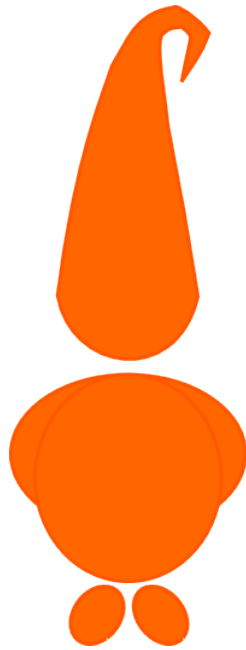




Code Lutin
Conseil & développement
Environnement libre



Code Lutin

Mon Jardinier Online

Installation de la plateforme



SOMMAIRE

1. Objet du document.....	4
2. Vue d'ensemble.....	5
2.1. Front.....	5
2.2. Backend.....	5
2.3. Base de données.....	5
3. Installation.....	6
3.1. Prérequis.....	6
3.2. Installation.....	6
3.2.1. Base.....	6
3.2.2. Apache.....	6
3.2.3. Tomcat.....	7
3.2.4. Postgresql.....	7
3.2.5. SMTP.....	8
4. Configuration.....	9
4.1. Fichiers de configuration.....	9
4.2. Gestion des logs.....	12
4.3. Gestion des images.....	12
4.4. Gestion du catalogue.....	12
4.5. Gestion des pages.....	12
4.6. Gestion des textes.....	12



Synthèse du document

Désignation	Manuel-Installation.doc
Objet	Manuel d'Installation Mon Jardinier Online
Rédacteurs	Julien Ruchaud
Validateurs	Jean Couteau

Suivi des évolutions du document

Version	Date	Modifications apportées	État
1.0	30/01/17	Initialisation du document	Validé

Documents de référence

Désignation	Version	Date



1. Objet du document

Le site développé pour MonJardinierOnLine, est un site grand public. Il demande une haute disponibilité et la possibilité d'évoluer sur différentes plateformes (CDN...).

Le document décrit les prérequis et la procédure d'installation pour le déploiement du site dans un environnement Debian (Linux). Il comporte également quelques procédures d'administration technique.

2. Vue d'ensemble

Le site est composé de trois éléments.

2.1. Front

Le front est la partie visible à l'utilisateur. Dans le cas de mjol il est composé de trois sites de type « single page » :

- ↘ un site pour l'administrateur : il sert à effectuer le suivi des clients, prestataires... et l'administration du site ;
- ↘ un pour les clients : il sert à la prise de commande ;
- ↘ un pour les prestataires : il sert à la gestion des agendas et des interventions.

2.2. Backend

Le backend est la partie serveur qui gère les règles métiers. Il est commun à tous les sites du front. Le front communique avec le backend en appel AJAX. Le backend s'occupe entre autres de l'envoi des emails et des SMS.

2.3. Base de données

La base de données permet le stockage de l'ensemble des éléments de façon cohérente par le biais des transactions.

Il est conseillé de mettre en place un serveur dédié pour la base de données. En cas de montée en charge importante, il faut prévoir un système de réplication master-master.



3. Installation

3.1. Prérequis

Voici les prérequis permettant le bon fonctionnement du site :

- ↘ Distribution Debian stable
- ↘ Java 8
- ↘ Tomcat 8.5
- ↘ Apache 2.4
- ↘ Postgresql 9.5
- ↘ Serveur SMTP (optionnel mais conseillé)

Il existe une image Docker utilisée pour le déploiement du site sur une instance de démonstration Code Lutin. Elle peut être utilisée comme référence pour l'installation.

3.2. Installation

3.2.1. Base

L'installation commence sur la base d'une distribution Debian stable vierge.

Il faut commencer par installer l'ensemble des paquets Debian pour java, tomcat, apache et postgresql avant de poursuivre.

3.2.2. Apache

Pour finir l'installation d'apache :

1. Il faut activer les modules proxy_ajp et rewrite :

```
$ a2enmod proxy_ajp
$ a2enmod rewrite
```

2. Copier le contenu du zip du front dans le répertoire /var/www/html
3. Créer le répertoire /var/www/html/pages
4. Ajouter dans le fichier /etc/apache2/conf-enabled/mjol.conf le contenu suivant :



```
<Location /backend/>
    ProxyPass "ajp://localhost:8009/"
    ProxyPassReverse "ajp://localhost:8009/"
</Location>

<Directory "/var/www/html/">
    RewriteEngine On
    RewriteCond /var/www/html/%{REQUEST_URI} -f
    RewriteRule ^(.+) $1 [L]

    RewriteRule "^admin$" "/admin.html" [L]
    RewriteRule "^admin/.*$" "/admin.html" [L]
    RewriteRule "^gardener$" "/gardener.html" [L]
    RewriteRule "^gardener/.*$" "/gardener.html" [L]
    RewriteRule "^[^.]*$" "/index.html" [L]
</Directory>
```

Le site doit fonctionner en HTTPS uniquement, il faut configurer apache pour les connexions SSL avec un certificat et prévoir une redirection du site pour aller sur le https.

