et inutile => en fait c'est un OUX (exclusif)~~

~~— => simplifier en ne gardant qu'UNE SEULE case à cocher qui s'appelle "matériel technique".~~

—=> Si cette case n'est pas cochée, le prix doit alors être supérieur à 1000E

—=> par contre, on garde quand même le champ ds la BD au cas où...

#### — Colorisation selon statut du matériel :

— CREATED => bleu

— VALIDATED => vert

— TBA => orange

— ARCHIVED => rouge

#### - Tests manquants :

— Tester qu'un user ne peut pas valider un matos !!!

— Superadmin (ou autre profil) ne doit pas pouvoir changer le statut d'un matériel qui est dans N'IMPORTE QUEL statut :

— if (\$this->action=='statusValidated') return \$this->isCreated(\$id);

— if (\$this->action=='statusToBeArchived') return \$this->isValidated(\$id);

— if (\$this->action=='statusArchived') return \$this->isToBeArchived(\$id);

— if (\$this->action=='statusCreated') return !\$this->isCreated(\$id);

— un "Utilisateur" ne doit pas pouvoir changer les statuts

#### - Fournisseur :

- Ajouter fournisseur si pas dans liste

- Ajouter => doit amener à view (vue détaillée) et non pas index

- problème de tri par nom bizarre pas logique... (caractères spéciaux cachés ?)

- Traitement liste Fournisseurs : vraiment fini ??? (je devais faire une boucle sur les fournisseurs existants...) => fait ?

- champ "matériels associés" ne se masque plus... (voir autres listes)

- find "anterr" => ne trouve pas la fiche 11123 carte accélétratrice de Anterrieu (car il est ancien utilisateur seulement)

- module métrologie => check AVEC ou SANS

<http://labinvent.devv/materiels/view/11123>

—superadmin doit dévalider avant de revalider (now il peut modifier un matos validé => NON, il vaut mieux que son comportement ressemble le plus possible à celui de "Administration", avec qq droits en plus seulement)

**Quand je vois ce matériel, je ne peux pas l'éditer, alors que dans la vue FIND "anterr", j'ai l'icone edit !!! alors que je devrais pas pouvoir car ce matos est validé !!! et en plus j'ai réussi à le modifier, je dois dé-valider d'abord !!! (pire que tout)**

**=> ajouter un test !!**

- **?Bugfix** nom\_createur et nom\_modificateur => sont tous à NULL !!! (alors que nom\_responsable est bien renseigné)

-

- Installer la nouvelle étiqueteuse
- 
- Faire les modifs demandées par l'équipe de direction (C. Feugeade)
- 
- Migration des changements faits en BD sur VM linux
- 
- **Suivis : améliorer + traitement cron**
- 
- MyButtonHelper => migrer dans HtmlHelper
- 
- Update script création BD général
- 
- \$RGs : règles de gestion doivent toutes être dans les modèles et non pas dans les vues en javascript, c'est insuffisant pour valider les données envoyées
- 
- Check que toutes les infos lues dans doc cakephp sont mises en pratique
- 
- 
- Emprunts : boutons x 2
- 
- Tester date et nombres (Number) localisés avec \_\_() pour voir si affichés différemment...
- 
- 
- Mettre à jour TODO list sur gdoc
- 
- fakeLDAP bugfix

#### Bugfix DocumentsTable.php :

- **buildRules() :**  

```

$checkPhotoFormat = function ($entity) {
if (! empty($entity->get('chemin_file')['tmp_name'])) {
    if ($entity->get('photo') || $entity->get('type_document_id')==4) {
        $extension = strtolower(pathinfo($entity->get('chemin_file')['name'], PATHINFO_EXTENSION));
        return in_array($extension, $this->photo_formats);
    }
}
return true;

```

```

}
if ( $entity->get('edit') && $entity->get('type_document_id')==4 && !in_array($entity->get('type_doc'), $this->photo_formats))
    return false;
return true;
};

```

- **beforeSave():**

```

if ( in_array($extension, $this->photo_formats) ) {
if ( $entity->get('type_document_id')==1 )
    $entity->set('type_document_id', 4);
}

```

**Bugfix Materiels/view.ctp :**

```

$sis_doc_photo = $document->type_document_id == 4;

```

**MOINS URGENT :**

- **CRUD :**

- <https://crud.readthedocs.io/en/latest/quick-start.html>
- <https://github.com/friendsofcake/crud>
- <https://github.com/lorenzo/cakephp3-bookmark>

- **Ajouter src/Application.php :**

- voir <https://book.cakephp.org/3/fr/development/application.html>
- aussi voir section \application

- update to cakephp last v3 (v3.8) => suivre pour cela le [guide de migrations](#)
- update to cakephp v4 et php7

— LOG pertinent : aucun pour l'instant ???

- **Séparer Authentification et Autorisation** en utilisant le nouveau plugin d'authentification **AuthenticationComponent** au lieu de l'actuel **AuthComponent** (en fait **LdapAuthComponent** qui hérite de **AuthComponent**, dans src/Controller/Component/), et ensuite en ajoutant le **plugin Authorization** pour gérer les autorisations.

Pour ça, voir <https://book.cakephp.org/authentication/2/en/migration-from-the-authcomponent.html>

- **Parallelisation des traitements longs, tels que les envois d'emails !!!** => <https://github.com/dereuromark/cakephp-queue>



~~— Utilisation du helper (raccourci) **Html** le plus souvent possible~~

- **vue "view"** : ajouter un bouton "TOUS" pour aller à la vue "index"

- **SUIVIS** :

- table "type\_suivi" => ajouter "Vérification métrologique" par défaut dans script installation (et faire une requete de mise à jour)
- edit :
  - onchange(type) => javascript pour changer les champs date...
  - ~~— champs trop larges (prennent toute la longueur écran)~~

~~- invalider => dé-valider~~

- **boutons Enregistrer & CANCEL sur toutes les vues Edit**

~~- recherche : doit s'adapter au controleur où on est => si je suis sur liste des fournisseurs, je dois pouvoir chercher fournisseur !!!~~

- **ajouter photos sur tous mes matos**

- **VM linux => snapshot**

- **Formulaire erreur** : bulle "Veuillez compléter ce champ" ne reste pas sur le champ concerné quand on scrolle..

-

~~- **REFACTORISER toutes les vues add et edit (en 1 seule add\_edit) des différents objets (materiel, suivi, emprunt...)**~~

- **doc attaché** :

- **tester avec un "suivi" (au lieu d'un matériel)**
- **factoriser edit.ctp et add.ctp => héritage de template**
- helpers pour faire echoButtons() ou autres...: <https://book.cakephp.org/3/fr/views/helpers.html>
- tester envoi par mail (from irap only)
- **ajouter action mv photos from files to img/photos/ dans UPDATE ???**

- Autorisations : tout le monde ?
  - type\_document\_id == 4 => remplacer par "photo"
  - type\_document\_id == 1 => remplacer par "N/A"
  - Ajouter TEST
  - **materiels/view/ :**
    - (rendre images cliquables pour aller sur leur view)
    - voir TOUTES les photos
    - Afficher QrCode tout en haut
    - ~~lier doc et photo : doublon~~
  - **documents/mail-devis/ :**
    - bugfix chemin quand c'est un doc photo !!!
    - **bugfix quand c'est un suivi**
  - **documents/view/ :**
    - afficher si photo (ou type photo)
    - **ajouter boutons "Edit" et "Delete"**
    - télécharger photo => sert à rien (en fait, c'est voir)
  - **documents/edit :**
    - ajouter bouton ~~CANCEL~~
    - ajouter simple lien vers conteneur (matos/suivi) et supprimer autres champs inutiles
    - (on doit pouvoir changer le nom)
    - (on doit pouvoir changer le doc !! (bouton upload absent)
      - => (check à nouveau la taille... comme pour add)
  - **documents/add :**
    - si doc type photo (png, jpeg...) => mettre dans img/photos (et non pas dans files/)
    - ajouter bouton ~~CANCEL~~
    - si add photo => positionner automatiquement type à photo (même si pas précisé)
    - si max upload limit (de php.ini ou config) => message erreur clair
  - **documents/delete :**
    - **doit se faire proprement !!**
- 
- boutons décalés : menu (pc) et boutons submit Cancel (telephone)
  - fournisseur => ajouter/modifier
  - gérer les listes, gérer les catégories... => on ne peut plus ajouter !!!

Cette section contient des actions notées en urgence en attendant d'être ajoutées à la roadmap officielle (lien redmine ci-dessus).

**(27/1/20) TODO au moins 1 fois sur chaque client git (CRAL, IRAP, ...):**

**Mise à jour des plugins php (seulement si nécessaire) :**

**\$ git pull**

**\$ mv vendor vendor.old**

**\$ mv composer.lock composer.lock.old**

**\$ cd install/**

**\$ ./plugins\_install.sh**

**\$ cd ../**

**\$ git status**

(devrait être clean)

**TODO install:**

**dompok/dompok suggests installing ext-imagick (Improves image processing performance)**

**dompok/dompok suggests installing ext-gmagick (Improves image processing performance)**

**psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)**

**=> dompok/dompok suggests installing ext-gd (Needed to process images)**

**Autres à installer :**

cakephp/cakephp suggests installing lib-ICU (The intl PHP library, to use Text::transliterate() or Text::slug())

symfony/console suggests installing symfony/event-dispatcher

symfony/console suggests installing symfony/lock

dompok/dompok suggests installing ext-imagick (Improves image processing performance)

dompok/dompok suggests installing ext-gmagick (Improves image processing performance)

psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)

psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)

asm89/twig-cache-extension suggests installing psr/cache-implementation (To make use of PSR-6 cache implementation via PsrCacheAdapter.)

aptoma/twig-markdown suggests installing michelf/php-markdown (Original Markdown engine with MarkdownExtra.)

aptoma/twig-markdown suggests installing knplabs/github-api (Needed for using GitHub's Markdown engine provided through their API.)

ajgl/breakpoint-twig-extension suggests installing ext-xdebug (The Xdebug extension is required for the breakpoint to work)

ajgl/breakpoint-twig-extension suggests installing symfony/framework-bundle (The framework bundle to integrate the extension into Symfony)  
ajgl/breakpoint-twig-extension suggests installing symfony/twig-bundle (The twig bundle to integrate the extension into Symfony)  
sebastian/global-state suggests installing ext-uopz (\*)  
phpunit/php-code-coverage suggests installing ext-xdebug (^2.5.1)  
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)  
phpunit/phpunit suggests installing ext-xdebug (\*)

Package aferrandini/phpqrcode is abandoned, you should avoid using it. Use endroid/qr-code instead.

Package zendframework/zend-diactoros is abandoned, you should avoid using it. Use laminas/laminas-diactoros instead.

Package phpunit/phpunit-mock-objects is abandoned, you should avoid using it. No replacement was suggested.

## **RUN tests**

**\$ ./UPDATE (=> va voir que composer.json a changé et donc provoquer une install des plugins)**

\* 1) Mise à jour du code source (git pull) (o/n) ? [o]

...

\* 2) Mise à jour des plugins php (seulement si nécessaire) :

**./plugins\_install.sh**

Execution de composer...

Loading composer repositories with package information

Installing dependencies (including require-dev) from lock file

Warning: The lock file is not up to date with the latest changes in composer.json. You may be getting outdated dependencies. Run update to update them.

Package operations: 78 installs, 0 updates, 0 removals

- Installing cakephp/plugin-installer (1.1.0): Loading from cache
- Installing bamarni/composer-bin-plugin (v1.3.0): Loading from cache
- Installing aferrandini/phpqrcode (1.0.1): Loading from cache
- Installing aura/intl (3.0.0): Loading from cache
- Installing jdorn/sql-formatter (v1.2.17): Loading from cache
- Installing symfony/process (v4.2.8): Loading from cache
- Installing symfony/finder (v4.2.8): Loading from cache
- Installing symfony/polyfill-ctype (v1.11.0): Loading from cache
- Installing symfony/filesystem (v4.2.8): Loading from cache
- Installing symfony/polyfill-mbstring (v1.11.0): Loading from cache
- Installing psr/log (1.1.0): Loading from cache
- Installing symfony/debug (v4.2.8): Loading from cache
- Installing symfony/console (v3.4.27): Loading from cache
- Installing seld/phar-utils (1.0.1): Loading from cache
- Installing seld/jsonlint (1.7.1): Loading from cache
- Installing justinrainbow/json-schema (5.2.8): Loading from cache
- Installing composer/xdebug-handler (1.3.2): Loading from cache
- Installing composer/spdx-licenses (1.5.1): Loading from cache
- Installing composer/semver (1.5.0): Loading from cache
- Installing composer/ca-bundle (1.1.4): Loading from cache
- Installing composer/composer (1.8.5): Loading from cache
- Installing cakephp/chronos (1.2.5): Loading from cache
- Installing psr/http-message (1.0.1): Loading from cache
- Installing zendframework/zend-diactoros (1.8.6): Loading from cache
- Installing psr/simple-cache (1.0.1): Loading from cache
- Installing cakephp/cakephp (3.7.7): Loading from cache
- Installing cakephp/debug\_kit (3.18.0): Loading from cache
- Installing symfony/yaml (v3.4.27): Loading from cache
- Installing symfony/config (v3.4.27): Loading from cache
- Installing robmorgan/phinx (v0.8.1): Loading from cache
- Installing cakephp/migrations (1.8.1): Loading from cache
- Installing sabberworm/php-css-parser (8.1.0): Loading from cache
- Installing phenx/php-svg-lib (v0.3.2): Loading from cache
- Installing phenx/php-font-lib (0.5.1): Loading from cache

- Installing dompdf/dompdf (v0.8.3): Loading from cache
- Installing daoandco/cakephp-dompdf (v1.2.1): Loading from cache
- Installing friendsofcake/bootstrap-ui (1.4.1): Loading from cache
- Installing mobiledetect/mobiledetectlib (2.8.33): Loading from cache
- Installing setasign/fpdf (1.8.1): Loading from cache
- Installing twig/twig (v1.40.1): Loading from cache
- Installing umpirsky/twig-php-function (v0.1): Loading from cache
- Installing jasny/twig-extensions (v1.2.0): Loading from cache
- Installing asm89/twig-cache-extension (1.3.2): Loading from cache
- Installing aptoma/twig-markdown (2.0.0): Loading from cache
- Installing ajgl/breakpoint-twig-extension (0.3.4): Loading from cache
- Installing wyrihaximus/twig-view (4.3.8): Loading from cache
- Installing cakephp/bake (1.9.6): Loading from cache
- Installing jakub-onderka/php-console-color (v0.2): Loading from cache
- Installing webmozart/assert (1.4.0): Loading from cache
- Installing phpdocumentor/reflection-common (1.0.1): Loading from cache
- Installing phpdocumentor/type-resolver (0.4.0): Loading from cache
- Installing phpdocumentor/reflection-docblock (4.3.1): Loading from cache
- Installing phpunit/php-token-stream (2.0.2): Loading from cache
- Installing sebastian/version (2.0.1): Loading from cache
- Installing sebastian/resource-operations (1.0.0): Loading from cache
- Installing sebastian/recursion-context (2.0.0): Loading from cache
- Installing sebastian/object-enumerator (2.0.1): Loading from cache
- Installing sebastian/global-state (1.1.1): Loading from cache
- Installing sebastian/exporter (2.0.0): Loading from cache
- Installing sebastian/environment (2.0.0): Loading from cache
- Installing sebastian/diff (1.4.3): Loading from cache
- Installing sebastian/comparator (1.2.4): Loading from cache
- Installing phpunit/php-text-template (1.2.1): Loading from cache
- Installing doctrine/instantiator (1.2.0): Loading from cache
- Installing phpunit/phpunit-mock-objects (3.4.4): Loading from cache
- Installing phpunit/php-timer (1.0.9): Loading from cache
- Installing phpunit/php-file-iterator (1.4.5): Loading from cache
- Installing sebastian/code-unit-reverse-lookup (1.0.1): Loading from cache
- Installing phpunit/php-code-coverage (4.0.8): Loading from cache
- Installing phpspec/prophecy (1.8.0): Loading from cache
- Installing myclabs/deep-copy (1.9.1): Loading from cache

