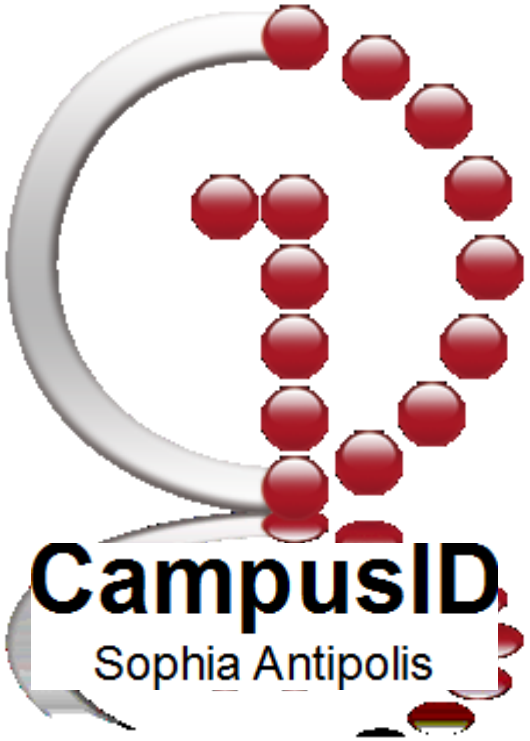
Software PHP

Framework Symfony 3

PARTIE 4 – Mise en production

B2/B3



Version 1.0

Crédits et références bibliographiques

- Site web de Symfony
- Site web de Sensiolabs
- Livre de référence : Développez votre site web avec le framework Symfony3

Plan

- PARTIE 4 – Mettre son site en production
  - Avant la mise en ligne
  - Installation d'une console
  - Configuration serveur
  - Fichier .htaccess

PARTIE 4 – Mettre son site en production

Avant la mise en ligne

- Avant de mettre le site en ligne il faut vérifier les points suivants :
  - Que le site fonctionne en local
  - Mettre à jour vos dépendances via Composer
  - Mettre à jour votre base de données
  - Installer une console accessible depuis le navigateur

- Une fois ces points vérifiés on peut vider les caches puis transférer le code par FTP sur le serveur distant

Installation d'une console pour la navigateur

- Tous les hébergeurs n'offrent pas d'accès SSH d'où l'intérêt d'installer un console accessible depuis le navigateur
- Il en existe plusieurs, on va choisir CoreSphereConsoleBundle
- On commence par ajouter dans le fichier `composer.json` les lignes suivantes :

```
"require": {
…,
            "coresphere/console-bundle": "dev-master"
 },
```

- Puis on exécute `php ../composer.phar update`

Enregistrement du bundle console dans le Kernel

- Il faut ensuite enregistrer le bundle CoreSphereConsoleBundle dans `app/AppKernel.php` dans la partie `dev` :

```
if (in_array($this->getEnvironment(), ['dev', 'test'], true)) {
  $bundles[] = new Symfony\Bundle\DebugBundle\DebugBundle();
  $bundles[] = new Symfony\Bundle\WebProfilerBundle\WebProfilerBundle();
  $bundles[] = new Sensio\Bundle\DistributionBundle\SensioDistributionBundle();
  $bundles[] = new Sensio\Bundle\GeneratorBundle\SensioGeneratorBundle();
  $bundles[] = new CoreSphere\ConsoleBundle\CoreSphereConsoleBundle();
}
```

Enregistrement des routes pour la console

- Il faut ajouter ces lignes au fichier `app/config/routing_dev.yml :`

```
coresphere_console:
    resource: .
    type: extra
```

- Puis finir l'installation en exécutant cette commande :

```
php bin/console assets:install web
```