Il faut aussi prévoir un filtrage sur IP pour l'accès au url en /admin.

3.2.3. Tomcat

Pour finir l'installation de tomcat, il faut récupérer le contenu du war pour le backend dans le répertoire /usr/local/tomcat/webapps/ROOT.

3.2.4. Postgresql

Pour finir l'installation de postgresql, il faut

1. Créer un utilisateur

```
$ createuser mjol -W
```



2. Créer la base de donnée

```
$ createdb -0 mjol mjol
```

3.

```
$ psql mjol < ddl.sql
```

Injecter le script sql contenant la définition de la base de donnée

3.2.5. SMTP

Nous préconisons l'installation d'un serveur smtp de type postfix, pour garantir une haute disponibilité de l'envoi des mails. De plus il faut configurer le serveur de mail pour recevoir et rediriger les erreurs d'envoi.

Une utilisation d'un serveur SMTP existant (par exemple SMTP OVH) est possible via la configuration (cf. 4.1 Fichiers de configuration).



4. Configuration

4.1. Fichiers de configuration

Pour le backend, il se trouve dans le répertoire `/usr/local/tomcat/webapps/ROOT/WEB-INF/classes/mjol.properties`. Il contient les propriétés suivantes :

- `mjol.frontend.url`, pour l'url du frontend
- `mjol.callback.url`, pour l'url du backend
- `mjol.pages.dir`, pour le répertoire des pages publiés
- `mjol.datasource.url`, pour l'url jdbc de la base de données
- `mjol.datasource.user`, pour l'utilisateur de la base de données
- `mjol.datasource.password`, pour le mot de passe de l'utilisateur de la base de données
- `mjol.smtp.server`, pour l'adresse du serveur smtp
- `mjol.smtp.port`, pour le port du serveur smtp
- `mjol.admin.email`, pour l'email de contact
- `mjol.secret`, pour la clé secrete pour le cryptage du cookie
- `mjol.search`, pour le rayon des recherches pour le prestataire en kilomètre
- `mjol.marge.freelancer`, pour la marge pour l'auto-entrepreneur
- `mjol.marge.company`, pour la marge pour la société
- `mjol.paid.skip`, pour sauter le paiement
- `mjol.paid.site`, pour le site paybox
- `mjol.paid.rang`, pour le rang paybox
- `mjol.paid.identifiant`, pour l'identifiant paybox
- `mjol.paid.hmac`, pour le hmac paybox
- `mjol.paid.url`, pour l'url paybox
- `mjol.sms.skip`, pour activer les sms
- `mjol.sms.service.name`, pour le nom du service sms sur ovh
- `mjol.sms.application.key`, pour la clé de l'application du service sms sur ovh
- `mjol.sms.application.secret`, pour la clé secrete du service sms sur ovh



Mon jardinier Online

Installation de la plateforme

- `mjol.sms.consumer.key`, pour la clé consumer du service sms sur ovh

Ne pas oublier à redémarrer le serveur tomcat après modification.

Exemple :

Pour le frontend, il se trouve dans le répertoire `/var/www/html/conf.js`. Il contient les propriétés suivantes :

▣ `window.BACKEND_URL`, pour l'url vers le backend.

Exemple :

```
window.BACKEND_URL = "http://192.168.99.149:8084/backend/api";
```

4.2. Gestion des logs

Il faut suivre l'ensemble des logs venant de tomcat pour détecter d'éventuelles erreurs.

4.3. Gestion des images

Les images sont disponibles dans le répertoire suivant `/var/www/html/img`, pour les modifier il suffit de remplacer une image par la nouvelle (les images doivent être nommées de manière identique).

4.4. Gestion du catalogue

Actuellement il n'est pas possible de rajouter ou modifier le catalogue ainsi que les formules de calcul pour les prix et les temps.

4.5. Gestion des pages

Les pages publiées sont disponibles dans le répertoire `/var/www/html/pages`.

4.6. Gestion des textes

Les textes sont contenu en fin de fichier dans les trois fichiers `admin.js`, `site.js` et `gardener.js` dans le répertoire `/var/www/html/`, pour modifier un texte, il suffit de changer le contenu du fichier en respectant la syntaxe JSON et JS.