- Installing phpunit/phpunit (5.7.27): Loading from cache
- Installing symfony/polyfill-php72 (v1.11.0): Loading from cache
- Installing symfony/var-dumper (v4.2.8): Loading from cache
- Installing nikic/php-parser (v4.2.1): Loading from cache
- Installing jakub-ondrejka/php-console-highlighter (v0.4): Loading from cache
- Installing dnoegel/php-xdg-base-dir (0.1): Loading from cache
- Installing psy/psysh (v0.9.9): Loading from cache

symfony/console suggests installing symfony/event-dispatcher

symfony/console suggests installing symfony/lock

cakephp/cakephp suggests installing lib-ICU (The intl PHP library, to use Text::transliterate() or Text::slug())

dompdf/dompdf suggests installing ext-gmagick (Improves image processing performance)

dompdf/dompdf suggests installing ext-imagick (Improves image processing performance)

asm89/twig-cache-extension suggests installing psr/cache-implementation (To make use of PSR-6 cache implementation via PsrCacheAdapter.)

aptoma/twig-markdown suggests installing knplabs/github-api (Needed for using GitHub's Markdown engine provided through their API.)

aptoma/twig-markdown suggests installing michelf/php-markdown (Original Markdown engine with MarkdownExtra.)

ajgl/breakpoint-twig-extension suggests installing ext-xdebug (The Xdebug extension is required for the breakpoint to work)

ajgl/breakpoint-twig-extension suggests installing symfony/framework-bundle (The framework bundle to integrate the extension into Symfony)

ajgl/breakpoint-twig-extension suggests installing symfony/twig-bundle (The twig bundle to integrate the extension into Symfony)

sebastian/global-state suggests installing ext-uopz (\*)

phpunit/php-code-coverage suggests installing ext-xdebug (^2.5.1)

phpunit/phpunit suggests installing ext-xdebug (\*)

phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)

psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)

psy/psysh suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)

**Package aferrandini/phpqrcode is abandoned, you should avoid using it. Use endroid/qr-code instead.**

**Package zendframework/zend-diactoros is abandoned, you should avoid using it. Use laminas/laminas-diactoros instead.**

**Package phpunit/phpunit-mock-objects is abandoned, you should avoid using it. No replacement was suggested.**

Generating autoload files

```
> Cake\Composer\Installer\PluginInstaller::postAutoloadDump
> App\Console\Installer::postInstall
```

**Script App\Console\Installer::postInstall handling the post-install-cmd event terminated with an exception**

[ErrorException]

**mkdir(): Permission denied**

```
install [--prefer-source] [--prefer-dist] [--dry-run] [--dev] [--no-dev] [--no-custom-installers] [--no-autoloader] [--no-scripts] [--no-progress]
[--no-suggest] [-v|vv|vvv|--verbose] [-o|--optimize-autoloader] [-a|--classmap-authoritative] [--apcu-autoloader] [--ignore-platform-reqs] [--]
[<packages>]...
```

Execution de cake plugin assets...

For plugin: BootstrapUI

-----  
Created symlink /Users/epallier/\_PROJ/\_W/LABINVENT/SOURCE/labinvent2\_MASTER/webroot/bootstrap\_u\_i

For plugin: DebugKit

-----  
Created symlink /Users/epallier/\_PROJ/\_W/LABINVENT/SOURCE/labinvent2\_MASTER/webroot/debug\_kit

For plugin: Dompok

-----  
Created symlink /Users/epallier/\_PROJ/\_W/LABINVENT/SOURCE/labinvent2\_MASTER/webroot/dompok

Done

**\$. /plugins\_update.sh**

Password:

Execution de composer...

Loading composer repositories with package information

Updating dependencies (including require-dev)

Package operations: 0 installs, 42 updates, 0 removals

- Updating cakephp/plugin-installer (1.1.0 => 1.2.0): Downloading (100%)
- Updating mobiledetect/mobiledetectlib (2.8.33 => 2.8.34): Downloading (100%)
- Updating symfony/polyfill-ctype (v1.11.0 => v1.14.0): Downloading (100%)
- Updating symfony/filesystem (v4.2.8 => v4.4.5): Downloading (100%)
- Updating symfony/polyfill-mbstring (v1.11.0 => v1.14.0): Downloading (100%)
- Updating psr/log (1.1.0 => 1.1.2): Downloading (100%)



- Updating symfony/debug (v4.2.8 => v4.4.5): Downloading (100%)
- Updating symfony/console (v3.4.27 => v3.4.38): Downloading (100%)
- Updating seld/phar-utils (1.0.1 => 1.1.0): Downloading (100%)
- Updating seld/jsonlint (1.7.1 => 1.7.2): Downloading (100%)
- Updating justinrainbow/json-schema (5.2.8 => 5.2.9): Downloading (100%)
- Updating composer/xdebug-handler (1.3.2 => 1.4.1): Downloading (100%)
- Updating composer/spdx-licenses (1.5.1 => 1.5.3): Downloading (100%)
- Updating composer/semver (1.5.0 => 1.5.1): Downloading (100%)
- Updating composer/ca-bundle (1.1.4 => 1.2.6): Downloading (100%)
- Updating symfony/process (v4.2.8 => v5.0.5): Downloading (100%)
- Updating symfony/finder (v4.2.8 => v5.0.5): Downloading (100%)
- **Updating composer/composer (1.8.5 => 1.10.0): Downloading (100%)**
- Updating cakephp/chronos (1.2.5 => 1.3.0): Downloading (100%)
- Updating zendframework/zend-diactoros (1.8.6 => 1.8.7): Downloading (100%)
- **Updating cakephp/cakephp (3.7.7 => 3.7.9): Downloading (100%)**
- Updating cakephp/debug\_kit (3.18.0 => 3.22.2): Downloading (100%)
- Removing dnoegel/php-xdg-base-dir (0.1)
- Installing dnoegel/php-xdg-base-dir (v0.1.1): Downloading (100%)
- Updating nikic/php-parser (v4.2.1 => v4.3.0): Downloading (100%)
- Updating symfony/polyfill-php72 (v1.11.0 => v1.14.0): Downloading (100%)
- Updating symfony/var-dumper (v4.2.8 => v4.4.5): Downloading (100%)
- Updating psy/psysh (v0.9.9 => v0.9.12): Downloading (100%)
- Updating twig/twig (v1.40.1 => v1.42.5): Downloading (100%)
- Updating asm89/twig-cache-extension (1.3.2 => 1.4.0): Downloading (100%)
- Updating cakephp/bake (1.9.6 => 1.10.1): Downloading (100%)
- Updating symfony/config (v3.4.27 => v3.4.38): Downloading (100%)
- Updating symfony/yaml (v3.4.27 => v3.4.38): Downloading (100%)
- Updating sabberworm/php-css-parser (8.1.0 => 8.3.0): Downloading (100%)
- Updating phenx/php-svg-lib (v0.3.2 => v0.3.3): Downloading (100%)
- **Updating dompdf/dompdf (v0.8.3 => v0.8.5): Downloading (100%)**
- Updating myclabs/deep-copy (1.9.1 => 1.9.5): Downloading (100%)
- Updating doctrine/instantiator (1.2.0 => 1.3.0): Downloading (100%)
- Updating webmozart/assert (1.4.0 => 1.7.0): Downloading (100%)
- Updating phpdocumentor/reflection-common (1.0.1 => 2.0.0): Downloading (100%)
- Updating phpdocumentor/type-resolver (0.4.0 => 1.1.0): Downloading (100%)
- Updating phpdocumentor/reflection-docblock (4.3.1 => 5.1.0): Downloading (100%)
- Updating phpspec/prophecy (1.8.0 => v1.10.3): Downloading (100%)

Package aferrandini/phpqrcode is abandoned, you should avoid using it. Use endroid/qr-code instead.

Package phpunit/phpunit-mock-objects is abandoned, you should avoid using it. No replacement was suggested.

Package zendframework/zend-diactoros is abandoned, you should avoid using it. Use laminas/laminas-diactoros instead.

Writing lock file

Generating autoload files

> Cake\Composer\Installer\PluginInstaller::postAutoloadDump

5 packages you are using are looking for funding.

Use the composer fund command to find out more!

Execution de cake plugin assets...

For plugin: BootstrapUI

---

For plugin: DebugKit

---

For plugin: Dompdf

---

Done

**Autres solutions existantes :**

- <https://github.com/abedputra/Inventory-Stock-With-Scanner>

**bootstrap:**

- 12 colonnes max
- offset-1 pour ajouter 1 colonne vide de séparation par exemple

## 10.2. ANCIENNE ROADMAP

Cette roadmap a déjà été transférée sur le Redmine officiel le 7/6/19.  
Elle est gardée ici juste pour info.

⇒ Accès direct aux thèmes:

- [GENERAL](#)
- [USERS](#)
- [ETIQUETTES](#)
- [MATERIELS](#)
- [SUIVIS](#)
- [EMPRUNTS](#)

migré dans redmine	THEME	ITEM	FAIT
<p><b>LÉGENDE:</b></p> <ul style="list-style-type: none"> <li>- <b>Urgence:</b> <i>Très Urgent</i>, <i>Assez Urgent</i>, par défaut = peut attendre</li> <li>- <b>Type:</b> <b>(B)</b> = Bug ou anomalie ; <b>(I)</b> = Improvement (amélioration) ; <b>(N)</b> = Nouvelle fonctionnalité ; <b>(M)</b> = Modification (changement)</li> </ul>			
<b>GENERAL</b>			
X	Version mobile	??? (oui, mais par un stagiaire)	
X	Database	<p><b>(I) Changer toutes les contraintes</b> pour remplacer “on delete no action” par “on delete SET NULL”. (ou CASCADE selon les cas)</p> <p>Déjà fait pour materiels.fk_materiels_gestionnaire_id, en 2 temps :</p> <ol style="list-style-type: none"> <li>1) supprimer contrainte actuelle</li> </ol> <pre>ALTER TABLE materiels DROP FOREIGN KEY fk_materiels_gestionnaire_id;</pre>	

		<p>2) Recréer contrainte  <u>ALTER TABLE</u> materiels  ADD CONSTRAINT fk_materiels_gestionnaire_id  FOREIGN KEY (gestionnaire_id)  REFERENCES users (id)  ON <u>DELETE SET</u> NULL ON <u>UPDATE</u> NO ACTION;</p>	
	<b>CONFIG</b>	<b>(M) Passer seuil de 800€ à 1000€</b>	26/03/19 (EP)
<b>X</b>	<b>Website</b>	<p><b>(I) (13/3/19) Fichiers logo ne doivent pas être versionnés (git)</b>  Il s'agit des 2 fichiers logo_entity.jpg et logo_software.jpg  Ils ne doivent pas être versionnés directement depuis le dossier "destination" webroot/img/  Ils doivent plutôt être copiés lors de la phase d'installation, depuis un dossier "source"  (versionné) vers le dossier "destination" (non versionné)  Cela afin d'éviter que le labo qui change ces logos les change pour tout le monde !!!</p>	
<b>X</b>	<b>Plugins update</b>	<p><b>(I) (31/1/19) Changer le plugin de création des QrCodes :</b>  <i>"Package aferrandini/phpqrcode is abandoned, you should avoid using it.  Use android/qr-code instead."</i></p> <p><b>(I) (15/4/19) Installer mpdf à la place de fpdf</b> (qui n'a pas de mise à jour depuis 2015 !!!)  <a href="https://mpdf.github.io">https://mpdf.github.io</a>  <a href="https://mpdf.github.io/installation-setup/installation-v7-x.html">https://mpdf.github.io/installation-setup/installation-v7-x.html</a></p>	
<h2>USERS</h2> <p>(Gestion des utilisateurs)</p>			
<b>inutile ?</b>	<b>DIVERS</b>	<p><b>(I) (12/2/19) Plugin Users</b> pour mieux gérer les users</p> <ul style="list-style-type: none"> <li>○ <a href="https://github.com/CakeDC/users">https://github.com/CakeDC/users</a></li> <li>○ Exemple app: <a href="https://github.com/CakeDC/users-example">https://github.com/CakeDC/users-example</a></li> </ul>	

		<p><del>(B) (25/4/19) ajouter un utilisateur avec privilège</del></p> <p>Dans le menu, on sélectionne la personne dans la liste fournie par LDAP, la champ "mail" se remplit automatiquement.</p> <p>Le "login" est grisé.</p> <p>Quand on valide =&gt; message d'erreur (le champ login reste inaccessible) =&gt; <b>voir mail</b></p> <p>Du coup j'ai rentré son utilisateur à la main dans la base mais ça n'est pas une solution.</p>	26/4/19
X	EMAIL	<p><b>(B) (19/4/19) Bugfix mail reçu par l' "utilisateur" d'un matériel :</b></p> <p>Jean-louis Lefort a ajouté le matériel "Optiplex 7060" (id=12111).</p> <p>Veillez vérifier et compléter si besoin la fiche correspondante.</p> <p><del>Vous recevez ce message car vous etes dans la liste spécifique des emails de LabInvent.</del></p> <p><del>Pour faire retirer votre mail de cette liste, veuillez contacter un SuperAdmin.</del></p> <p><b>(I) (1/2/19) Améliorer email sur materiel/add :</b></p> <ul style="list-style-type: none"> <li>- ajouter un lien direct vers le matériel (plutôt que son id qui ne sert à rien)</li> <li>- changer "contacter un SuperAdmin" en "contacter un administrateur du logiciel (email1, email2...)"</li> </ul> <p><i>"Etienne Pallier a ajouté le matériel 'new201901' (id=11964). Veillez vérifier et compléter si besoin la fiche correspondante. Vous recevez ce message car vous etes dans la liste spécifique des emails de LabInvent. Pour faire retirer votre mail de cette liste, veuillez contacter un SuperAdmin."</i></p> <p><b>(I) (2/4/19) Améliorer email sur materiel/delete !!! :</b></p> <p>Marjorie Cloup a supprimé le matériel "LATT - Software For Sidecar ASIC Development Kit" (id=10421).</p> <p><b>Domaine : 2</b></p> <p><b>Catégorie : 19</b></p> <p><i>Vous recevez ce message car vous etes dans la liste spécifique des emails de LabInvent. Pour faire retirer votre mail de cette liste, veuillez contacter un SuperAdmin.</i></p>	
X (fakeldapusers)	LDAP	<p><del><b>(I) (EP 4/2/19) Optimisation : optimiser la recherche LDAP</b></del></p> <p><del>Pb : Eviter de rechercher plusieurs fois les utilisateurs, ça peut prendre beq de temps à chaque fois... (surtout au GRAL).</del></p>	6/6/19

		<p><b>Solution</b> : utiliser la table Utilisateurs de la BD comme CACHE du LDAP ; à la première recherche ldap, mettre dans la table Utilisateurs de la BD les utilisateurs qui n'y sont pas encore (s'ils sont "non privilégié", les mettre avec un profil "utilisateur") ainsi que tous leurs attributs nécessaires au traitement (nom, pnom, email, ..., mais aussi ldap PASSWORD);  Mais, nouveau pb : quand on modifie ou supprime un utilisateur du ldap, il faut penser à le modifier aussi dans cette table... le risque d'incohérences est fort avec le temps...  =&gt; optimisation: avec une date de création, comme ça au bout de 3 mois par exemple, on peut rechercher à nouveau dans le ldap et mettre à jour éventuellement (sinon, les utilisateurs modifiés ou supprimés dans le ldap ne seront pas mis à jour dans la BD... =&gt; incohérence)  Plus de détails encore à la fin de ce tableau (cf (*))</p>	
<b>ETIQUETTES</b>			
X	Etiqueteuse	<p><b>(B) Imprimantes ne fonctionnent plus sur Win10, why ?...</b></p> <ul style="list-style-type: none"> <li>- ok sur Mac</li> <li>- ok sur PC win10 Etienne</li> <li>- pas ok sur pc win10 Elodie</li> <li>- pas ok sur pc win10 des gestionnaires</li> </ul>	
<b>MATERIELS</b>			
X	RECHERCHE	<p><b>(I) (28/3/19) Recherche numéro IRAP plein texte</b>  Ex: retrouver les materiels de + de 10.000 E-  Dans Labinvent, j'aimerais pouvoir chercher sur une partie du Num Irap et la, j'ai une liste deroulante. Pas top. C'est possible de remettre un champ texte ?</p> <p><b>(I) (28/3/19) avoir le prix sur la liste des materiels suite a une recherche</b></p> <p><b>(I) (12/2/19) Search Plugin</b> : <a href="https://github.com/CakeDC/search">https://github.com/CakeDC/search</a></p>	<p>2/4/19</p> <p>2/4/19</p>