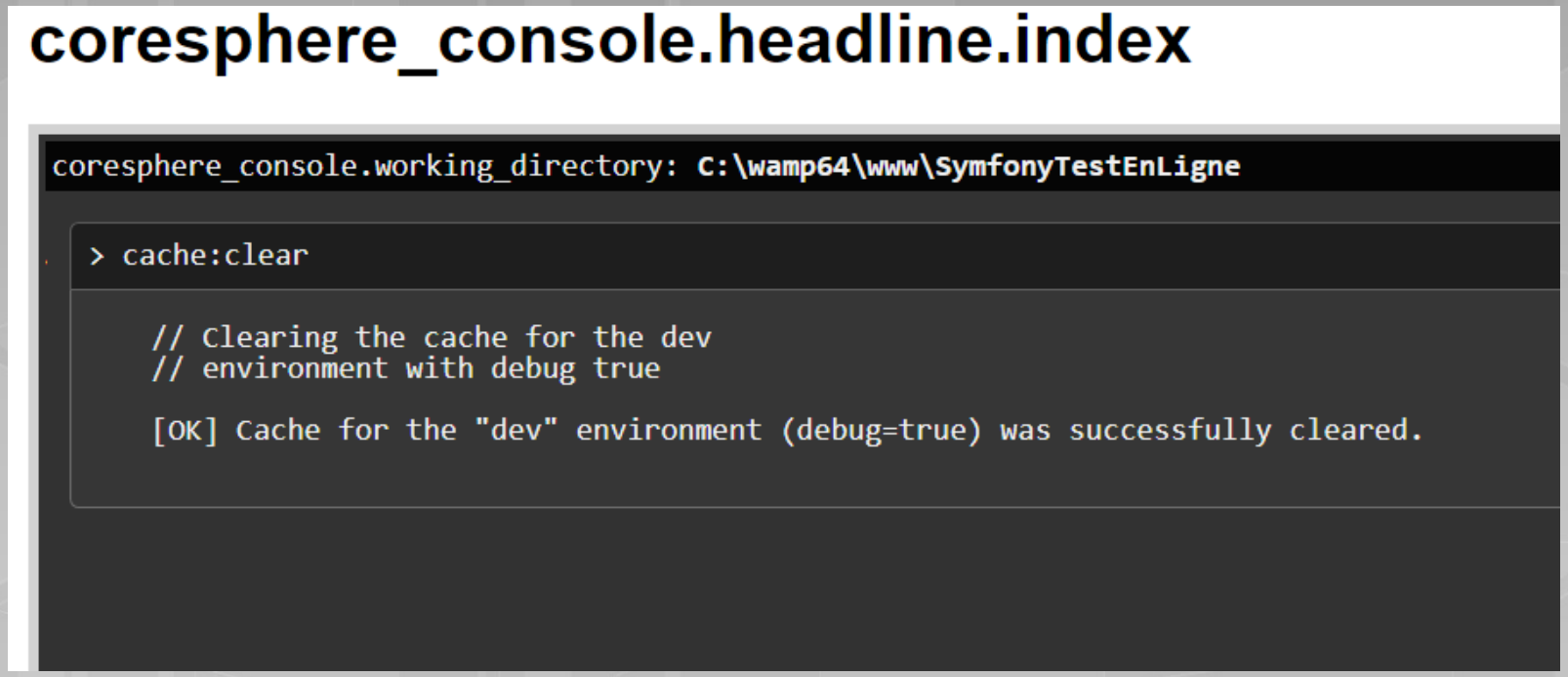
Accéder à la console dans le navigateur

- On accède à la console via l'URL :

  http://localhost/Symfony/web/app_dev.php/_console

Utiliser la console

- On tape directement les commande i.e on ne met plus `php bin/console`
- Exemple : `cache:clear`



- Note : `.clear` pour effacer l'écran

Tester l'environnement de production

- Pour tester l'environnement de production on utilise le contrôleur frontal `app.php` pour lequel on peut activer le débuggeur si besoin :

```
<?php
// web/app.php

// …

$kernel = new AppKernel('prod', true); // Définissez ce 2e argument à true
```

- Remettre ce paramètre à `false` une fois les tests validés

Vider les caches

- Il faut veiller à vider les caches de l'environnement de développement et de production :

```
cache:clear

cache:clear --env=prod
```

Vérifier la compatibilité du serveur

- Voici les **points obligatoires** qu'il faut que votre serveur respecte pour pouvoir **faire tourner Symfony** :
  - La version de PHP doit être **supérieure** ou égale à **PHP 5.5.9**
  - L'extension **SQLite 3** doit être activée
  - L'extension **JSON** doit être activée (par défaut dans PHP)
  - L'extension **Ctype** doit être activée (par défaut dans PHP)
  - Le paramètre `date.timezone` doit être défini dans le `php.ini`

