Monwoo Web Starter Symfony PDF Billings (Free)

 Build by Miguel Monwoo, **Open Source Apache-2.0 with Copyright © MONWOO 2017-2024**

www.monwoo.com/don

Aim

Provide a PHP local tool for pre-fillable self signed PDF billing templates. Bonus : Basic controller ok for simple JWT authentification if needed.

Demonstration

@demo mws.monwoo.com/demos/sf-pdf-billings/

Build

```
# tested under php 8.1.2
# Other php versions might work, but it's not tested yet.
php -v

wget https://getcomposer.org/composer.phar
alias composer="php '$PWD/composer.phar'"
cd apps/mws-sf-pdf-billings/backend

mkdir config/jwt
openssl genrsa -out config/jwt/private.pem -aes256 4096 # pass : jwt_test
openssl rsa -pubout -in config/jwt/private.pem -out config/jwt/public.pem
# pass : jwt_test

rm .env.local.php
cp .env.dev.dist .env # Env for Symfony AND Svelte frontend

export APP_ENV=dev
# Same for packages first :
(cd ../../../packages/mws-moon-manager && composer install)
composer install
php bin/console assets:install --symlink public
php bin/console fos:js-routing:dump

# generate logs BEFORE frontend build since embeded by
# packages/mws-moon-manager-
ux/components/layout/widgets/GitLogsChart.svelte
git log --pretty=$'x\tx\t%ai' --numstat \
-i --grep='\[mws-pdf-billings\]' \
--grep='\[mws-moon-manager\]' \
--branches --tags --remotes --full-history \
--date-order --date=iso-local> git-logs.tsv
```

```
pnpm install
pnpm run build

php bin/console doctrine:migrations:migrate -n
# Generate one user to be able to login : (-c 1 will remove existing
users)
# php bin/console mws:add-user -c 1
php bin/console mws:add-user

# Use SYMFONY dev server (not same as php builtin or bin/console)
wget https://get.symfony.com/cli/installer -O - | bash
alias symfony="~/.symfony5/bin/symfony"
symfony server:start

# if you dev css/js side too, for watch mode :
pnpm run watch

# Same for packages first :
(cd ../../../packages/mws-moon-manager && composer install)
# install with local PRIVATE bundles :
COMPOSER=composer.private.json composer install
# if already installed and did update package composer.json :
COMPOSER=composer.private.json composer update
# + You might need to restart symfony dev server, relaunch :
symfony server:start


# bootstrap database
php bin/console doctrine:migrations:migrate

# use php builtin
# (TODO : php router and missing .htaccess checks ok ?)

# All is ready for your PDF edition (TODO : dev in progres...)
# TODO : Just fill the form or pre-fill by json POST request...
open http://localhost:8000
```

Build for Production

```
# Clean all un-wanted files for production :
rm -rf **/node_modules

# re-install for production only :
pnpm install --prod

# TIPS : if you use optional features in packages.json :
rm -rf **/node_modules
pnpm install --prod --no-optional

# tested under php 8.1.2
# Other php versions might work, but it's not tested yet.
php -v
alias composer="php '$PWD/composer.phar'"
cd apps/mws-sf-pdf-billings/backend

# CLEAN DEV ENV (will lose your dev, be sure of it :)
rm -rf mws-sf-pdf-billings.zip var vendor config/jwt .env.local.php

# Clean possible dev configs (if no previous prod builds...)
# mkdir -p config-disabled/packages config-disabled/routes
# mv config/routes/web_profiler.yaml config-disabled/routes/
# mv config/packages/debug.yaml config/packages/web_profiler.yaml \
# config-disabled/packages/

export APP_ENV=prod

cp .env.prod.dist .env # Env for Symfony AND Svelte frontend
# echo 'APP_ENV=prod' > .env
# cp .env.prod.distXXX .env.prod # put your private extended PHP Symfony
info inside...

# Build for prodution
mkdir config/jwt
# WARNING : use hard pass other than : jwt_test (and setup accordingly in
.env.prod)
openssl genrsa -out config/jwt/private.pem -aes256 4096
openssl rsa -pubout -in config/jwt/private.pem -out config/jwt/public.pem

# Same for packages first :
(cd ../../../packages/mws-moon-manager && composer install \
 --no-ansi --no-dev \
--no-interaction --no-scripts --no-progress \
--optimize-autoloader)

# APP_ENV=prod composer install --no-dev
APP_ENV=prod composer install --no-ansi --no-dev \
--no-interaction --no-scripts --no-progress \
--optimize-autoloader
```

```
# generate .env.local.php
APP_ENV=prod composer dump-env prod

# refresh assets :
# php bin/console --env=prod assets:install
php bin/console --env=prod assets:install --symlink public
# php bin/console assets:install --symlink public
bin/console fos:js-routing:dump --format=json --target=assets/fos-
routes.json

# bootstrap database
rm var/data.db.sqlite # clean old one
php bin/console doctrine:migrations:migrate -n

# bootstrap one user ONLY to let it be change and do wiziwig updates :
php bin/console mws:add-user -c 1

cp var/data.db.sqlite var/data.gdpr-ok.db.sqlite

# rebuild assets for production :
pnpm run build

# clean un-wanted node module files, all is now build
# cd - # DO it at repo root since may not follow symlinks for delete
# rm -rf **/node_modules
# cd - # TIPS : using " -x '**/node_modules/**'" to avoid node module

APP_ENV=prod composer dump-env prod
rm -rf var/cache var/log

zip -r mws-sf-pdf-billings.zip \
.htaccess composer.json package.json config public src \
templates translations \
vendor var .env.local.php \
-x '**/node_modules/**'

# test local prod (if no base href configs ?)
APP_ENV=prod symfony server:start 2> /dev/null

# clean for local dev :
rm .env.local.php
```

