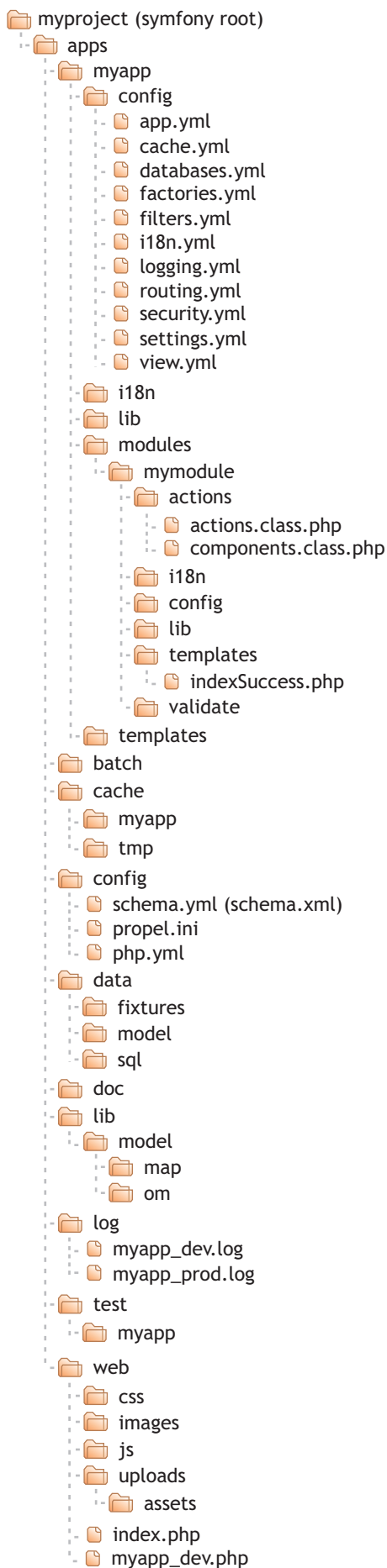


## Struttura delle cartelle e CLI

### Struttura predefinita delle cartelle



### Interfaccia a linea di comando (CLI)

```

$ symfony -T
Visualizza la lista delle operazioni di amministrazione disponibili

$ symfony -V
Visualizza la versione di symfony installata

$ symfony clear-cache <application_name> [template|config]
Svuota le informazioni nella cache (scorciatoia: cc)

$ symfony init-project <project_name>
Inizializza un progetto e genera i file e le cartelle di base per l'esecuzione

$ symfony init-app <application_name>
Per inizializzare una applicazione. I nomi di alcuni file php corrispondono al nome del front controller e sono creati anche nella cartella web/ del progetto.
index.php (prod) and myapp_dev.php (dev)

$ symfony init-module <application_name> <module_name>
Per inizializzare un modulo. Dopo l'esecuzione del comando il nuovo modulo è pronto per essere usato: http://myapp.example.com/index.php/mymodule

$ symfony propel-build-schema [xml]
Genera la rappresentazione schema.yml partendo da un database esistente.
Per generare schema.xml usare l'opzione xml

$ symfony propel-build-model
Genera le classi PHP per il modello in accordo con il modello di dati descritto nello schema.yml.
Le classi base per l'accesso sono automaticamente create nella cartella myproject/lib/model/om/:
BaseArticle.php      BaseComment.php
BaseArticlePeer.php  BaseCommentPeer.php
Inoltre vengono create le classi di accesso ai dati nella cartella myproject/lib/model:
Article.php          Comment.php
ArticlePeer.php      CommentPeer.php

$ symfony propel-generate-crud <application_name> <module_name> <ClassName>
Scaffolding - Genera un nuovo modulo Propel CRUD basandosi su una classe del modello

$ symfony propel-build-sql
Crea il codice SQL per creare le tabelle descritte nel schema.yml. Il file generato è
myproject/data/sql/lib.model.schema.sql.

$ symfony propel-build-db
Crea un database vuoto

$ symfony propel-insert-sql
Esegui il codice SQL myproject/data/sql/lib.model.schema.sql nel database

$ symfony sync <environment_name> [go]
Sincronizza il progetto attuale con un'altra macchina

$ symfony propel-init-admin <application_name> <module_name> <ClassName>
Inizializza un nuovo modulo di amministrazione Propel basandosi su una classe del modello

$ symfony test <application_name>
Esegui l'insieme di test per una applicazione

$ symfony plugin-install [local|global] <channel_name>/<plugin_name>
Installa una nuova plugin

$ symfony freeze
Converti il progetto in una applicazione indipendente (stand-alone)

$ symfony unfreeze
Riprot il progetto al suo stato iniziale (applicazione non stand-alone). Questo cancella il contenuto di data/symfony/, lib/symfony/ e web/sf/

$ symfony disable <application_name> <environment_name>
Disabilita momentaneamente l'applicazione. Per esempio quando c'è la necessità di aggiornare una libreria o una grande quantità di dati

$ symfony enable <application_name> <environment_name>
Riabilita l'applicazione e svuota la cache

$ symfony clear-controllers
Cancella la cartella web/ di tutti i controller che non servono in un ambiente di produzione. Se non si è messo il controller di sviluppo nel file rsync_exclude.txt, questo comando garantisce che non siano presenti backdoor che rivelino la struttura interna dell'applicazione

$ symfony fix-perms
Imposta i permessi delle cartelle, cambia i permessi di log/ e cache/ a 0777 (queste cartelle devono essere scrivibili affinché il framework funzioni correttamente)

$ symfony log-purge
Cancella i file di log dell'applicazione e dell'ambiente dove il file logging.yml lo specifichi (che è il valore di default)
    
```