X	ADD (création) et EDIT (modification) et ADMINISTRER	<p><del>(B) (25/4/19) Copie de matériel (cf Yann Parot)</del> Certains champs ne sont pas modifiables...</p> <p><del>(I) (5/2/19) Optimisation : refactoriser les vues add et edit en UNE SEULE VUE</del> Car elles se ressemblent énormément ! Eviter cette redondance !</p> <p><b>(B) Après erreur de saisie, on perd la liste des categ et sous-categ !!!</b></p> <p><b>(B) Vue “administrer” perdue ?</b> Pour la voir, taper “administrer” à la place de “view” dans une URL Si cette fonction est toujours utile, il faudrait factoriser cette action avec add_or_edit() de MaterielsController</p> <p><b>(B) (29/1/19) Edition de la section "Partie administrative" d'un matériel en profil superadmin (et autres profils ?)</b> Cette section devrait être masquée par défaut et ouverte quand on clique dessus, mais ça ne marche plus</p> <p><b>(B) (29/1/19) Ajouter champ manquant nom_ancien_responsable dans table materiels</b> Puis activer le code commenté de la fonction beforeSave() de MaterielsTable.php pour garder trace du responsable précédent ou d'un responsable parti qui n'est plus dans le ldap</p> <p><del>(B) (31/1/19) Bugfix SurCategoriesController::getFromCategorie()</del></p> <ul style="list-style-type: none"> <li>○ Request URL: /<del>SurCategories/getFromCategorie</del></li> <li>○ Referer URL: <a href="http://labinvent.test/materiels/add">http://labinvent.test/materiels/add</a></li> <li>○ Client IP: 127.0.0.1</li> <li>○ =&gt; reprogrammer différemment : obliger à choisir dans l'ordre domaine puis categ puis sous-categ : <ul style="list-style-type: none"> <li>■ <b>NEW</b> si domaine=null =&gt; mettre categ à null aussi</li> <li>■ <b>CHECK</b> si categ null =&gt; mettre souscateg à null aussi</li> <li>■ si domaine selected =&gt; changer liste categ</li> </ul> </li> </ul>	25/3/19 (EP)
			20/3/19 (EP)



		<p>■ <del>NEW</del> si categ choisie =&gt; changer liste sous-categ, mais ne pas toucher à domaine!</p>	
X	index (liste matos)	<p><b>(I) (21/5/19) Demandes Yann Parot :</b></p> <ul style="list-style-type: none"> <li>· <b>Serait-il possible dans la liste de matériel d'afficher le numéro de série ?</b> <i>possible, mais ya déjà bcp de trucs, ça risque de pas tenir sur des petits écrans pc portables (12 à 14") et le num série est rarement entré, mais s'il était entré plus souvent oui ça serait bien utile...</i></li> <li>· <b>Serait-il possible dans la liste de matériel d'afficher si il est emprunté ou non ?</b> <i>Là aussi, très utile mais trop rarement utilisé..., mais bon, à la limite c'est juste une case à cocher de plus (en mettant pour titre de colonne juste "E") Ou alors, mettre toute la ligne du matériel concerné en couleur (orange ?) On pourrait aussi se contenter de le préciser quand on va sur la fiche détaillée du matériel (style: "Attention, ce matériel est emprunté") Au même titre, il faudrait aussi signaler un suivi ou une réparation en cours (style: "Attention, ce matériel est en cours d'opération de maintenance")</i></li> <li>· <b>Serait-il possible dans le cas de création multiple de pouvoir lui faire avaler un fichier excel par exemple ?</b> <i>Oui dans la version 145..., ça c'est plutôt du rêve, mais je le note quand même dans la TODO list, on sait jamais si on trouve un super stagiaire...</i></li> </ul>	
X	Documents	<p><b>(B) <del>(15/4/19) Bugfix Doc admission:</del></b></p> <ul style="list-style-type: none"> <li><del>— Plantage quand on clique sur le bouton "Doc. admission" d'un matériel pour lequel on n'a pas précisé l'organisme ou/et le fournisseur</del></li> <li>- Si l'organisme ou le fournisseur n'est pas précisé, on ne doit pas pouvoir obtenir un doc admission. Mais la fiche materiel oui</li> </ul>	15/4/19

		<p><b>(I) (21/5/19, CRAL) Adapter les logos des documents pdf (notamment les admission*.ctp)</b></p> <p>:</p> <p>Il faudra que j'adapte en passant la liste des organismes en paramètre, via le tableau \$data de la fonction admission() du contrôleur src/Controller/DocumentsController.php</p> <p>Ce tableau \$data est lu dans admission_pdf.ctp et il suffira alors d'adapter le code de la ligne 30 et suivantes.</p> <p>Il faut aussi que tout ça soit réglable via le menu de configuration... (upload des logos)</p>	
	<b>SUIVIS</b>		
	...		
	<b>EMPRUNTS</b>		
	...		

⇒ [Retour au début du tableau](#)

**(\*) Détail sur l'action "Optimisation du LDAP" ci-dessus :**

Où se trouve le code qui affiche la liste des utilisateurs dans les vues ajout/édition de matériel ?

Il suffit de suivre cette logique :

- Ajout de matériels = src/Template/Materiels/add.ctp
- Edition de matériels = src/Template/Materiels/edit.ctp

Dans la vue "add.ctp" (et "edit.ctp") tu trouves le champ "Nom de l'utilisateur"

Il contient une variable 'options' => \$utilisateurs

C'est donc cette variable \$utilisateurs qui est utilisée.

Là ça se complique un peu :

Cette variable n'est pas définie dans la vue, mais dans le contrôleur de matériels qui l'a créée et passée à la vue :

Tu cherches donc "\$utilisateurs" dans src/Controller/MaterielsController.php, on le trouve à la ligne 811 (elle est ensuite passée à la vue à la ligne 830 avec set(..., 'utilisateurs', ...))

Elle vient de "\$users" qui est défini juste au-dessus, ligne 807 :

```
$users = TableRegistry::get('LdapConnections')->getListUsers();
```

getListUsers() est définie dans src/Model/Table/LdapConnectionsTable.php

*Si je peux modifier ce code et entrer les personnes à la main ça pourrait tourner en attendant une vraie correction?*

oui, ça pourrait suffire, mais non, car une fois que tu choisis un utilisateur dans la liste, son email est cherché via le ldap pour l'afficher dans le champ suivant nommé "Email de l'utilisateur"

En plus, c'est un code javascript (ajax) qui fait ça à la fin du fichier add.ctp :

```
$("#nom-responsable").bind("change", function(event) {  
    var url = document.URL;  
    var reg = new RegExp("(materiels).*$", "g");  
    var emailUrl = url.replace(reg, "Users/getLdapEmail/");  
    $.ajax({  
        url: emailUrl + $("#nom-responsable").val()  
    }).done(function(data) {  
        $("#email-responsable").val(data)  
    });  
});
```

Mais tu peux feinter, car ce code appelle la fonction getLdapEmail() de src/Controller/UsersController.php

Il te suffit donc de modifier aussi cette fonction... (ou alors modifier le code javascript)

Ca n'est pas bien optimal, mais comme chez nous ça prend 1 seconde, pas besoin de mieux...

### ALGO:

- De manière générale, utiliser la DB comme un "LDAP CACHE" :
  - remplacer toutes les recherches ldap par : search DB 1st then "search ldap + update DB"

Cas possibles:

- **un user se logue** => search ldap pour valider son pass
- **matos/add/edit** => search ldap pour liste des users
- **matos/index or view** => ouf, **pas de recherche dans le ldap** car le nom et email sont écrits en dur dans la table matériels

## 10.3. VERY OLD TODO LIST A VIRER BIENTOT (en italique ce qui a déjà été migré dans le tableau “TODO LIST” ci-dessus)

- (EP 26/3/19)
  - **Recherche numéro IRAP plein texte**
    - *ex: retrouver les matériels de + de 10.000 E .*
    - *Dans Labinvent, j'aimerais pouvoir chercher sur une partie du Num Irap et la, j'ai une liste déroulante. Pas top.*
    - *C'est possible de remettre un champ texte ?*
  - **Vue “administrer” perdue ?**
    - *pour la voir taper “administrer” à la place de “view” dans une URL*
    - *si cette fonction est toujours utile, il faudrait factoriser cette action avec add\_or\_edit() de MatérielsController*
  - **Changer toutes les contraintes** pour remplacer “on delete no action” par “**on delete SET NULL**”.

Déjà fait pour `matériels.fk_materiels_gestionnaire_id`, en 2 temps :

3) supprimer contrainte actuelle

```
ALTER TABLE matériels
```

```
  DROP FOREIGN KEY fk_materiels_gestionnaire_id;
```

4) Recréer contrainte

```
ALTER TABLE matériels
```

```
  ADD CONSTRAINT fk_materiels_gestionnaire_id
```

```
  FOREIGN KEY (gestionnaire_id)
```

```
  REFERENCES users (id)
```

```
  ON DELETE SET NULL ON UPDATE NO ACTION;
```

○ ~~Passer seuil de 800€ à 1000€~~

○ Imprimantes ne fonctionnent plus sur Win10, why ?...

○ (matos add) Après erreur de saisie, on perd la liste des categ et sous-categ !!!

- (EP 13/3/19) **Fichiers logo ne doivent pas être versionnés (git)**

- Il s'agit des 2 fichiers `logo_entity.jpg` et `logo_software.jpg`
  - Ils ne doivent pas être versionnés directement depuis le dossier **“destination”** `webroot/img/`
  - Ils doivent plutôt être copiés lors de la phase d'installation, depuis un dossier **“source”** (versionné) vers le dossier **“destination”** (non versionné)
  - Cela afin d'éviter que le labo qui change ces logos les change pour tout le monde !!!
- **(EP 12/2/19) Plugin Users** pour mieux gérer les users
    - <https://github.com/CakeDC/users>
    - Exemple app: <https://github.com/CakeDC/users-example>
  - **(EP 12/2/19) Search Plugin** : <https://github.com/CakeDC/search>
  - **(EP 5/2/19) Optimisation** : refactoriser les vues `add` et `edit` en **UNE SEULE VUE**, car elles se ressemblent énormément ! Eviter cette redondance facheuse !
  - **(EP 4/2/19) Optimisation** : optimiser la recherche LDAP

*Pb : Eviter de rechercher plusieurs fois les utilisateurs, ça peut prendre bcp de temps à chaque fois... (surtout au CRAL).*

*Solution : utiliser la table Utilisateurs de la BD comme CACHE du LDAP ; à la première recherche ldap, mettre dans la table Utilisateurs de la BD les utilisateurs qui n'y sont pas encore (s'ils sont “non privilégié”, les mettre avec un profil “utilisateur”) ainsi que tous leurs attributs nécessaires au traitement (nom, pnom, email, ..., mais aussi ldap PASSWORD) ;*

*Mais, nouveau pb : quand on modifie ou supprime un utilisateur du ldap, il faut penser à le modifier aussi dans cette table... le risque d'incohérences est fort avec le temps...*

*=> optimisation: avec une date de création, comme ça au bout de 3 mois par exemple, on peut rechercher à nouveau dans le ldap et mettre à jour éventuellement (sinon, les utilisateurs modifiés ou supprimés dans le ldap ne seront pas mis à jour dans la BD... => incohérence)*

*Où se trouve le code qui affiche la liste des utilisateurs dans les vues ajout/édition de matériel ?*

*Il suffit de suivre cette logique :*

- Ajout de matériels = `src/Template/Materiels/add.ctp`
- Edition de matériels = `src/Template/Materiels/edit.ctp`

*Dans la vue "add.ctp" (et "edit.ctp") tu trouves le champ "Nom de l'utilisateur"*

*Il contient une variable 'options' => \$utilisateurs*

*C'est donc cette variable \$utilisateurs qui est utilisée.*

*Là ça se complique un peu :*

*Cette variable n'est pas définie dans la vue, mais dans le contrôleur de matériels qui l'a créée et passée à la vue :*

*Tu cherches donc "\$utilisateurs" dans src/Controller/MaterielsController.php, on le trouve à la ligne 811 (elle est ensuite passée à la vue à la ligne 830 avec set(..., 'utilisateurs', ...))*

*Elle vient de "\$users" qui est défini juste au-dessus, ligne 807 :*

*\$users = TableRegistry::get('LdapConnections')->getListUsers();*

*getListUsers() est définie dans src/Model/Table/LdapConnectionsTable.php*

*Si je peux modifier ce code et entrer les personnes à la main ça pourrait tourner en attendant une vraie correction?*

*oui, ça pourrait suffire, mais non, car une fois que tu choisis un utilisateur dans la liste, son email est cherché via le ldap pour l'afficher dans le champ suivant nommé "Email de l'utilisateur"*

*En plus, c'est un code javascript (ajax) qui fait ça à la fin du fichier add.ctp :*

```
$("#nom-responsable").bind("change", function(event) {  
    var url = document.URL;  
    var reg = new RegExp("(materiels).*$", "g");  
    var emailUrl = url.replace(reg, "Users/getLdapEmail/");  
    $.ajax({  
        url: emailUrl + $("#nom-responsable").val()  
    }).done(function(data) {  
        $("#email-responsable").val(data)  
    });  
});
```

*Mais tu peux feinter, car ce code appelle la fonction getLdapEmail() de src/Controller/UsersController.php*

*Il te suffit donc de modifier aussi cette fonction... (ou alors modifier le code javascript)*

*Ca n'est pas bien optimal, mais comme chez nous ça prend 1 seconde, pas besoin de mieux...*

## ALGO:

- De manière générale, utiliser la DB comme un "LDAP CACHE":
  - remplacer toutes les recherches ldap par : search DB 1st then "search ldap + update DB"

## Cas possibles:

- **un user se logue** => search ldap pour valider son pass
- **matos/add/edit** => search ldap pour liste des users
- **matos/index or view** => ouf, **pas de recherche dans le ldap** car le nom et email sont écrits en dur dans la table matériels

- **(EP 1/2/19) Améliorer email sur materiel/add => ajouter un lien direct vers le matériel (plutôt que son id qui ne sert à rien) :**

- "Etienne Pallier a ajouté le matériel 'new201901' (id=11964).  
Veuillez vérifier et compléter si besoin la fiche correspondante.  
Vous recevez ce message car vous êtes dans la liste spécifique des emails de LabInvent. Pour faire retirer votre mail de cette liste, veuillez contacter un SuperAdmin."

- ~~**(EP 31/1/19) Bugfix** SurCategoriesController::getFromCategorie()~~

- ~~Request URL: /SurCategories/getFromCategorie~~
- ~~Referer URL: <http://labinvent.test/materiels/add>~~
- ~~Client IP: 127.0.0.1~~
- ~~=> reprogrammer différemment : obliger à choisir dans l'ordre domaine puis categ puis sous-categ :~~
  - ~~**NEW** si domaine=null => mettre categ à null aussi~~
  - ~~**CHECK** si categ null => mettre souscateg à null aussi~~
  - ~~si domaine selected => changer liste categ~~
  - ~~**NEW** si categ choisie => changer liste sous-categ, mais ne pas toucher à domaine!~~

- **(EP 31/1/19) Changer le plugin de création des QrCodes :**

- "Package aferrandini/phpqrcode is abandoned, you should avoid using it. Use endroid/qr-code instead."