Build production for debugs (for full pre-prod debugs)

```
# tested under php 8.1.2
# Other php versions might work, but it's not tested yet.
php -v
alias composer="php '$PWD/composer.phar'"
cd apps/mws-sf-pdf-billings/backend

# CLEAN DEV ENV (will lose your dev, be sure of it :)
rm -rf mws-sf-pdf-billings.zip var vendor config/jwt .env.local.php

# bring back dev configs (that MIGHT move due to production builds)
mv config-disabled/packages/* config/packages/

export APP_ENV=dev

# Build for prodution
mkdir config/jwt
# WARNING : use hard pass other than : jwt_test (and setup accordingly in
.env.prod)
openssl genrsa -out config/jwt/private.pem -aes256 4096
openssl rsa -pubout -in config/jwt/private.pem -out config/jwt/public.pem
# pass : jwt_test

# Same for packages first :
(cd ../../../packages/mws-moon-manager && composer install)

# APP_ENV=prod composer install --no-dev
composer install

# bootstrap database
php bin/console doctrine:migrations:migrate -n
cp var/data.db.sqlite var/data.gdpr-ok.db.sqlite

# bootstrap one user ONLY to let it be change and do wiziwig updates :
php bin/console mws:add-user -c 1

# build assets for dev :
pnpm run dev

zip -r mws-sf-pdf-billings.zip .env.dev \
.htaccess composer.json config public src \
templates translations \
vendor var .env
```

Launching Tests for debugs

```
alias composer="php -d memory_limit=2G '$PWD/composer.phar'"
cd apps/mws-sf-pdf-billings/backend

export APP_ENV=dev
# Same for packages first :
(cd ../../../packages/mws-moon-manager && composer install)
composer install

# Re-do manually if you change some routes path :
bin/console fos:js-routing:dump --format=json --target=assets/fos-
routes.json

# watch assets for code changes to work
# on full page reloads without caches
# (keep it running in new terminal) :
pnpm run watch

# TODO : doc for e2e tests
# alias symfony="~/.symfony5/bin/symfony"
alias symfony="~/.symfony6/bin/symfony"
symfony server:start
```

