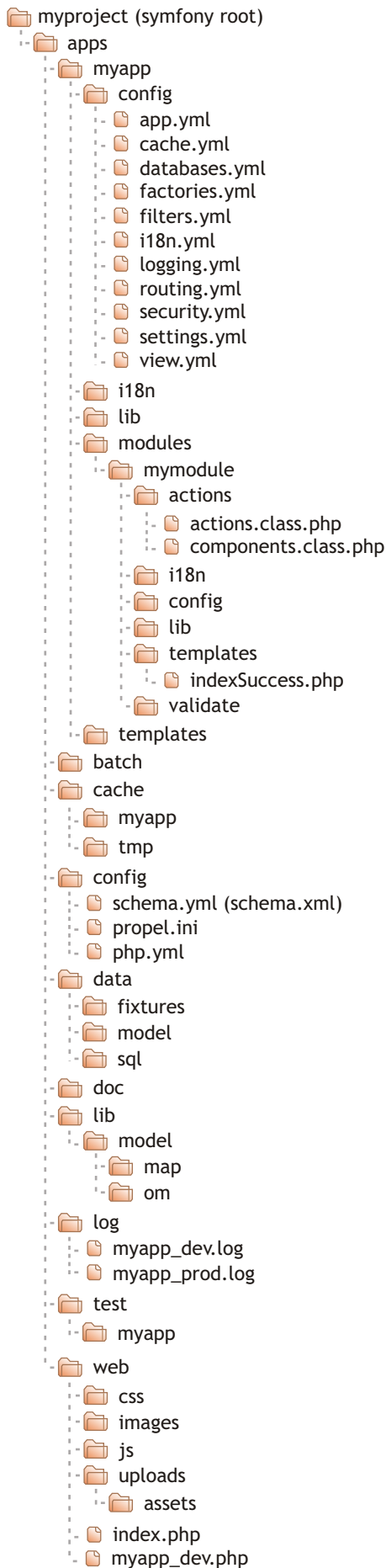


Structure des répertoires et CLI

STRUCTURE DES RÉPERTOIRES PAR DÉFAUT



INTERFACE EN LIGNE DE COMMANDE (CLI)

```

$ symfony -T
Liste Complète des commandes en ligne

$ symfony -V
Version du package symfony

$ symfony clear-cache <application_name> [template|config]
Vider le cache (raccourci: cc)

$ symfony init-project <project_name>
Initialise un projet et génère les fichiers et répertoires indispensables à son execution

$ symfony init-app <application_name>
Initialiser une application. Les fichiers contrôleurs sont aussi générés dans le répertoire web/
pour chaque environnement par défaut:
index.php (prod) et myapp_dev.php (dev)

$ symfony init-module <application_name> <module_name>
Initialise un module. Le module est prêt à être utilisé:
http://myapp.example.com/index.php/mymodule

$ symfony propel-build-schema [xml]
Génère le schema.yml de la représentation de la base existante. Pour la génération en xml
utiliser l'option xml

$ symfony propel-build-model
Génère les classes PHP pour le modèle selon les informations définies dans le fichier
schema.yml. Les classes de bases seront créées dans
myproject/lib/model/om/ directory:
BaseArticle.php      BaseComment.php
BaseArticlePeer.php  BaseCommentPeer.php
De plus, les fichiers à modifier seront créés dans myproject/lib/model:
Article.php          Comment.php
ArticlePeer.php      CommentPeer.php

$ symfony propel-generate-crud <application_name> <module_name> <ClassName>
Scaffolding - Génère un nouveau module CRUD (création, récupération, mise à jour et suppression)

$ symfony propel-build-sql
Génère le code SQL pour créer les tables décrites à partir de config/schema.yml, dans le
fichier myproject/data/sql/lib.model.schema.sql

$ symfony propel-build-db
Crée une base de données vierge

$ symfony propel-insert-sql
Insère le code SQL à partir de myproject/data/sql/lib.model.schema.sql dans la base de données

$ symfony sync <environment_name> [go]
Synchronise le projet sur une machine distante

$ symfony propel-init-admin <application_name> <module_name> <ClassName>
Initialise un module Propel d'administration basé sur une classe du modèle

$ symfony test <application_name>
Lance la suite de tests relative à l'application

$ symfony plugin-install [local|global] <channel_name>/<plugin_name>
Installe un nouveau plugin

$ symfony freeze
Convertir le projet en application autonome

$ symfony unfreeze
Effectue l'opération inverse. Efface les répertoires data/symfony/, lib/symfony/ et web/sf/

$ symfony disable <application_name> <environment_name>
Suspendre temporairement votre application, si vous avez besoin de faire une mise à jour
impactante pour les utilisateurs

$ symfony enable <application_name> <environment_name>
Ré-active l'application et vide le cache

$ symfony clear-controllers
Nettoie le répertoire web/ en supprimant tout les contrôleurs autres que celui de
l'environnement de production. Si vous n'incluez pas ces contrôleurs dans votre fichier
rsync_exclude.txt, cette commande fera le ménage afin de ne pas laisser de failles laissant
apparaître le fonctionnement interne de votre application.

$ symfony fix-perms
Corrige les permissions des répertoires log/ et cache/ pour 0777
(ces répertoires ont besoin d'être éditables pour que le framework fonctionne correctement)

$ symfony log-purge
Supprime les fichiers de log symfony pour les environnements dont la purge est spécifiée à
purge: on (valeur par défaut)
    
```