- **(EP 29/1/19) Bugfix** Ajouter champ manquant nom\_ancien\_responsable dans table matériels

- puis activer le code commenté de la fonction beforeSave() de MaterielsTable.php pour garder trace du responsable précédent ou d'un responsable parti qui n'est plus dans le ldap

- (EP 29/1/19) **Bugfix** Edition de la section "Partie administrative" d'un matériel en profil superadmin (et autres profils ?)
  - Cette section devrait être masquée par défaut et ouverte quand on clique dessus, mais ça ne marche plus



# 11. Migrations de la BD

<https://book.cakephp.org/3.0/fr/migrations.html>

et

<https://book.cakephp.org/3.0/fr/phinx.html>

et

<https://blog.osmosys.asia/2017/04/17/schema-migration-in-cakephp-3-x/>

et

<https://www.sanisoft.com/blog/2014/10/20/migrations-cakephp-3-quickstart/>

## **IMPORTANT:**

Ce chapitre est dans une **forme temporaire**. C'est une **réflexion en cours** sur l'opportunité ou pas d'utiliser le plugin "migrations" pour gérer les modifications faites sur la BD. Pour l'instant, on n'utilise pas ce plugin, on gère les modifs manuellement en créant des scripts db-update.sh (dans le dossier database/update/). Il est donc **inutile de lire ce chapitre pour l'instant**, sauf si vous êtes intéressés par le sujet.

## 11.1. Procédure proposée pour garder la BD à jour suite à nos modifs (pour qu'on ait tous la même version de la BD)

**1 - Création d'un fichier de migration initial contenant tout le schéma de la BD actuelle, qui sera le schéma de référence**

**Sans doute pas nécessaire, on peut passer directement à l'étape suivante**

```
$ bin/cake bake migration_snapshot Initial
```

```
Creating file config/Migrations/20190111105729_Initial.php
```

**2 - Dump initial de la BD pour avoir une version de référence (version initiale)**

```
$ bin/cake migrations dump
```

```
using migration paths config/Migrations and seed paths config/Seeds
```

```
Writing dump file config/Migrations/schema-dump-default.lock
```

### 3 - Modification de la BD

On peut imaginer tout un tas de modifs telles que ajout ou suppression de tables ou colonnes, **mais attention CakePhp ne sait pas gérer le “renommage” de colonnes**. Dans ce cas, il faudra ajouter manuellement ces renommages dans les fichiers de migration générés.

### 4 - Générer un fichier de migration qui fait un DIFF entre la BD actuelle et la BD de référence

```
$ bin/cake bake migration_diff NameOfTheMigrations
```

Eventuellement, modifier ce fichier généré pour y ajouter manuellement les “renommages” de colonnes (car pas gérés automatiquement)

### 5 - Commiter (push) ce fichier NameOfTheMigrations sur le dépôt de référence pour que tout le monde y ait accès

```
$ git commit
```

```
$ git push
```

### 6 - Application de la migration pour être synchro

- Pour appliquer toutes les migrations :

```
$ bin/cake migrations migrate
```

*(Pour annuler ces migrations (rollback, retour en arrière): \$ bin/cake migrations rollback)*

*(# You can also pass a migration version number to rollback to a specific version:: \$ bin/cake migrations rollback -t 20150103081132)*

*(optionally up to a specific version : migrate -e development)*

- Pour appliquer une migration particulière (la dernière par exemple) :

```
# Migrate to a specific version using the ``--target`` option or ``-t`` for short.
```

```
# The value is the timestamp that is prefixed to the migrations file name::
```

```
$ bin/cake migrations migrate -t 20150103081132
```

## 11.2. Présentation du concept

Migrations est un plugin pour aider à **gérer les changements dans la base de données** en écrivant des fichiers PHP (qui peuvent d’ailleurs être versionnés par le système de gestion de version). Il permet de faire évoluer les tables au fil du temps. Au lieu d’écrire les modifications de schéma en SQL, ce plugin permet d’utiliser un ensemble intuitif de méthodes qui facilite la mise en œuvre des modifications au sein de la base de données.

Une migration est simplement un fichier PHP qui décrit les changements à effectuer sur la base de données. Un fichier de migration peut créer ou supprimer des tables, ajouter ou supprimer des colonnes, créer des index et même insérer des données dans votre base de données.

Ci-dessous un exemple de migration:

```
<?php
use Migrations\AbstractMigration;

class CreateProducts extends AbstractMigration
{
    /**
     * Change Method.
     *
     * More information on this method is available here:
     * http://docs.phinx.org/en/latest/migrations.html#the-change-method
     * @return void
     */
    public function change()
    {
        $table = $this->table('products');
        $table->addColumn('name', 'string', [
            'default' => null,
            'limit' => 255,
            'null' => false,
        ]);
        $table->addColumn('description', 'text', [
            'default' => null,
            'null' => false,
        ]);
        $table->addColumn('created', 'datetime', [
            'default' => null,
            'null' => false,
        ]);
        $table->addColumn('modified', 'datetime', [
            'default' => null,
```

```

        'null' => false,
    ]);
    $table->create();
}
}

```

Cette migration va ajouter une table à votre base de données nommée products avec les définitions de colonne suivantes:

- id colonne de type integer comme clé primaire
- name colonne de type string
- description colonne de type text
- created colonne de type datetime

*La colonne avec clé primaire nommée id sera ajoutée **implicitement**.*

*Notez que ce fichier décrit ce à quoi la base de données devrait ressembler **après** l'application de la migration. À ce stade la table products n'existe pas dans votre base de données, nous avons simplement créé un fichier qui est à la fois capable de créer la table products avec les bonnes colonnes mais aussi de supprimer la table quand une opération de rollback (retour en arrière) de la migration est effectuée.*

Une fois que le fichier a été créé dans le dossier **config/Migrations**, vous pourrez exécuter la commande migrations suivante pour créer la table dans votre base de données:

**\$ bin/cake migrations migrate**

La commande migrations suivante va effectuer un rollback (retour en arrière) et supprimer la table de votre base de données:

**\$ bin/cake migrations rollback**

Les fichiers de migrations sont stockés dans le répertoire **config/Migrations** de votre application. Le nom des fichiers de migration est précédé de la date/heure du jour de création, dans le format **YYYYMMDDHHMMSS\_MigrationName.php**

## 11.3. Exemples divers

Synchroniser la BD à partir des classes du Model (src/Model/) :

**\$ bin/cake migrations migrate**

Rollback:  
\$ bin/cake **migrations rollback**

Avec bake:  
\$ bin/cake **bake migration** CreateProducts name:string description:text created modified  
avec :

- le nom de la migration que vous allez générer (CreateProducts dans notre exemple)
- les colonnes de la table qui seront ajoutées ou retirées dans la migration (name:string description:text created modified dans notre exemple)

Créer un fichier de migration vide pour avoir un contrôle total sur ce qui doit être exécuté, en ne spécifiant pas de définition de colonnes:  
\$ bin/cake **migrations create** MyCustomMigration

## 11.4. Générer une Migration à partir d'une Base de Données Existante

<https://book.cakephp.org/3.0/fr/migrations.html#generer-une-migration-a-partir-d-une-base-de-donnees-existante>

Si on a affaire à une base de données pré-existante (labinvent) et qu'on veut commencer à utiliser migrations, ou qu'on souhaite versionner le schéma initial de la base de données, on peut exécuter la commande migration\_snapshot:

```
$ bin/cake bake migration_snapshot Initial
```

Elle va générer un fichier de migration appelé Initial contenant toutes les déclarations pour toutes les tables de la base de données. Par défaut, le snapshot va être créé en se connectant à la base de données définie dans la configuration de la connection **default** (si vous devez créer un snapshot à partir d'une autre source de données, vous pouvez utiliser l'option **--connection**). Vous pouvez aussi vous assurer que le snapshot inclut **seulement les tables pour lesquelles vous avez défini les classes de model correspondantes** en utilisant le flag **--require-table**. Quand vous utilisez ce flag, le shell va chercher les classes Table de votre application et va seulement ajouter les tables de model dans le snapshot.

Notez que quand vous créez un snapshot, il est automatiquement **marqué** (dans la table de log de phinx) **comme migré**.

⇒ Exécution faite le 11/1/19 (par EP) :

```
$ bin/cake bake migration_snapshot DB_complete
```

```
Creating file config/Migrations/20190111105729_DBComplete.php
```

```
Marking the migration 20190111105729_DBComplete as migrated... (using migration paths config/Migrations/, and seed paths config/Seeds/)
```

*Creating a dump of the new database state...*  
*Writing dump file config/Migrations/schema-dump-default.lock`...*

## 11.5. Nom de Fichier des Migrations

Les noms des migrations peuvent suivre l'une des structures suivantes:

- `(/^(Create)(.*)/)` Crée la table spécifiée.
- `(/^(Drop)(.*)/)` Supprime la table spécifiée. Ignore les arguments de champ spécifié.
- `(/^(Add).*(?:To)(.*)/)` Ajoute les champs à la table spécifiée.
- `(/^(Remove).*(?:From)(.*)/)` Supprime les champs de la table spécifiée.
- `(/^(Alter)(.*)/)` Modifie la table spécifiée. Un alias pour CreateTable et AddField.

Vous pouvez aussi utiliser la `_forme_avec_underscores` comme nom pour vos migrations par exemple `create_products`.

## 11.6. Définition de Colonnes

Quand vous définissez des colonnes avec la ligne de commande, il peut être pratique de se souvenir qu'elles suivent le modèle suivant:  
`fieldName:fieldType?[length]:indexType:indexName`

Par exemple, les façons suivantes sont toutes des façons valides pour spécifier un champ d'email:

- `email:string?`
- `email:string:unique`
- `email:string?[50]`
- `email:string:unique:EMAIL_INDEX`
- `email:string[120]:unique:EMAIL_INDEX`

Le point d'interrogation qui suit le type du champ entrainera que la colonne peut être null.  
Le paramètre `length` pour `fieldType` est optionnel et doit toujours être écrit entre crochets.

Les champs nommés `created` et `modified`, tout comme les champs ayant pour suffixe `_at`, vont automatiquement être définis avec le type `datetime`.

Les types de champ sont ceux qui sont disponibles avec la librairie Phinx. Ce sont les suivants:

- `string`
- `text`
- `integer`
- `biginteger`
- `float`
- `decimal`
- `datetime`
- `timestamp`
- `time`
- `date`
- `binary`
- `boolean`
- `uuid`

Il existe quelques heuristiques pour choisir les types de champ quand ils ne sont pas spécifiés ou définis avec une valeur invalide. Par défaut, le type est `string`:

- `id`: `integer`
- `created`, `modified`, `updated`: `datetime`

## 11.7. Créer une Table

Vous pouvez utiliser `bake` pour créer une table:

```
$ bin/cake bake migration CreateProducts name:string description:text created modified
```

La ligne de commande ci-dessus va générer un fichier de migration qui ressemble à:

```
<?php
```

```
use Migrations\AbstractMigration;
```

```
class CreateProducts extends AbstractMigration
```

```

{
  /**
   * Change Method.
   *
   * More information on this method is available here:
   * http://docs.phinx.org/en/latest/migrations.html#the-change-method
   * @return void
   */
  public function change()
  {
    $table = $this->table('products');
    $table->addColumn('name', 'string', [
      'default' => null,
      'limit' => 255,
      'null' => false,
    ]);
    $table->addColumn('description', 'text', [
      'default' => null,
      'null' => false,
    ]);
    $table->addColumn('created', 'datetime', [
      'default' => null,
      'null' => false,
    ]);
    $table->addColumn('modified', 'datetime', [
      'default' => null,
      'null' => false,
    ]);
    $table->create();
  }
}

```

## 11.8. Ajouter des Colonnes à une Table Existante



Si le nom de la migration dans la ligne de commande est de la forme « AddXXXToYYY » et est suivie d'une liste de noms de colonnes et de types alors un fichier de migration contenant le code pour la création des colonnes sera généré:

```
$ bin/cake bake migration AddPriceToProducts price:decimal
```

L'exécution de la ligne de commande ci-dessus va générer:

```
<?php
use Migrations\AbstractMigration;

class AddPriceToProducts extends AbstractMigration
{
    public function change()
    {
        $table = $this->table('products');
        $table->addColumn('price', 'decimal')
        ->update();
    }
}
```

## 11.9. Générer un diff entre deux états de base de données

Nouveau dans la version cakephp/migrations: 1.6.0

Vous pouvez générer un fichier de migrations qui regroupera toutes les différences entre deux états de base de données en utilisant le template `bake migration_diff`. Pour cela, vous pouvez utiliser la commande suivante:

```
$ bin/cake bake migration_diff NameOfTheMigrations
```

Pour avoir un point de comparaison avec l'état actuel de votre base de données, le **shell migrations va générer, après chaque appel de migrate ou rollback un fichier « dump »**. Ce fichier dump est un fichier qui contient l'ensemble de l'état de votre base de données à un point précis dans le temps.

**Quand un fichier dump a été généré, toutes les modifications que vous ferez directement dans votre SGBD seront ajoutées au fichier de migration qui sera généré quand vous appellerez la commande `bake migration_diff`.**

Si vous souhaitez utiliser la fonctionnalité de diff sur une application qui possède déjà un historique de migrations, vous allez avoir besoin de créer le fichier dump manuellement pour qu'il puisse être utilisé comme point de comparaison:

## **\$ bin/cake migrations dump**

L'état de votre base de données devra être le même que si vous aviez migré tous vos fichiers de migrations avant de créer le fichier dump. Une fois que le fichier dump est créé, vous pouvez opérer des changements dans votre base de données et utiliser la commande `bake migration_diff` quand vous voulez.

**ATTENTION: le système n'est pas capable de détecter les colonnes renommées.**

## **11.10. Les Commandes**

### **11.10.1. migrate : Appliquer les Migrations**

Une fois que vous avez généré ou écrit votre fichier de migration, vous devez exécuter la commande suivante pour appliquer les modifications à votre base de données:

```
# Exécuter toutes les migrations  
$ bin/cake migrations migrate
```

### **11.10.2. status : Statuts de Migrations**

La commande `status` affiche une liste de toutes les migrations, ainsi que leur état actuel. Vous pouvez utiliser cette commande pour déterminer les migrations qui ont été exécutées:

```
$ bin/cake migrations status
```

### 11.10.3. seed : Remplir votre Base de Données (Seed)

Depuis la version 1.5.5, vous pouvez utiliser le shell migrations pour remplir votre base de données.  
Par défaut, les fichiers de seed vont être recherchés dans le répertoire config/Seeds de votre application.

Une interface bake est fournie pour les fichiers de seed:

```
# Ceci va créer un fichier ArticlesSeed.php dans le répertoire config/Seeds
# de votre application
# Par défaut, la table que le seed va essayer de modifier est la version
# "tableized" du nom de fichier du seed
```

**\$ bin/cake bake seed Articles**

⇒ **Ca ne copie pas les données, ça crée seulement un “seeder” que l’on peut customiser pour remplir la BD, mais par défaut, il n’insèrera aucune donnée dans la base**

**Ex: table “configurations”**

```
$ bin/cake bake seed Configurations
```

*Creating file **config/Seeds/ConfigurationsSeed.php***

```
<?php
use Migrations\AbstractSeed;

/**
 * Configurations seed.
 */
class ConfigurationsSeed extends AbstractSeed
{
    /**
     * Run Method.
     *
     * Write your database seeder using this method.
     *
     * More information on writing seeds is available here:
     * http://docs.phinx.org/en/latest/seeding.html
     *
     * @return void
     */
    public function run()
```

```

{
    $data = [];

    $table = $this->table('configurations');
    $table->insert($data)->save();
}
}

```

Les options **--data**, **--limit** and **--fields** ont été ajoutées pour permettre d'exporter des données extraites depuis votre base de données. A partir de 1.6.4, la commande `bake seed` vous permet de créer des fichiers de seed avec des lignes exportées de votre base de données en utilisant l'option **--data**:

**\$ bin/cake bake seed --data Articles**

Par défaut, cela exportera toutes les lignes trouvées dans la table.