Going further

```
# TIPS : in DEV ONLY : NEED
export APP_ENV=dev
# Same for packages first :
(cd ../../../packages/mws-moon-manager && composer install)
composer install # in case of env change

# usefull :
php bin/console debug:form

# add users
php bin/console make:user
# list of available make commandes
php bin/console list make

# other tools for models :
php bin/console make:entity BillingConfig

rm src/Form/BillingConfigType.php
# might be needed to succed commands
# (still loading BillingConfigType from cached php files )
rm -rf var/cache
# # use below if you have error on missing BillingConfigType :
# mv src/Form/BillingConfigSubmitableType.php \
# src/Form/_BillingConfigSubmitableType.php
php bin/console make:form BillingConfigType BillingConfig
# mv src/Form/_BillingConfigSubmitableType.php \
# src/Form/BillingConfigSubmitableType.php

# https://symfony.com/doc/current/doctrine/associations.html
php bin/console make:entity Outlay
rm src/Form/OutlayType.php
php bin/console make:form OutlayType Outlay

php bin/console make:entity Transaction
rm src/Form/TransactionType.php
php bin/console make:form Transaction Transaction

php bin/console make:entity Product
rm src/Form/ProductType.php
php bin/console make:form Product Product

# ensure your databse is clean and in sync with existing migration
# (WARNING : will reset your dev database) :
# rm var/data.db.sqlite && php bin/console doctrine:migrations:migrate -n

# Below MIGHT be ok for little change :
php bin/console make:migration
php bin/console doctrine:migrations:migrate -n

php bin/console make:entity --regenerate
```

```
php bin/console make:entity --help
# Make migration from your database diff :
php bin/console doctrine:migrations:diff --help
# If you change your model,
# you need to generate the associated migrations :
php bin/console make:migration
rm var/data.db.sqlite
php bin/console doctrine:migrations:migrate -n
cp var/data.db.sqlite var/data.gdpr-ok.db.sqlite

# add a new controller
php bin/console make:controller PdfBillings

# You can test the JWT feature with curl :
# Add a user (you will have to build up all other security aspects)
curl -X POST -H "Content-Type: application/json" \
 -d '{"username": "test", "password": "123", "email": "test@test.fr"}' \
 http://127.0.0.1:8000/api/register

# Get the JWT token from custom 'login_check' api url
curl -X POST -H "Content-Type: application/json" \
 -d '{"username": "test", "password": "123"}' \
 http://127.0.0.1:8000/api/login_check

# Load some client (if exist) :
open http://localhost:8000/?
billing_config_submitable[clientSlug]=newClient

# Create empty quotation for new client slug if do not exist
# (WRONG CSRF, but will create client with empty value if don't exist)
curl -X POST -d "billing_config_submitable[clientSlug]=test2" \
http://127.0.0.1:8000/

# Create or update quotation for new client slug (WRONG CSRF)
curl -X POST \
-H "application/x-www-form-urlencoded" \
-d "billing_config_submitable[clientSlug]=test3" \
-d "billing_config_submitable[clientName]=ClientTestedName" \
http://127.0.0.1:8000/

# Create or update quotation for new client slug (WRONG CSRF)
curl -F billing_config_submitable[clientSlug]=test3 \
-F billing_config_submitable[clientName]=ClientTestedName \
http://127.0.0.1:8000/

# HANDELING Csrf with curl :
rm cookies.txt
InSrc=$(curl -c cookies.txt -b cookies.txt http://127.0.0.1:8000/ \
2>/dev/null | tr '\n' ' ')
TokenSep='billing_config_submitable[_token]" value="'
HalfPart=$(echo "${InSrc//$TokenSep/$\n}" | tail -n 1)
CSRF=$(echo "${HalfPart//\"/$\n}" | head -n 1)

curl -c cookies.txt -b cookies.txt -F
```

```
"billing_config_submitable[clientSlug]=test3" \
-F "billing_config_submitable[_token]=$CSRF" \
-F "billing_config_submitable[clientName]=ClientTestedName" \
http://127.0.0.1:8000/

# Similar (ok thanks to SF5 framework...)
curl -X POST -c cookies.txt -b cookies.txt \
-H "application/x-www-form-urlencoded" \
-d "billing_config_submitable[_token]=$CSRF" \
-d "billing_config_submitable[clientSlug]=test3" \
-d "billing_config_submitable[clientName]=ClientTestedName" \
http://127.0.0.1:8000/

# After some moment, you want to start back from fresh data.
# to clean and rebuild the database :
rm var/data.db.sqlite && php bin/console doctrine:migrations:migrate -n
# save the empty fresh database as GDPR safe :
cp var/data.db.sqlite var/data.gdpr-ok.db.sqlite

# Updating twig to bring extension if not already setup in composer.json
# twig/extensions soud depreciated and replaced by twig/extra-bundle
# as done with next command :
composer require twig
# For currency filter to work :
composer require twig/intl-extra
# https://symfony.com/doc/current/controller/error_pages.html
composer require symfony/twig-pack
# open error page in dev :
open http://localhost:8000/index.php/_error/404

# https://symfony.com/doc/current/reference/configuration/twig.html
php bin/console config:dump-reference twig
php bin/console debug:config twig

# MacOS quick tool to resize png pictures (and convert to jpeg) :
find . -name '*.png' | sed 's/\.png//g' \