Test de la compatibilité du serveur

- Modifiez le fichier `web/config.php`

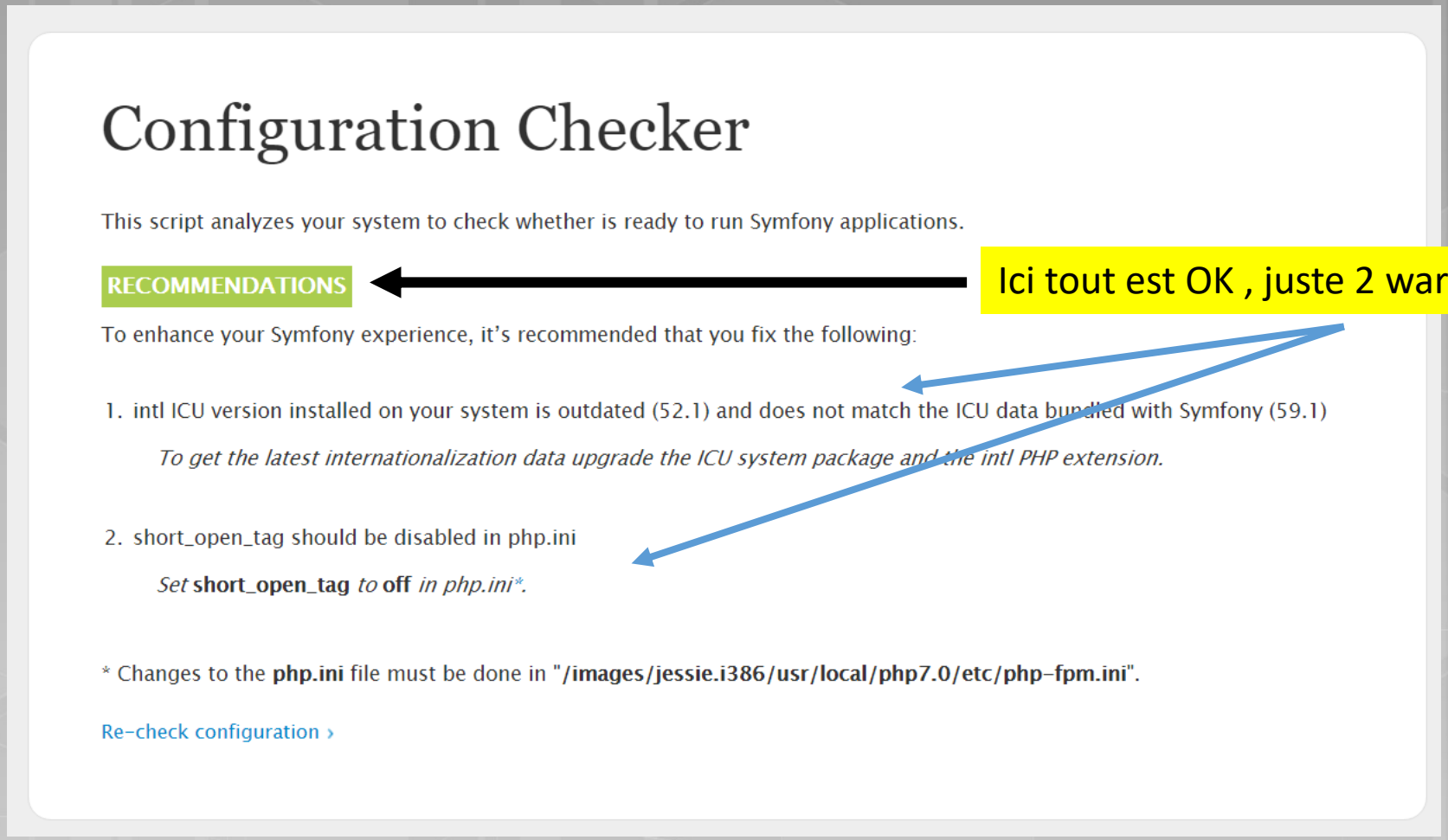
```
<?php
// web/config.php

// …

if (!in_array(@$_SERVER['REMOTE_ADDR'], array(
    '127.0.0.1',
    '::1',
))) {
    header('HTTP/1.0 403 Forbidden');
    exit('This script is only accessible from localhost.');
}
```

A supprimer

- Puis transférez l'ensemble de vos fichiers à la racine de l'hébergement

Test de la compatibilité du serveur

- Enfin exécutez le sur le serveur distant :



Configuration du serveur Apache

- Si vous avez accès à la configuration du serveur Apache vous pouvez ajouter un VirtualHost
- Exemple en local (à adapter pour un site en ligne) :