**Ex: table "configurations"**

**\$ bin/cake bake seed --data Configurations**

*Creating file **config/Seeds/ConfigurationsSeed.php***

```

<?php
use Migrations\AbstractSeed;

/**
 * Configurations seed.
 */
class ConfigurationsSeed extends AbstractSeed
{
    /**
     * Run Method.
     *
     * Write your database seeder using this method.
     *
     * More information on writing seeds is available here:
     * http://docs.phinx.org/en/latest/seeding.html
     *
     * @return void
     */
}

```

```

public function run()
{
$data = [
[
    'id' => '1',
    'nom' => 'default',
    'mode_install' => '0',
    'mode_debug' => '0',
    'labNameShort' => 'IRAP',
    'labPresent' => 'de l"',
    'labUmr' => 'UMR 5277',
    'hasPrinter' => '1',
    'nom_groupe_thematique' => 'Groupe thematique',
    'nom_groupe_metier' => 'Groupe metier',
    'envoi_mail' => '1',
    'envoi_mail_guests' => '1',
    'emailGuest1' => 'inventirap@labo',
    'emailGuest2' => "",
    'emailGuest3' => "",
    'emailGuest4' => "",
    'emailGuest5' => 'etienne.pallier@labo',
    'test' => NULL,
    'prix_inventaire_administratif' => '800',
    'emailGuest6' => 'epallier@labo',
    'emailGuest7' => 'etienne.pallier@perso',
    'emailGuest8' => "",
    'emailGuest9' => "",
    'emailGuest10' => "",
    'sender_mail' => 'inventirap@labo',
    'labName' => 'Institut de Recherche en Astrophysique et Planétologie',
    'date_commande_facultative' => '0',
    'numero_labosans_annee' => '0',
    'taille_max_doc' => '8000000',
    'metrologie' => '1',
    'aff_par_defaut' => '20',
    'procedure_sur_accueil' => '1',
    'ldap_used' => '0',

```

```

        'ldap_authenticated' => '1',
        'ldap_bindDn' => "",
        'ldap_bindPass' => "",
        'ldap_host' => 'ldaps://ldap2.labo',
        'ldap_port' => 'port',
        'ldap_authenticationType' => 'uid',
        'ldap_baseDn' => '...',
        'ldap_filter' => '...',
    ],
];






```

## Appliquer un seed pour alimenter la BD

Pour faire un seed de votre base de données, vous pouvez utiliser la sous-commande **seed**:

# **Sans paramètres**, la sous-commande seed va exécuter tous les seeders

# disponibles du répertoire cible, dans l'ordre alphabétique.

**\$ bin/cake migrations seed**

# Vous pouvez **spécifier seulement un seeder** à exécuter en utilisant l'option `--seed`

**\$ bin/cake migrations seed --seed ArticlesSeed**

Ex: on supprime les données de la table "configurations", puis on les remet avec "migrations seed" en utilisant le seed

**ConfigurationsSeed** créé ci-dessus, easy !!! :

(from PhpMyadmin) **delete from configurations;**

**\$ bin/cake migrations seed --seed ConfigurationsSeed**

Notez que, **à l'opposé des migrations, les seeds ne sont pas suivis**, ce qui signifie que le même seeder peut être appliqué plusieurs fois.

## 11.11. Trucs et Astuces

### 11.11.1. Mettre à jour les Noms de Colonne et Utiliser les Objets Table

Si vous utilisez un objet Table de l'ORM de CakePHP pour manipuler des valeurs de votre base de données, comme renommer ou retirer une colonne, assurez-vous de créer une nouvelle instance de votre objet Table après l'appel à update(). Le registre de l'objet Table est nettoyé après un appel à update() afin de rafraîchir le schéma qui est reflété et stocké dans l'objet Table lors de l'instanciation.

### 11.11.2. Migrations et déploiement

Si vous utilisez le plugin dans vos processus de déploiement, assurez-vous de vider le cache de l'ORM pour qu'il renouvelle les `_metadata_` des colonnes de vos tables. Autrement, vous pourriez rencontrer des erreurs de colonnes inexistantes quand vous effectuerez des opérations sur vos nouvelles colonnes. Le Core de CakePHP inclut un [Shell de Cache du Schéma](#) que vous pouvez utiliser pour vider le cache:

```
// Avant 3.6, utilisez orm_cache
$ bin/cake schema_cache clear
```

Veillez vous référer à la section du cookbook à propos du [Shell du Cache du Schéma](#) si vous voulez plus de détails à propos de ce shell.

### 11.11.3. Renommer une table

Le plugin vous donne la possibilité de renommer une table en utilisant la méthode rename(). Dans votre fichier de migration, vous pouvez utiliser la syntaxe suivante:

```
public function up()
{
    $this->table('old_table_name')
        ->rename('new_table_name');
}
```

### 11.11.4. Ne pas générer le fichier schema.lock

Nouveau dans la version cakephp/migrations: 1.6.5

Pour que la fonctionnalité de « diff » fonctionne, un fichier **.lock** est généré à chaque fois que vous faites un migrate, un rollback ou que vous générez un snapshot via bake pour permettre de suivre l'état de votre base de données à n'importe quel moment. Vous pouvez empêcher que ce fichier ne soit généré, comme par exemple lors d'un déploiement sur votre environnement de production, en utilisant l'option --no-lock sur les commandes mentionnées ci-dessus:

```
$ bin/cake migrations migrate --no-lock
```

```
$ bin/cake migrations rollback --no-lock
```

```
$ bin/cake bake migration_snapshot MyMigration --no-lock
```



# 12. Gestion du temps (DATE & TIME) et localisation

\date \time

Références :

- <https://www.dereuromark.de/2016/08/12/chronos-let-there-be-time/>
- TimeHelper pour les vues : <https://book.cakephp.org/4/fr/views/helpers/time.html>
- [Date & Time \(cakephpv4 remplace Time par FrozenTime\)](#)
- [Date & Time \(cakephpv3 avec Time\)](#)

Voir aussi : <https://stackoverflow.com/questions/42067742/cakephp-3-datetime-validation-iso-8601-with-timezone>

## 12.1. Contexte général (pour votre culture, mais vous pouvez zapper)

Depuis 2016 (Cakephp 3.2):

- In PHP most know of **DateTime** class to handle date and time. At least with more modern PHP versions **it is now not advised anymore to use the plain old date() and time() functions.**
- Cakephp passe de l'utilisation de **Carbon** (une amélioration de DateTime) à **Chronos** (un clone de Carbon réalisé pour cakephp) de meilleure qualité que Carbon.

### 12.1.1. Main differences and improvements from Carbon to Chronos :

- Implements ChronosInterface for **proper typehinting**, e.g. in methods
- **DateTime** and **Date (no time) handling separately** per use case
- **Immutable by default** for cleaner coding and less errors
- Correct some faulty standards (ISO code violations) and behavior (difference calculation)
- No external dependencies
  
- **Immutable c'est mieux :**

```
// Bad practice - and doesn't work with immutable objects
$datetime->addDay(1);
$this->doSomething($datetime);
return $datetime;
```

```
// Better to never touch the original object - this works like you'd expect
$datetime = $datetime->addDay(1);
$datetime = $this->doSomething($datetime);
return $datetime;
```

- **Ajouter 1 mois correctement :**

Intuitively, if you add months instead of specific days to a date, you would expect this to be "month-exact", not "day-exact".

```
$dt = new DateTime('2015-01-31');
$dt->modify("+1 month");
echo $dt->format("Y-m-d H:i:s"); //2015-03-03 00:00:00
```

Clearly, this overflows in unexpected ways.

So Chronos actually gives you `addMonths()/subMonths()` that actually work as desired:

```
$dt = new Chronos('2015-01-31');
$dt = $dt->addMonths(1);
echo $dt->format("Y-m-d H:i:s"); //2015-02-28 00:00:00
```

*To get the former PHP behavior back, you can explicitly use `addMonthsWithOverflow()/subMonthsWithOverflow()` methods. Not that is is ever useful or recommended*

### 12.1.2. Testing and fixating time

Everyone knows those one second issues when writing tests and `(date)time`. Sometimes tests fail because the time for `now()` jumped to the next second.

When writing unit tests, it is helpful to fixate the current time. Chronos lets you fix the current time for each class. As part of your test suite's bootstrap process you can include the following:

```
Chronos::setTestNow(Chronos::now());
MutableDateTime::setTestNow(MutableDateTime::now());
Date::setTestNow(Date::now());
MutableDate::setTestNow(MutableDate::now());
```

This will fix the current time of all objects to be the point at which the test suite started.

### 12.1.3. Usage

The Chronos API offers a very fast and intuitive way to work with datetime.

Let's say you want to find the next Tuesday, if the current one is not already one:

```
$dt = new Chronos('2015-01-31');
if (!$dt->isTuesday()) {
    $dt = $dt->next(ChronosInterface::TUESDAY);
}
```

Quite convenient are also the checks to find out whether a date is in the past or the future:

```
$dt->isPast();
$dt->isFuture();
```

Of course, you could also use a more verbose way with `gt()/lt()` and a current "now" datetime.

Check out the official [chronos docs](#) for how to use it in general.

#### 12.1.3.1. Usage in frameworks

Usually, frameworks should be able to switch inside DB layer from DateTime or Carbon to Chronos easily.

In CakePHP for example the type conversation is setup in the bootstrap, and it already uses the immutable Chronos objects by default:

```
// bootstrap.php
/**
 * Enable immutable time objects in the ORM.
 *
 * You can enable default locale format parsing by adding calls
 * to `useLocaleParser()`. This enables the automatic conversion of
 * locale specific date formats. For details see
 * @link http://book.cakephp.org/3.0/en/core-libraries/internationalization-and-localization.html#parsing-localized-datetime-data
 */
Type::build('time')
    ->useImmutable();
Type::build('date')
    ->useImmutable();
Type::build('datetime')
    ->useImmutable();
```

The underlying classes here extend Carbon and so in the ORM all ingoing and outgoing datetimes are Chronos objects.

## 12.2. Et dans LabInvent ?

Voir <https://book.cakephp.org/3/en/core-libraries/internationalization-and-localization.html#localizing-dates-and-numbers>

- **Dans src/config/bootstrap.php : on est passé en heure locale (FR)**

```
/**
 * Set server timezone to UTC. You can change it to another timezone of your
 * choice but using UTC makes time calculations / conversions easier.
```

```
*/  
//date_default_timezone_set('UTC'); // GMT+0  
date_default_timezone_set('Europe/Paris');
```

- **Dans src/config/app.php : on est passé en heure locale (FR)**

```
'App' => [  
    'namespace' => 'App',  
    'encoding' => env('APP_ENCODING', 'UTF-8'),  
    'defaultLocale' => env('APP_DEFAULT_LOCALE', 'fr_FR'),  
    'base' => false,  
    'dir' => 'src',  
    'webroot' => 'webroot',  
    'wwwRoot' => WWW_ROOT,  
    // 'baseUrl' => env('SCRIPT_NAME'),  
    'fullBaseUrl' => false,  
    'imageBaseUrl' => 'img/',  
    'cssBaseUrl' => 'css/',  
    'jsBaseUrl' => 'js/',  
    'paths' => [  
        'plugins' => [ROOT . DS . 'plugins' . DS],  
        'templates' => [APP . 'Template' . DS],  
        'locales' => [APP . 'Locale' . DS],  
    ],  
],
```

This will control several aspects of the application, including the default translations language, the date format, number format and currency whenever any of those is displayed using the localization libraries that CakePHP provides.

# 13. Le framework CakePhp

\cakephp \essentiel \cheatsheet \synthese  
(EP updated 07/05/20)

Communautés : <https://plus.google.com/communities/108328920558088369819>

## 13.1. Partage de logique dans les modèles, les vues, et les contrôleurs

En vert les composants déjà utilisés dans ce projet

Couche MVC	Partage de logique (mise en commun)	Classe (fichier)
Modèle	<b>Behaviors</b> <a href="https://book.cakephp.org/4/en/orm/behaviors.html">https://book.cakephp.org/4/en/orm/behaviors.html</a> Comportements partagés entre N modèles (leurs méthodes s'ajoutent à un objet Table comme un mixin)	<b>src/Model/Behavior/NomClasseBehavior.php</b>  <u>Définition :</u> class NomClasse <b>Behavior</b> extends <b>Behavior</b>  <u>Utilisation:</u> class ArticlesTable extends <b>Table</b> { public function initialize(array \$config): void { \$this-> <b>addBehavior</b> ('Timestamp'); } }
	<b>Virtual Field</b> Rendre accessible à toutes les vues d'un modèle un	<b>src/Model/Entity/Emprunt.php</b>

	<p><b>traitement commun effectué sur certains champs</b> de la table</p> <p>Comment ? en <b>ajoutant un nouveau champ</b> au modèle (table) qui est <b>calculé</b> à partir des autres champs</p> <p>Exemple : un champ “statut” pour la table Emprunts qui affiche “en cours”, “terminé” ou “à venir”, en fonction des dates emprunt et retour</p>	<pre>class Emprunt extends Entity { protected function _getStatusFromDates() { ... return ... } =&gt; ce champ virtuel peut être lu dans toutes les vues de Emprunt en tant que \$status_from_dates</pre>
<p><b>Vue (View)</b></p>	<p><b>Helpers</b> Logique de présentation partagée entre N Views, Elements, ou Layouts</p> <p><b>Elements</b> Elements répétés dans différentes vues (composants) tels qu’un menu, un formulaire de login, une boîte d’aide, ...</p> <p><b>Cells</b> (plus haut niveau que Elements, plus complexe) Small mini-controllers that can invoke view logic and render out templates. Ideal for building reusable page components that require interaction with models, view logic, and rendering logic. A simple example would be the <b>cart in an online store, or a data-driven navigation menu in a CMS.</b></p>	<p><b>src/View/Helper/NomClasseHelper.php</b></p> <p><u>Définition</u> :</p> <pre>class NomClasseHelper extends Helper</pre> <p><u>Utilisation</u>:</p> <pre>class AppView extends View { public function initialize(): void { parent::initialize(); \$this-&gt;loadHelper('Html'); \$this-&gt;loadHelper('Form'); \$this-&gt;loadHelper('Flash'); } }</pre>
<p><b>Contrôleur</b></p>	<p><b>Components</b> <a href="https://book.cakephp.org/4/en/controllers/components.html">https://book.cakephp.org/4/en/controllers/components.html</a></p> <p>Packages of logic shared between controllers</p>	<p><b>src/Controller/Component/NomClasseComponent.php</b></p> <p><u>Définition</u> :</p> <pre>class NomClasseComponent extends Component</pre> <p><u>Utilisation</u>:</p> <pre>class PostsController extends AppController { public function initialize(): void {</pre>

		<pre>parent::initialize(); \$this-&gt;loadComponent('Flash'); }  public function delete() {     if (\$this-&gt;Post-&gt;delete(\$this-&gt;request-&gt;getData('Post.id')) {         \$this-&gt;Flash-&gt;success('Post deleted.');</pre> <pre>        return \$this-&gt;redirect(['action' =&gt; 'index']);     } }</pre>
--	--	---