| xargs -I % sips -Z 1800 \
--setProperty format jpeg \
--setProperty formatOptions 70.00 \
"%.png" --out "%.jpg"

# optimise your signature to avoid huge pdf outputs (Keep transparency)
sips -Z 400 \
"var/SignatureCleanV1.png" --out "var/businessSignature.png"

# optimise your signature to avoid huge pdf outputs (best compressions)
sips -Z 400 \
--setProperty format jpeg \
--setProperty formatOptions 90.00 \
"var/SignatureCleanV1.png" --out "var/businessSignature.jpg"

# Add some tests :
# php bin/console console make:functional-test
# Intalls for tests :
```

```
composer require --dev phpunit/phpunit symfony/test-pack
composer require symfony/panther --dev # required for e2e scenarios
# then use ChromeDriver :
# https://github.com/symfony/panther
# # Ubuntu :
# apt-get install chromium-chromedriver firefox-geckodriver
# # Mac Os : https://brew.sh/
# brew install chromedriver geckodriver
# # Windows : https://chocolatey.org/
# choco install chromedriver selenium-gecko-driver
# # Monwoo (Mac OS) :
# https://codeception.com/docs/modules/WebDriver#local-chrome-andor-
firefox
# https://codeception.com/docs/modules/WebDriver#usage
# https://chromedriver.storage.googleapis.com/index.html?
path=108.0.5359.71/
# In my case, for my chrome :
wget
https://chromedriver.storage.googleapis.com/112.0.5615.49/chromedriver_mac
64.zip
rm chromedriver
unzip chromedriver_mac64.zip
rm chromedriver_mac64.zip

# launch the chrome webdriver (in background) :
./chromedriver --url-base=/wd/hub &

composer require --dev dbrekelmans/bdi
vendor/bin/bdi detect drivers

# install test tools :
composer require --dev phpunit/phpunit symfony/test-pack
composer require --dev symfony/phpunit-bridge
php vendor/bin/simple-phpunit

# use CLI to add some tests :
php bin/console make:test

php vendor/bin/simple-phpunit tests/BillingConfigTest.php

# list recipes :
composer recipes

# Do some package AUTO-UPDATINGS :
rm -rf vendor composer.lock var/cache var/log
composer update

# Preview new translations from source codes by locale :
php bin/console translation:extract --dump-messages fr
# Update translations file from source codes by locale :
php bin/console translation:extract --format=yaml \
--as-tree=7 --force fr

# prefix, output format, domain, sorting, etc... :
```

```
php bin/console translation:extract --help

# Each time you create a new message catalog
# (or install a bundle that includes a translation catalog),
# be sure to clear your cache so that Symfony
# can discover the new translation resources:
php bin/console cache:clear

# tips : if you have some html to convert to yaml string, you can use :
open https://olayaml.com/yaml-stringifier
# MayBe : internal page and service for wysiwyg translations ?

# https://symfony.com/doc/current/translation.html
# Installing and Configuring a Third Party Provider :
composer require symfony/loco-translation-provider
# You'll now have a new line in your .env file that you can uncomment:
LOCO_DSN=loco://API_KEY@default
# Pushing Translations (OVERWRITE provider values)
php bin/console translation:push loco --force
# and Pulling (OVERWRITE local files values)
php bin/console translation:pull loco --force

# Forseen :
# https://symfonycasts.com/screencast/stimulus/controllers
php bin/console make:stimulus-controller

# We did use file structure changes to avoid not loaded bundles,
# might be possible with PHP code too :
# https://symfony.com/doc/current/bundles/extension.html

# https://symfony.com/doc/current/serializer.html
# https://okazy.github.io/symfony-docs/serializer.html
composer require symfony/serializer-pack

man md5
md5 -q /Users/miguel/Downloads/FactureMonwoo_________________.yaml
# Will give your the backup file md5 hash.
# That's the same as the url param dataMD5 encoded in the QrCode signature

# https://stackoverflow.com/questions/28700659/how-to-configure-tcpdf-
when-installing-with-composer
rm -rf var/cache
composer dump-autoload

# TIPS Axelo : use Vite Inspector
# https://github.com/sveltejs/vite-plugin-
svelte/blob/main/docs/config.md#inspector
# [23:36, 01/08/2023] Axelo: svelte.config
# [23:37, 01/08/2023] Axelo: "@sveltejs/vite-plugin-svelte": "^2.4.2"
# [23:37, 01/08/2023] Axelo: "vite": "^4.3.9",


# php bin/console fos:js-routing:dump
bin/console fos:js-routing:dump --format=json --target=assets/fos-
```

```
routes.json

# bootstrap one user ONLY to let it be change and do wiziwig updates :
php bin/console mws:add-user -c 1
```

Useful Links

Learn more about the power of pdf billings:

- Décret n° 2023-377 du 16 mai 2023 - PDF signing requirements for big FR buisiness
- Building an SPA (SF doc)
- QipsiusTCPDFBundle (SF bundle for PDF)
- PHP PDF Library
- Using simple TCPDF
- Using advanced TCPDF
- SF Form types
- Free PDF Document Importer
- Html table
- Symfony debug forms errors
- Style css borders
- Tuto for Terms of sales

Supports

- You can use regular features of : github.com/Monwoo/web-starters-free/issues

To support us and/or help us open more software, send a subvention with :

- www.monwoo.com/don