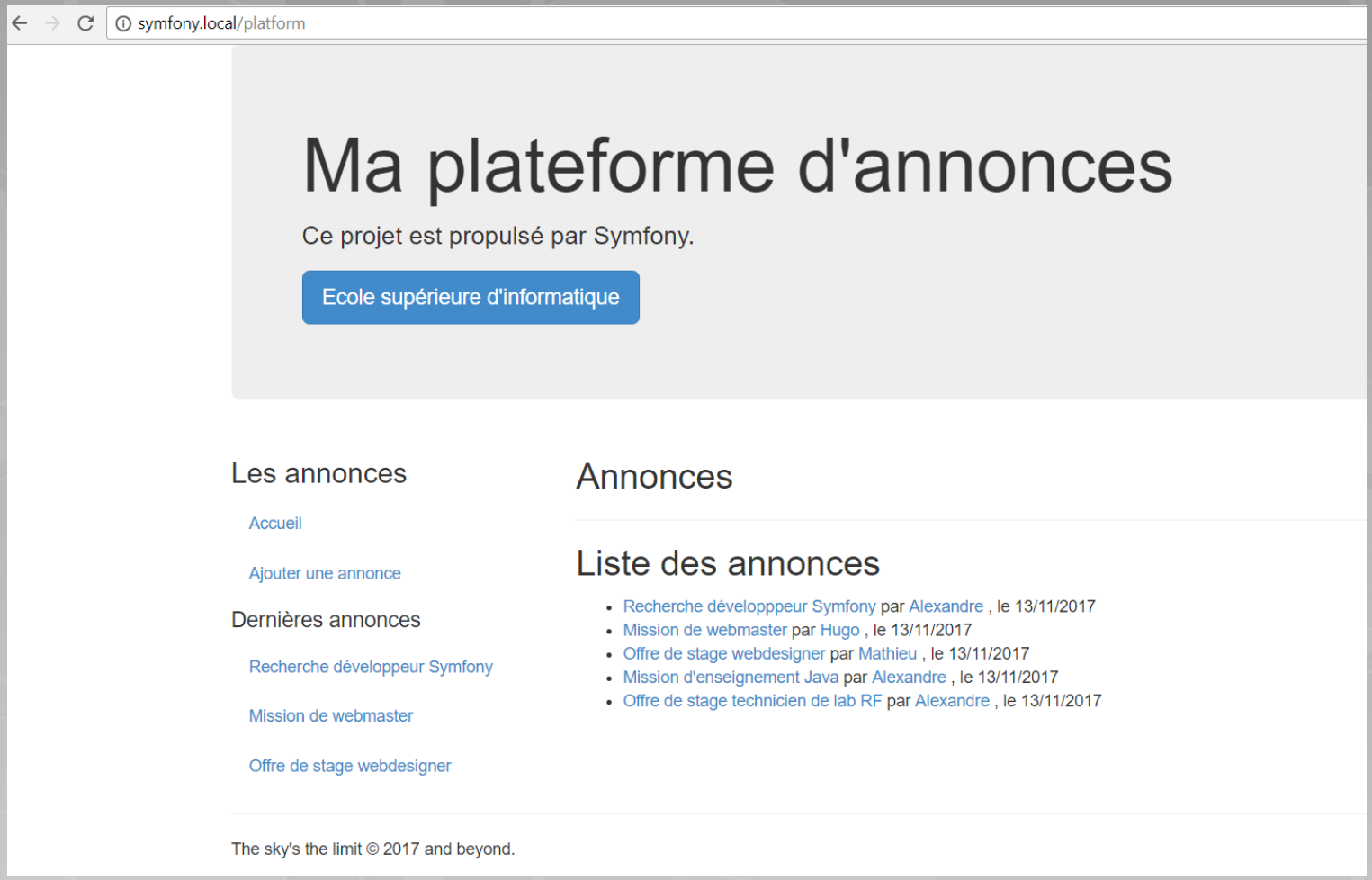
```
<VirtualHost *:80>
        ServerName localhost
        DocumentRoot c:/wamp64/www/Symfony/web
        <Directory  "c:/wamp64/www/Symfony/web">
            DirectoryIndex app.php
                Options -Indexes
                AllowOverride All
                Allow from All
        </Directory>
</VirtualHost>
```

- Fichier C:\Windows\System32\drivers\etc\hosts :

```
127.0.0.1 symfony.local
```

Test de la configuration du serveur Apache

- Avec l'URL suivante : http://symfony.local/platform



URL rewriting

- Si vous n'avez pas accès à la configuration du serveur il faut passer par la méthode de ré-écriture d'URL en modifiant le fichier .htaccess qui se trouve à la racine du site :

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ http://<url_du_site>/web/app.php/$1 [QSA,L]
</IfModule>
```