## Helpers standards fournis par CakePhp

<b>Html</b>	<a href="https://book.cakephp.org/4/en/views/helpers/html.html">https://book.cakephp.org/4/en/views/helpers/html.html</a>
<b>Time</b>	<p><a href="https://book.cakephp.org/3/fr/core-libraries/time.html#Cake\18n\Time">https://book.cakephp.org/3/fr/core-libraries/time.html#Cake\18n\Time</a></p> <p><a href="https://book.cakephp.org/4/en/views/helpers/time.html">https://book.cakephp.org/4/en/views/helpers/time.html</a></p> <p><a href="https://api.cakephp.org/3.8/class-Cake.18n.Time.html#_i18nFormat">https://api.cakephp.org/3.8/class-Cake.18n.Time.html#_i18nFormat</a> <b>Time</b> class : extends the built-in <b>DateTime</b> class to provide handy methods and locale-aware formatting helpers</p> <p><a href="https://api.cakephp.org/3.8/class-Cake.18n.Date.html">https://api.cakephp.org/3.8/class-Cake.18n.Date.html</a> Date class : extends the Date class provided by <a href="#">Chronos</a></p> <p><a href="https://book.cakephp.org/4/en/core-libraries/time.html">https://book.cakephp.org/4/en/core-libraries/time.html</a></p>
<b>Numbers</b>	<a href="https://book.cakephp.org/4/en/views/helpers/number.html">https://book.cakephp.org/4/en/views/helpers/number.html</a>
<b>Url</b>	<a href="https://book.cakephp.org/4/en/views/helpers/url.html">https://book.cakephp.org/4/en/views/helpers/url.html</a>
<b>Text</b>	<a href="https://book.cakephp.org/4/en/views/helpers/text.html#linking-both-urls-and-email-addresses">https://book.cakephp.org/4/en/views/helpers/text.html#linking-both-urls-and-email-addresses</a>
<b>Form</b>	<a href="https://book.cakephp.org/4/en/views/helpers/form.html">https://book.cakephp.org/4/en/views/helpers/form.html</a>

## 13.2. Le fichier Application (Application.php)

\application

since: v3.3

<https://book.cakephp.org/3/en/appendices/3-3-migration-guide.html#psr-7-middleware-support-added>

Info :

<https://book.cakephp.org/4/fr/development/application.html>

Fichier (squelette) :

<https://github.com/cakephp/app/blob/master/src/Application.php>

## 13.3. Mon CakePhp Cheat sheet

\cheat \sheet \synthese

(voir aussi <https://alvinalexander.com/php/cakephp-cheat-sheet-reference-page-examples/>)

Saving data :

<https://book.cakephp.org/4/en/orm/saving-data.html>

Besoin	Solution
Récupérer une <b>Table</b> (un <b>Model</b> )	(dans Table ou Controller) use Cake\ORM\TableRegistry; <b>\$articlesTable = TableRegistry::getTableLocator()-&gt;get('Articles');</b> // shortcut pour la suite <b>\$t = \$articlesTable</b>
<u>Créer une Entity</u>	use App\Model\Entity\Article; // vide \$e = <b>new Article()</b> ; // \$e pour faire court (e comme entité), on devrait plutôt utiliser <b>\$article</b> // avec data : \$e = new Article([ 'id' => 1, 'title' => 'New Article', 'created' => new DateTime('now') ]);  <b>OU BIEN</b>  // vide \$e = <b>\$t-&gt;newEmptyEntity()</b> ; // on pourrait aussi utiliser <b>newEntity()</b> // avec data : \$e = <b>\$t-&gt;newEntity()</b> ([ 'id' => 1, 'title' => 'New Article', 'created' => new DateTime('now')

	<pre>]);</pre> <p>⇒ \$e est une instance de App\Model\Entity\Article ou bien par défaut une instance de Cake\ORM\Entity si on n'a pas créé de classe Article.</p>
<p><a href="#">(add) ou (edit)</a></p> <p><b>(add) Ajouter 1 entité en BD</b></p> <p><b>(edit) Modifier 1 entité en BD</b></p>	<p>CakePHP saura s'il doit faire un ajout ou une mise à jour en se basant sur le résultat de la méthode <b>isNew()</b>. Les entités qui sont récupérées via <b>get()</b> ou <b>find()</b> renverrons toujours <b>false</b> lorsque la méthode <b>isNew()</b> est appelée sur eux.</p> <pre>\$e = \$t-&gt;newEmptyEntity(); \$e-&gt;title = 'A New Article'; \$e-&gt;body = 'This is the body of the article'; if (\$t-&gt;save(\$article))     \$id = \$e-&gt;id; // The \$article entity contains the id now</pre> <pre>\$e = \$t-&gt;get(12); // Return article with id 12 \$e-&gt;title = 'CakePHP is THE best PHP framework!'; \$t-&gt;save(\$article);</pre>
<p><a href="#">Enregistrements avec Associations</a></p>	<p><i>Par défaut, la méthode save() ne sauvegardera qu'un seul niveau d'association</i></p> <pre>\$author = \$t-&gt;Authors-&gt;findByUserName('mark')-&gt;first(); \$e = \$t-&gt;newEmptyEntity(); \$e-&gt;title = 'An article by mark'; \$e-&gt;author = \$author; // shortcut pour \$e-&gt;author_id = \$author-&gt;id if (\$t-&gt;save(\$e)) {     // The foreign key value was set automatically.     echo \$e-&gt;author_id; }  // Sauvegarde seulement l'association avec les commentaires \$t-&gt;save(\$entity, ['associated' =&gt; ['Comments']]); // Sauvegarde la company, les employees et les addresses liées pour chacun d'eux. \$companies-&gt;save(\$entity, ['associated' =&gt; ['Employees.Addresses']]); // Si on construit ou modifie une donnée d'association après avoir construit l'entity, la marquer comme dirty \$company-&gt;author-&gt;name = 'Master Chef';</pre>

```
$company->setDirty('author', true);
```

La méthode **save()** est également capable de **créer de nouveaux enregistrements pour les associations**:

```
$c1 = $t->Comments->newEmptyEntity();  
$c1->body = 'Un super article';  
$c2 = $t->Comments->newEmptyEntity();  
$c2->body = "J'aime lire ceci!";
```

```
$tag1 = $t->Tags->findByName('cakephp')->first();  
$tag2 = $t->Tags->newEmptyEntity();  
$tag2->name = 'Génial';
```

```
$e = $t->get(12);
```

#### **// SUPER SHORTCUTS:**

```
$e->comments = [$c1, $c2]; // Articles 1-N Comments  
$e->tags = [$tag1, $tag2]; // Articles M-N Tags  
$t->save($e);
```

OU BIEN

```
$t->Tags->link($e, [$tag1, $tag2]);
```

Pour dissocier des tags :

```
$tags = $t  
->Tags  
->find()  
->where(['name IN' => ['cakephp', 'awesome']])  
->toList();  
$t->Tags->unlink($e, $tags);
```

Plus technique encore si besoin :

<https://book.cakephp.org/4/fr/orm/saving-data.html#sauvegarder-des-donnees-dans-la-table-de-jointure>

#### **Conversion données**

Récupération des données requêtées dans \$data :

## formulaire => Entity

```
$data = $this->request->getData();  
// Par défaut, la méthode newEntity() valide les données qui lui sont passées  
// Pour ne pas valider : newEntity($data, ['validate' => false]);  
$e = $t->newEntity($data);
```

Les données requêtées doivent suivre la structure de vos entités. Par exemple si vous avez un article qui appartient à un utilisateur, et plusieurs commentaires, \$data devrait ressembler à ceci:

```
$data = [  
  'title' => 'My title',  
  'body' => 'The text',  
  'user_id' => 1,  
  'user' => [  
    'username' => 'mark'  
  ],  
  'comments' => [  
    ['body' => 'First comment'],  
    ['body' => 'Second comment'],  
  ]  
];
```

Lors de la construction de formulaires qui sauvegardent des associations imbriquées, vous devez **définir quelles associations doivent être prises en compte**:

```
// Nouvelle entity avec des associations imbriquées  
$e = $t->newEntity($data, [  
  'associated' => ['Tags', 'Comments.Users']  
]);
```

⇒ les “Tags”, “Comments” et “Users” pour les Comments doivent être prises en compte.

On peut désactiver le marshalling d’associations imbriquées comme ceci:

```
$e = $t->newEntity($data, ['associated' => []]);  
// avec patchEntity :  
$e = $t->patchEntity($e, $data, ['associated' => []]);
```

Données 1-N  
(HasMany)

```
$data = [  
  'title' => 'Mon titre',  
  'body' => 'Le texte',  
  'comments' => [  
    // liens vers commentaires existants :  
    ['id' => 1, 'comment' => 'Mettre à jour le premier commentaire'],  
    ['id' => 2, 'comment' => 'Mettre à jour le deuxième commentaire'],  
    // création nouveau commentaire :  
    ['comment' => 'Créer un nouveau commentaire'],  
  ]  
];
```

Données M-N  
(BelongsToMany)

```
$data = [  
  'title' => 'My title',  
  'body' => 'The text',  
  'user_id' => 1,  
  'tags' => [  
    // création de 2 nouveaux tags :  
    ['name' => 'A new tag'],  
    ['name' => 'Another new tag'],  
    // liens vers tags existants :  
    ['id' => 5],  
    ['id' => 21]  
  ]  
];
```

Convertir données  
**MULTIPLES** (plusieurs  
entités d'un coup)

```
$data = [  
  // entité 1:  
  [  
    'title' => 'First post',  
    'published' => 1  
  ],  
  // entité 2:
```

```

[
  'title' => 'Second post',
  'published' => 1
],
];
$entities = $t->newEntities($data);

// save ou delete chaque entité, une par une...
foreach ($entities as $e) {
  $t->save($e);
  // ou bien
  $t->delete($e);
}
// ... ou bien avec une transaction unique :
$t->getConnection()->transactional(function() use ($t, $entities) {
  foreach ($entities as $e) $t->save($e, ['atomic' => false]);
});

// ... ou bien en une seule fois (TOUS d'un coup, de manière atomique)
// Get a boolean indicating success
$t->saveMany($entities);
// Get a PersistenceFailedException if any records fail to save.
$t->saveManyOrFail($entities);

```

(edit) [Mettre à jour](#) une entité (**patcher**)

```

$e = $t->patchEntity($e, $data);
// ajouter ['validate' => false] pour ne pas valider les données
// ajouter [ 'fieldList' => ['title', 'tags']] pour ne modifier QUE cette liste de champs (y-compris associations)
// ajouter [ 'associated' => ['Tags' => ['fieldList' => ['name']]] ] pour autoriser seulement le nom du tag
$t->save($e);
// L'option fieldList est aussi acceptée par les méthodes newEntity(), newEntities() et patchEntities()

```

- **Patcher des HasMany et**

```

$associated = ['Tags', 'Comments.Users'];
$e = $t->get(1, ['contain' => $associated]);

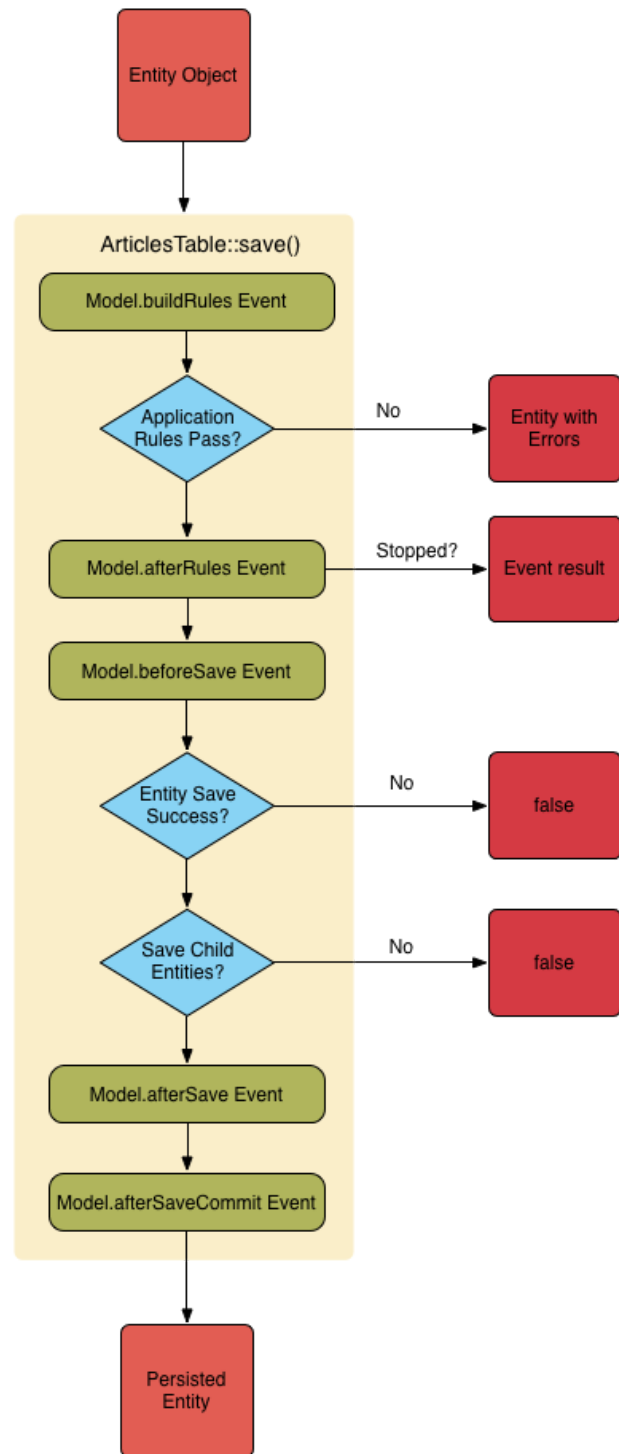
```



<p><b>BelongsToMany</b></p>	<pre>\$t-&gt;patchEntity(\$e, \$data, [     'associated' =&gt; \$associated ]); \$t-&gt;save(\$e);  // Faire un patch d'une entity sans entity dans la propriété user va créer une nouvelle entity user: \$data = [     'title' =&gt; 'My title',     'user' =&gt; [         'username' =&gt; 'mark'     ] ]; \$e = \$t-&gt;patchEntity(new Article, \$data); echo \$e-&gt;user-&gt;username; // affiche 'mark'</pre>
<p>- <b>Patcher plusieurs entities d'un coup</b></p>	<pre>\$list = \$t-&gt;find('popular')-&gt;toArray(); // on pourrait ajouter ['associated' =&gt; ['Tags', 'Comments.Users']] pour contrôler les associations fusionnées \$patched = \$t-&gt;patchEntities(     \$list, \$data,     ['associated' =&gt; ['Tags', 'Comments.Users']] // uniquement si on veut contrôler les associations à fusionner ); foreach (\$patched as \$e) \$t-&gt;save(\$e);</pre>
<p>- <b>Mise à jour en masse (bulk update) (updateAll)</b></p>	<pre>// Publie tous les articles non publiés. function publishAllUnpublished() {     \$this-&gt;updateAll(         [ // fields             'published' =&gt; true,             'publish_date' =&gt; FrozenTime::now()         ],         [ // conditions             'published' =&gt; false         ]     ); }</pre> <p>Si on doit <b>utiliser des expressions SQL</b>, utiliser un objet expression :</p>

	<pre> use Cake\Database\Expression\QueryExpression; ... function incrementCounters() {     \$expression = new QueryExpression('view_count = view_count + 1');     \$this-&gt;updateAll([\$expression], ['published' =&gt; true]); } </pre> <p><i>updateAll() est uniquement une fonction de commodité. On peut aussi utiliser cette interface plus flexible:</i>  // Publication de tous les articles non publiés.</p> <pre> function publishAllUnpublished() {     \$this-&gt;query()         -&gt;update()         -&gt;set(['published' =&gt; true])         -&gt;where(['published' =&gt; false])         -&gt;execute(); } </pre>
<p><b><u>Modifier les Données Requêtées Avant de Construire les Entities (beforeMarshal)</u></b></p>	<p>// Dans une classe <b>Table</b> ou <b>Behavior</b></p> <pre> use Cake\Event\Event; use ArrayObject; public function beforeMarshal(Event \$event, ArrayObject \$data, ArrayObject \$options) {     // suppression des espaces superflus des champs avant l'enregistrement :     foreach (\$data as \$key =&gt; \$value) {         if (is_string(\$value)) {             \$data[\$key] = trim(\$value);         }     }     // mise en minuscule du nom :     if (isset(\$data['username'])) {         \$data['username'] = mb_strtolower(\$data['username']);     } } </pre> <p><i>Le but principal de beforeMarshal est d'aider les utilisateurs à passer le process de validation lorsque des erreurs simples peuvent être résolues automatiquement, ou lorsque les données doivent être restructurées pour être mises dans les bons champs. Il est lancé juste au début du process de validation. Une des raisons</i></p>

	<p>à cela est que <code>beforeMarshal</code> est autorisé à modifier les règles de validation et les options d'enregistrement, telle que la <code>whitelist</code> des champs. Si un champ ne passe pas la validation, il sera automatiquement supprimé du tableau de données et ne sera pas copié dans l'entity. cela évite d'avoir des données incohérentes dans l'objet entity.</p>
<p><b>Sauvegarde d'Entity (save)</b></p> <p>Enchaînement des événements lors de la sauvegarde d'une entity</p>	<p>Quand une entity est sauvegardée, voici ce qui se passe:</p> <ol style="list-style-type: none"> <li>1. La vérification des règles commencera si elle n'est pas désactivée.</li> <li>2. La vérification des règles va déclencher l'<b>event Model.beforeRules</b>. Si l'event est stoppé, l'opération de sauvegarde va connaître un échec et retourner false.</li> <li>3. Les règles seront vérifiées. Si l'entity est en train d'être créée, les règles create seront utilisées. Si l'entity est en train d'être mise à jour, les règles update seront utilisées.</li> <li>4. L'<b>event Model.afterRules</b> sera déclenché.</li> <li>5. L'<b>event Model.beforeSave</b> est dispatché. S'il est stoppé, la sauvegarde sera annulée, et <code>save()</code> va retourner false.</li> <li>6. Les <b>associations parentes</b> sont sauvegardées. Par exemple, toute association <b>belongsTo</b> listée sera sauvegardée.</li> <li>7. Les <b>champs modifiés</b> sur l'entity seront sauvegardés.</li> <li>8. Les <b>associations Enfant</b> sont sauvegardées. Par exemple, toute association <b>hasMany</b>, <b>hasOne</b>, ou <b>belongsToMany</b> listée sera sauvegardée.</li> <li>9. L'<b>event Model.afterSave</b> sera dispatché.</li> <li>10. L'<b>event Model.afterSaveCommit</b> sera dispatché.</li> </ol> <p>Le diagramme suivant illustre le procédé ci-dessus:</p>



Saving associations :

- **BelongsTo**

```
// nom de l'association au singulier et en underscore
$data = [
    'title' => 'First Post',
    'user' => [
        'id' => 1,
        'username' => 'mark'
    ]
];
$e = $t->newEntity($data, [
    'associated' => ['Users']
]);
$t->save($article);
```

- **HasOne**

```
// nom de l'association au singulier et en underscore
$data = [
    'id' => 1,
    'username' => 'cakephp',
    'profile' => [
        'twitter' => '@cakephp'
    ]
];
$e = $t->newEntity($data, [
    'associated' => ['Profiles']
]);
```

- **HasMany**

```
// nom de l'association au pluriel et en underscore
$data = [
    'title' => 'First Post',
    'comments' => [
        ['body' => 'Best post ever'],
        ['body' => 'I really like this.']
    ]
];
```

<p>- <b>BelongsToMany</b></p>	<pre> ] ];  \$e = \$t-&gt;newEntity(\$data, [     'associated' =&gt; ['Comments'] ]); // Sans l'appel à dirty(), les commentaires mis à jour ne seront pas sauvegardés. \$e-&gt;comments[] = \$comment; \$e-&gt;setDirty('comments', true); Dans les cas où l'enregistrement a déjà des enregistrements associés dans la base de données, vous avez le choix entre deux stratégies de sauvegarde:     - <b>append (default)</b> : Les enregistrements associés sont mis à jour dans la base de données ou, si ils ne correspondent à aucun enregistrement existant, sont insérés.     - <b>replace</b> : Tout enregistrement existant qui ne correspond pas aux enregistrements fournis sera supprimé de la base de données. Seuls les enregistrements fournis resteront (ou seront insérés). Voir <a href="#">Associations HasMany</a> pour plus de détails sur la définition de <b>saveStrategy</b>.  // nom de l'association au <b>pluriel</b> et <u>en underscore</u> \$data = [     'title' =&gt; 'First Post',     'tags' =&gt; [         ['tag' =&gt; 'CakePHP'],         ['tag' =&gt; 'Framework']     ] ]; \$e = \$t-&gt;newEntity(\$data, [     'associated' =&gt; ['Tags'] ]); // Sans l'appel à dirty(), les tags mis à jour ne seront pas sauvegardés. \$e-&gt;tags[] = \$tag; \$e-&gt;setDirty('tags', true); Dans les cas où l'enregistrement a déjà des enregistrements associés dans la base de données, vous avez le choix entre deux stratégies de sauvegarde:     - <b>append</b> : Seuls les nouveaux liens seront créés de chaque côté de cette association. Cette stratégie ne va pas détruire les liens existants même s'ils ne sont pas présents dans le tableau d'entités à </pre>
-------------------------------	--

*sauvegarder*

- **replace (default)** : Lors de la sauvegarde, les liens existants seront retirés et les nouveaux liens seront créés dans la table de jointure. S'il y a des liens existants dans la base de données vers certaines entités que l'on souhaite sauvegarder, ces liens seront mis à jour, non supprimés et re-sauvegardés

Voir [Associations BelongsToMany](#) pour plus de détails sur la définition de **saveStrategy**.

**Créer une association entre deux entités existantes**, par exemple un utilisateur co-auteur d'un article :

```
$article = $this->Articles->get($articleId);  
$user = $this->Users->get($userId);  
$this->Articles->Users->link($article, [$user]);
```

**Sauvegarder les données supplémentaires de la table jointe (M-N)** :

Nous pourrions avoir une **table de jointure** qui ressemble à ceci:

```
id | student_id | course_id | days_attended | grade
```

```
$student->courses[0]->_joinData->grade = 80.12;  
$student->courses[0]->_joinData->days_attended = 30;  
$studentsTable->save($student);
```

Pour sauvegarder des données de tables jointes depuis données requêtées, données POST doivent être :

```
$data = [  
  'first_name' => 'Sally',  
  'last_name' => 'Parker',  
  'courses' => [  
    [  
      'id' => 10,  
      '_joinData' => [  
        'grade' => 80.12,  
        'days_attended' => 30  
      ]  
    ]  
  ],  
  // d'autres cours (courses)  
]  
];  
$student = $this->Students->newEntity($data, [  
  'associated' => ['Courses._joinData']  
]);
```

### findOrCreate()

```
$record = $table->findOrCreate(  
    ['email' => 'bobbi@example.com'],  
    function ($entity) use ($otherData) {  
        // Only called when a new record is created.  
        $entity->name = $otherData['name'];  
    }  
);
```

### Associations

Les quatre types d'association sont: **hasOne**, **hasMany**, **belongsTo**, et **belongsToMany**.

Relation	Type d'Association	Exemple
one to one	<b>hasOne</b>	Un user a un profile.
one to many	<b>hasMany</b>	Un user peut avoir plusieurs articles.
many to one	<b>belongsTo</b>	Plusieurs articles appartiennent à un user.
many to many	<b>belongsToMany</b>	Les Tags appartiennent aux articles.

Les Associations sont définies durant la méthode **initialize()** de votre objet table.  
Par exemple, si nous souhaitons définir une association **belongsTo** dans notre ArticlesTable:

```
namespace App\Model\Table;  
use Cake\ORM\Table;  
class ArticlesTable extends Table  
{  
    public function initialize(array $config)  
    {  
        $this->belongsTo('Authors');  
    }  
}
```



## 13.4. Erreurs sur le framework (TODO)

- **Deprecated Error:** strpos(): Non-string needles will be interpreted as strings in the future.

Use an explicit chr() call to preserve the current behavior in **vendor/cakephp/cakephp/src/Routing/RequestActionTrait.php, line 119**

**Solution:** remplacer

```
//if (is_string($url) && strpos($url, $baseUrl) === 0) {  
par  
if (is_string($url) && strpos($url, chr($baseUrl)) === 0) {
```

- **Correction d'une petite erreur sur MessageTrait.php, ligne 126**, qui provoque un warning à l'intérieur des pdf (doc de sortie notamment), ce qui empêche de l'ouvrir avec Acrobat Reader.

Ouvre le fichier `vendor/zendframework/zend-diactoros/src/MessageTrait.php`

et changer la ligne 126 en ceci :

```
//return isset($this->headerNames[strtolower($header)]);  
return isset($this->headerNames[strtolower($header[0]]);
```

## 13.5. Mise à jour du framework CakePhp

### 13.5.1. Upgrade de la version 3.6 à la version 3.7 (16/1/19 pm)

cf <https://book.cakephp.org/3.next/en/appendices/3-7-migration-guide.html>

**Version courante:**

\$ bin/cake version

=> 3.6

**Upgrade:**

\$ rm composer.lock (c'est mieux)

\$ php composer.phar **require** --update-with-dependencies "cakephp/cakephp:3.7.\*"

```
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Removing zendframework/zend-diactoros (1.7.0)
- Installing zendframework/zend-diactoros (1.8.6)
  Loading from cache

- Installing psr/simple-cache (1.0.1)
  Loading from cache

- Removing psr/log (1.0.2)
- Installing psr/log (1.1.0)
  Loading from cache

- Removing cakephp/chronos (1.1.4)
- Installing cakephp/chronos (1.2.3)
  Loading from cache

- Removing cakephp/cakephp (3.5.11)
- Installing cakephp/cakephp (3.7.2)
  Downloading: 100%

Writing lock file
Generating autoload files
> Cake\Composer\Installer\PluginInstaller::postAutoloadDump
```

### **Nouvelle version:**

**\$ bin/cake version**

**=> 3.7.2**

(avec plein de "Deprecated Error")

ex:

```
Deprecated Error: EventDispatcherTrait::eventManager() is deprecated. Use
EventDispatcherTrait::setEventManager()/getEventManager() instead. - vendor/cakephp/debug_kit/src/Panel/PanelRegistry.php, line: 35
You can disable deprecation warnings by setting `Error.errorLevel` to `E_ALL & ~E_USER_DEPRECATED` in your
config/app.php. [vendor/cakephp/cakephp/src/Core/functions.php, line 311]
```

Mais pb sur le site web : "Property \_transportConfig does not exist"

Cette page me donne la solution : <https://github.com/cakephp/cakephp/issues/12797>

⇒ *You have to update debug kit also, not documented yet.*

"cakephp/debug\_kit": "^3.17.0",

Je mets donc à jour la ligne debug\_kit dans mon composer.json et je réinstalle les dépendances de composer :

\$ rm composer.lock (c'est mieux)

\$ **php composer.phar install**

Loading composer repositories with package information

Updating dependencies (including require-dev)

- Removing cakephp/debug\_kit (3.16.7)

- **Installing cakephp/debug\_kit (3.17.0)**

Loading from cache

- Removing cakephp/bake (1.8.7)

- **Installing cakephp/bake (1.9.2)**

Downloading: 100%

**Writing lock file**

Generating autoload files

> Cake\Composer\Installer\PluginInstaller::postAutoloadDump

**Et là, c'est nickel !**

Bon, il y a encore énormément de "deprecated" errors dans vendor/ (et cela aussi avec cakephp 3.6)

### **13.5.2. Upgrade de la version 3.5 à la version 3.6 (16/1/19 matin)**

<https://book.cakephp.org/3.next/en/appendices/3-6-migration-guide.html>

et

<https://www.dereuromark.de/2018/03/14/cakephp-3-6-is-coming/#migration-tooling>

**Version courante:**

\$ **bin/cake version**

=> 3.5.11

### Upgrade:

```
$ php composer.phar require --update-with-dependencies "cakephp/cakephp:3.6.*"
```

```
./composer.json has been updated
```

```
Loading composer repositories with package information
```

```
Updating dependencies (including require-dev)
```

- Removing psr/simple-cache (1.0.1)
- Removing cakephp/cakephp (3.7.2)
- Installing cakephp/cakephp (3.6.14)

```
Downloading: 100%
```

```
Writing lock file
```

```
Generating autoload files
```

```
> Cake\Composer\Installer\PluginInstaller::postAutoloadDump
```

### Nouvelle version:

```
$ bin/cake version
```

**3.6.14**

⇒ Mais, il y a plein de “deprecated” errors !!!

### Remettre au propre les dépendances de Composer :

```
$ rm composer.lock
```

```
$ php composer.phar install
```

⇒ ce qui crée un nouveau composer.lock plus à jour

(notamment, phpunit est passé de la version 5 à la version 6)

### Check cakephp version à nouveau:

=> il n’y a plus que 1 erreur “deprecated” contre 5 avant !!!

```
$ bin/cake version
```

**3.6.14**

Deprecated Error: Use Cake\Http\ServerRequest instead of Cake\Network\Request. - vendor/composer/ClassLoader.php, line: 412  
You can disable deprecation warnings by setting `Error.errorLevel` to `E\_ALL & ~E\_USER\_DEPRECATED` in your config/app.php

**Bugfix quelques deprecated errors avec l'outil rector :**

<https://www.dereuromark.de/2018/03/14/cakephp-3-6-is-coming/>

⇒ mais pb de dépendance pour installer rector, donc je laisse tomber

## 13.6. PLUGINS

### 13.6.1. “Composer”, le gestionnaire de plugins

Pour pouvoir gérer les plugins, il faut d'abord installer **composer**.

Cet exécutable est **déjà installé dans LabInvent**, mais voici comment le ré-installer si besoin :

Depuis la racine du projet, exécuter les 4 lignes de php qui sont données au début de cette page web :

<https://getcomposer.org/download/>

Cela crée un fichier composer.phar qui va nous permettre de gérer les plugins.

Connaitre la version actuellement utilisée dans le projet :

```
$ php composer.phar --version
```

### 13.6.2. Plugins spécifiques à Cakephp

<https://book.cakephp.org/3.0/fr/plugins.html>

On peut facilement étendre les capacités de CakePhp à l'aide de plugins (à tester).

**Les meilleurs plugins** : <https://github.com/FriendsOfCake/awesome-cakephp>

- plugin PDF: <https://github.com/FriendsOfCake/awesome-cakephp#pdf>
- search : <https://github.com/FriendsOfCake/awesome-cakephp#search>

Tous les plugins officiels : <https://plugins.cakephp.org/>

### 13.6.3. Les plugins installés pour LabInvent

*(updated 13/5/19 - EP)*

*TODO: cette liste n'est pas exhaustive, il faut la compléter*

#### 13.6.3.1. Génération des QrCodes (plugin phpqrcode)

**TODO: Attention, ce plugin est obsolète, il faudrait le remplacer dès que possible**

“Package aferrandini/phpqrcode is abandoned, you should avoid using it. Use **android/qr-code** instead”

#### 13.6.3.2. Génération des documents PDF

Historiquement on a utilisé le plugin fpdf (depuis cakphp v2), puis en mai 2019 on a installé le plugin cakephp-dompdf pour le CRAL (car fpdf ne marchait pas bien sur cakephp v3 chez eux en php7).

Autres solutions possibles (mais pas testées) :

- Avec le plugin Cakepdf :
  - <https://github.com/FriendsOfCake/CakePdf>
  - <https://pritomkumar.blogspot.com/2017/04/generate-pdf-in-cakephp-3x-with-cakepdf.html>
- Sans plugin:
  - <http://caketuts.key-conseil.fr/index.php/2015/05/18/generer-du-pdf-sous-cakephp-v3/>

##### 13.6.3.2.1. Le Plugin fpdf

Voir la faq de fpdf: <http://www.fpdf.org/fr/FAQ.php>

### 13.6.3.2.2. Le plugin cakephp-dompdf pour générer des docs pdf

(installé le 19/5/2019)

(Test démo : <http://labinvent.test/demo/view/test.pdf>)

cf <http://www.flavienbeninca.fr/2016/05/04/generer-des-pdf-avec-dompdf-et-cakephp.html>

(voir aussi <https://github.com/DaoAndCo/cakephp-dompdf>)

Installation en 2 instructions :

```
$ php composer.phar require daoandco/cakephp-dompdf
```

```
$ bin/cake plugin assets symlink
```

(cette dernière instruction est sans doute à faire après installation de tout plugin)

Cela a créé 2 liens à la racine du dossier webroot/ :

- debug\_kit@ -> vendor/cakephp/debug\_kit/webroot
- dompdf@ -> vendor/daoandco/cakephp-dompdf/webroot

Ensuite, il faut faire quelques changements dans les fichiers config/bootstrap.php et config/routes.php

... etc (voir lien ci-dessus)

### 13.6.4. Mettre à jour ou Réinstaller tous les plugins (dossier vendor/)

(Réinstallation faite le 15/4/19 avec php 7)

**D'abord mettre à jour le fichier /composer.phar :**

```
$ php composer.phar self-update
```

**Ensuite, mettre à jour ou réinstaller tous les plugins:**



- Pour une simple **mise à jour** :  
\$ php composer.phar **update**
- Pour une **réinstallation complète** :  
(Pour être bien sûr, on peut carrément supprimer le dossier vendor/, il sera re-créé à la réinstallation) : \$ rm -rf vendor/  
\$ php composer.phar **install**

**Warnings après installation:**

- Package aferrandini/**phpqrcode** is abandoned, you should avoid using it. Use endroid/qr-code instead.
- Package phpunit/**phpunit-mock-objects** is abandoned, you should avoid using it. No replacement was suggested

**Attention, dans le cas d'une réinstallation, les dossiers tmp/cache/views, tmp/sessions, et tmp/tests sont re-crés et leurs permissions mises à jour :**

*Loading composer repositories with package information*

*Installing dependencies (including require-dev) from lock file*

*Nothing to install or update*

**Generating autoload files**

> Cake\Composer\Installer\PluginInstaller::postAutoloadDump

> App\Console\Installer::postInstall

**Created `Users/epallier/Documents/\_W\_more/PROJECTS/LABINVENT/SOURCE/labinvent2/tmp/cache/views` directory**

**Created `Users/epallier/Documents/\_W\_more/PROJECTS/LABINVENT/SOURCE/labinvent2/tmp/sessions` directory**

**Created `Users/epallier/Documents/\_W\_more/PROJECTS/LABINVENT/SOURCE/labinvent2/tmp/tests` directory**

**Set Folder Permissions ? (Default to Y) [Y,n]? Y**

*Permissions set on /Users/epallier/Documents/\_W\_more/PROJECTS/LABINVENT/SOURCE/labinvent2/tmp/cache/views*

*Permissions set on /Users/epallier/Documents/\_W\_more/PROJECTS/LABINVENT/SOURCE/labinvent2/tmp/sessions*

*Permissions set on /Users/epallier/Documents/\_W\_more/PROJECTS/LABINVENT/SOURCE/labinvent2/tmp/tests*

*No Security.salt placeholder to replace.*

### 13.6.5. Ajouter un plugin spécifique

<https://book.cakephp.org/3.0/fr/plugins.html#installer-un-plugin-avec-composer>

Exemple: on veut ajouter le plugin **fpdf de setasign** (pour générer des fichiers pdf)

Ne le faites pas, car c'est déjà fait pour LabInvent. C'est juste pour expliquer comment faire.

## Installation automatique (méthode conseillée) :

Il suffit de taper cette commande :

```
$ php composer.phar require setasign/fpdf
```

Ceci installe la dernière version de fpdf, et met à jour vos fichiers **composer.json**, **composer.lock**, met à jour **vendor/cakephp-plugins.php** et met à jour votre autoloader.

## Installation manuelle :

Il suffit d'ajouter le nom du plugin dans le fichier composer.json dans la clé "require", à la fin, comme ceci :

```
"require" : {  
    "php" : ">=5.6",  
    "cakephp/cakephp" : "3.7.*",  
    "cakephp/migrations" : "^2.0.0",  
    "cakephp/plugin-installer" : "^1.0",  
    "mobiledetect/mobiledetectlib" : "2.*",  
    "aferrandini/phpqrcode" : "1.0.1",  
    "setasign/fpdf" : "1.8.1"  
},
```

Ensuite, il faut relancer l'installation des plugins (ils sont tous installés dans le dossier **vendor/**), comme ceci :

```
$ php composer.phar update
```

Si jamais il y a un conflit avec d'autres plugins, on peut essayer de supprimer le fichier composer.lock afin de laisser composer libre de mettre à jour certains plugins. Dans ce cas, attention toutefois à relancer les tests (anti-régression) ensuite afin de s'assurer que tout fonctionne comme avant.

```
$ mv composer.lock composer.lock.ORIG
```

```
$ php composer.phar install
```

On devrait maintenant voir un sous-dossier **setasign/fpdf/** dans le dossier **vendor/**

De plus, un nouveau fichier **composer.lock** doit avoir été créé.

Si ça ne marche toujours pas, on peut aussi essayer de ré-installer tous les plugins. Pour cela, il suffit de taper "install" au lieu de "update". Cela re-crèera entièrement le dossier vendor/. Pour en être bien sûr on peut même supprimer complètement ce dossier (rm -rf vendor/) :

```
$ php composer.phar install
```

### 13.6.6. Supprimer un plugin

\$ php composer.phar **remove** vendor/package

ex: php composer.phar **remove setasign/fpdf**

Cela va supprimer le package (plugin) du dossier vendor/, et des fichiers composer.json et composer.lock...

(Il faudra aussi supprimer toute référence à ce plugin dans LabInvent).

# 14. Historique du logiciel - Changements importants

---

## Année 2020 - Changements importants

---

### 11/03/2020 Version 3.7.9.0 (EP)

NOUVELLE VERSION RESPONSIVE validée et installée officiellement à l'IRAP et au CRAL (mobile ready)  
Nouveau format de version qui colle à la version utilisée du framework Cakephp (3.7.9)

### 10/03/2020 Version 2.14.7 (EP)

Nombreux petits bugfixes (dont 1 gros : bouton CANCEL) :

- bugfix materiels/add : bouton CANCEL faisait un SUBMIT (comme pour /edit) !!!
- bugfix materiels/view : bugfix css responsive.css pour apparence et position boutons
- bugfix doc attaché :
  - commentaire si doc attaché avec mauvais nom
  - emplacement photo matériel (à gauche ou au-dessus et non pas à droite du QrCode, moche)
  - lien téléchargement documents attachés
- bugfix pb accent sur liste materiels associés à une catégorie
- étiquette posée => vert (rouge sinon)
- apropos updated
- lien vers page etiqueteuse updated
- ...

### 06/03/2020 Version 2.14.6 (EP)

6 gros bugfixes de la version 2.14 (responsive) sur 1 semaine de boulot !!

### 28/02/2020 Version 2.14.0 (Jeanne Prugniel, stagiaire CRAL)

Application maintenant "responsive" avec l'utilisation de bootstrap (ok sur pc, tablette, et mobile)

DATE VERSION	CHANGEMENTS VISIBLES	CHANGEMENTS NON VISIBLE (STRUCTURE INTERNE)
15/5/19 Version ???	Bugfix, simplification, et amélioration de la fonctionnalité "Copier ce matériel"	
15/04/19 Version 2.12.4 (EP)		<p>Mise à jour du fichier <b>/composer.phar</b> (php composer.phar self-update, avec php version 7)</p> <p><b>Warnings après installation:</b></p> <ul style="list-style-type: none"> <li>- Package aferrandini/phpqrcode is abandoned, you should avoid using it. Use endroid/qr-code instead.</li> <li>- Package phpunit/phpunit-mock-objects is abandoned, you should avoid using it. No replacement was suggested</li> </ul> <p>— Installation du plugin <b>tcpdf</b> pour remplacer <b>fpdf</b> :</p> <pre>\$ php composer.phar require tecnickcom/tcpdf Using version ^6.2 for tecnickcom/tcpdf ./composer.json has been updated Loading composer repositories with package information Updating dependencies (including require-dev) Package operations: 1 install, 0 updates, 0 removals — Installing tecnickcom/tcpdf (6.2.26): Downloading (100%)</pre>
02/04/19 version 2.12.3 (EP)	Recherche matériels sur numéro interne labo en plein texte Affichage du prix dans le résultat de recherche	

<p><b>26/03/19</b>  <b>version 2.12</b>  <b>(EP)</b></p>		<p>Bugfix Gestionnaire de reference : maintenant bien sauvegardé (avant il était perdu...) =&gt; modif de la BD</p> <p>Gros cleanup du code LDAP dans LdapConnectionsTable.php</p> <p>Refactorisation des actions add() et edit() de MaterielsController en une seule fonction add_or_edit() car elles étaient très semblables</p>
<p><b>14/03/19</b>  <b>version 2.11.00</b>  <b>(EP)</b></p>		<p>Format étiquette configurable (pour que CRAL puisse faire sa version), et QrCode peut être imprimé dessus</p> <p>Doc du logiciel grandement mise à jour</p>
<p><b>26/02/19</b>  <b>version 2.10.21</b>  <b>(EP)</b></p>		<p>Simplification des interactions domaine&lt;-&gt;categorie (materiel) : la selection de la categorie ne met plus à jour le domaine, ça ne sert à rien</p> <p>Bugfix "Categorie sélectionnée ne met plus à jour le domaine"</p>
<p><b>20/02/19</b>  <b>version 2.10.15-19</b>  <b>(EP)</b></p>		<p>Retour de la nouvelle version du <b>LDAP avec 2 modes (anonyme pour IRAP, et authentifié pour le CRAL)</b>, refactorisé</p> <p>Ajout d'un lien "<b>What's New</b>" en bas de page web, qui pointe vers cette page "Historique du logiciel"</p> <p>LDAP refactorisation &amp;&amp; optimisation : LdapConnectionsTable.php/getAllLdapUsers() remplacé par getAllLdapUsersNEW</p> <p>LDAP refactor progressif pour remettre le mode LDAP authentifié (pour CRAL)</p>
<p><b>15-20/02/19</b>  <b>version 2.10.7-14</b>  <b>(EP)</b></p>		<p><b>(IRAP only) Gros bugfix du LDAP</b> : en mode LDAP anonyme, n'importe qui pouvait se connecter avec n'importe quel compte (il n'y avait pas de vérification du mot de passe !!!)</p>

<p><b>23/01/19</b>  <b>version</b>  <b>2.10.1-6 (EP)</b></p>	<p><b>Recherche Matériels, Bugfixes et améliorations :</b></p> <ul style="list-style-type: none"> <li>- Bugfix Recherche générale sur plusieurs mots : fait un AND de ces mots et non pas un OR !!!</li> <li>- Bugfix Recherche générale d'une date</li> <li>- <b>Ajout recherche générale d'un fournisseur et d'une catégorie</b> <ul style="list-style-type: none"> <li>- Bugfix recherche sur un montant à virgule...</li> <li>- Complète refactorisation de MaterielsController::find(), y avait besoin ! <ul style="list-style-type: none"> <li>- Ajout de tests</li> <li>- ...</li> </ul> </li> </ul> </li> </ul>	<p><b>Mise à jour du framework cakephp</b> à la version courante :</p> <ul style="list-style-type: none"> <li>- passage de v3.5 à v3.7</li> <li>- adaptations pour cakephp 3.7</li> <li>- ajout du dossier vendor/ dans git</li> </ul> <p>(désormais versionné pour faciliter la mise à jour du framework cakephp)</p> <p><b>Nouveaux fichiers</b> VERSION, UPDATE.sh, et mise à jour de TESTS.sh</p> <p><b>LDAP authentifié bugfix</b></p>
<p><b>15/01/2019</b>  <b>version 2.9.2</b>  <b>(EP)</b></p>		<p><b>Nouveau script UPDATE</b> ("install/update.sh") pour mettre à jour automatiquement le logiciel (git pull) ET la BD (si besoin)</p>
<p><b>10/01/19</b>  <b>version</b>  <b>2.9.1.11 (EP)</b></p>		<p><b>Nouveau mode LDAP AUTHENTIFIÉ</b> opérationnel (pour le CRAL):</p> <ul style="list-style-type: none"> <li>- renommé tous les champs et variables *_ldap en ldap_* pour meilleure lisibilité</li> <li>- adapté la section LDAP de la config</li> <li>- update fichier database/labinvent_last_version.sql contenant la BD COMPLETE</li> <li>- cleanup</li> </ul>

---

## Année 2018

---

### 21/12/2018 - version: 2.9.1.4 (EP)

Version et date affichés automatiquement et merge vues listes... :

- Version et date affichés automatiquement à partir du README.md
- Les vues "Voir les autres listes" et "Outils/Gérer le contenu variable..." sont désormais les mêmes

### 14/12/2018 - version: 2.9.1.0 (EP)

Ajout en cours du mode LDAP authentifié & Nouvelle documentation (dossier doc/)

### ETE 2018 - version: 2.9.0 (Malik Imelhaine, stagiaire)

Nombreux petits bugfixes et petites améliorations (a priori sans changement structure BD)

### 02/02/2018 - version: 2.8.1 (EP)

Nombreux bugfixes (saisie dates, etiquette, doc admission...)

---

## Année 2017

---

### 15/09/2017 Version: 2.8.0 (EP)

- Nouveau système ACL simplifié basé sur les tableaux \$easyACL (en cours)
- Tests paramétrés
- Bugfix emails

### 30/08/2017 Version: 2.7.9 (EP)

- fonction intelligente ApplicationController::getUserRole() qui donne le role "Utilisateur" par défaut pour un utilisateur non privilégié
- Refactorisation des ACL (authorizations) dans isAuthorized() et beforeFilter()

### 28/08/2017 Version: 2.7.6 (EP)



- renforcement important des TESTS : généralisation, refactorisation, simplification + numérotation systématique (cf doc ACL)

- nouvelle classe General dont héritent tous les tests

- nouvelle philo mise en place : 1 fichier tests par Controleur (c'était déjà le cas), puis pour un controleur donné, tri des tests par

ACTION, puis pour chaque action, tests systématique de tous les ROLES (profils) avec les cas particuliers de chacun

06/07/2017 Version: 2.7.3 (EP)

- suppression du contenu de vendor/ => désormais ignoré car autogénéré

- update des plugins phpqrcode et fpdf => désormais via composer.json

- amélioration script install => pour générer automatiquement le contenu de vendor/

30/06/2017 Version: 2.7.0 (EP)

Passage de Cakephp3.2 à 3.4, Phpunit 5 à 6, et compatibilité avec Php5.6 et 7

---

12/05/2017 Version: 2.6.2 (Thibaud Ajas?)

L'ajout d'un gestionnaire du matériel est désormais obligatoire à l'ajout d'un matériel

09/05/2017 Version: 2.6.0 (Thibaud Ajas?)

Les noms des matériels sont en rouge dans la liste lorsque la date de garantie à été dépassée, ainsi que la date elle-même dans leur fiche détaillée

Petits bugfixes divers

---

25/04/2017 Version: 2.5.6 (Thibaud Ajas?)

On peut maintenant configurer et choisir sur la page le nombre de materiels affichés (20 par défaut)

Petits bugfixes divers

Amélioration du script et de la doc d'installation

---

## Année 2016

---

24/06/2016 Version: **2.4.7.2** (Version 2.4 finale)  
Implémentation des ACL (droits) & Autres demandes  
<https://projects.irap.omp.eu/versions/107>

---

30/05/2016 Version: **2.3.2.1** (Version 2.3 finale)  
Implémentation du LDAP (vrai et fake)  
<https://projects.irap.omp.eu/versions/108>

---

23/05/2016 Version: **2.2.5.4** (Version 2.2 finale)  
Implémentation de toutes les autres actions  
<https://projects.irap.omp.eu/versions/106>

---

12/05/2016 Version: **2.1.10** (Version 2.1 finale)  
Implémentation complète du CRUD  
<https://projects.irap.omp.eu/versions/101>

---

04/05/16 Version **2.0.8** (Version 2.0 finale)  
Version de base (from bake) : php5 + cakephp3  
<https://projects.irap.omp.eu/versions/105>

---

21/01/16 Version 1.3.636 (Version 1.3 finale)  
php5 + cakephp2.1  
<https://projects.irap.omp.eu/versions/17>