

# Manuel d'installation de e-colle

e-colle est un gestionnaire de colles destiné aux enseignants de CPGE, à leurs élèves, leurs colleurs ainsi qu'à l'administration pour le décompte des heures effectuées. Son installation ne nécessite que des logiciels libres.

Toute l'installation et la configuration est décrite sur un système d'exploitation de type GNU/Linux Debian (testé précisément sur Ubuntu 14.04).

On donne à la fin les liens vers les documentations des différents logiciels utilisés.

## Table des matières

|   |    |
|---|----|
| 1.Prérequis logiciels.....                                | 1  |
| 1.Python version > 3.4.....                               | 1  |
| 2.ImageMagick.....  | 2  |
| 3.Apache2.....  | 2  |
| 4.Le SGBD (Système de Gestion de Bases de Données).....   | 2  |
| 2.Configuration de e-colle.....                           | 4  |
| 1.Copie des fichiers et configuration de settings.py..... | 4  |
| 2.Création de la base de données.....                     | 6  |
| 3.Configuration de la base de données.....                | 7  |
| 4.Configuration du site dans Apache2.....                 | 7  |
| 3.Création de l'admin et du secrétariat.....              | 11 |
| 4.Personnalisation de e-colle.....                        | 11 |
| 5.Exécution en local (développement).....                 | 11 |
| 6.Sauvegardes de la base de données.....                  | 12 |
| 7.Quelques liens utiles.....                              | 12 |

## 1. Prérequis logiciels

On dénombre ici la liste des logiciels nécessaires à l'installation de e-colle.

### 1. Python version > 3.4

Déjà installé par défaut en théorie, mais si ce n'est pas le cas, dans un terminal :

```
sudo apt-get install python3
```

Il est recommandé d'installer aussi le paquet python3-dev :

```
sudo apt-get install python3-dev
```

Voici la liste des bibliothèques python3 nécessaires :

- django version  $\geq 1.9$  (pour faire des applications web dynamiques)
- reportlab (pour créer des fichiers pdf)

- pillow (pour gérer les images)
- la bibliothèque qui permet de communiquer avec la base de donnée, qui va dépendre du logiciel de SGBD que vous allez utiliser : mysqlclient pour MySQL, psycopg2 pour PostgreSQL, pysqlite2 pour SQLite, .....

Pour cela, commencer par installer pip3 (pip pour python3) :

```
sudo apt-get install python3-pip
```

Vous pouvez maintenant installer les bibliothèques python manquantes avec pip :

```
sudo pip3 install django reportlab pillow
```

Si vous désirez installer une version en particulier de django, par exemple la 1.9 (pour laquelle e-colle est écrite à la base), remplacer django par django==1.9 dans la commande précédente.

## 2. ImageMagick

ImageMagick est un logiciel de gestion d'images, que e-colle utilise pour convertir la première page d'un fichier pdf en fichier jpg (il est inutile de l'installer si vous configurez e-colle avec IMAGEMAGICK=False dans settings.py). Pour l'installer :

```
sudo apt-get install imagemagick
```

## 3. Apache2

Apache2 est un logiciel permettant à l'ordinateur de jouer le rôle de serveur. Il est tout-à-fait possible d'utiliser un autre logiciel, mais on décrit ici uniquement le cas du logiciel apache2, qui est le plus utilisé de tous, et qui est libre.

En sus d'Apache2, il faut installer le module mod-wsgi permettant de communiquer avec python3 :

```
sudo apt-get install apache2 libapache2-mod-wsgi-py3
```

Il peut être judicieux d'installer le langage php ainsi que le module d'apache permettant de communiquer avec php, si jamais on veut utiliser une interface graphique pour l'utilisation du logiciel de SGBD (phpmyadmin pour mysql ou phppgadmin pour postgresql)

```
sudo apt-get install php5 libapache2-mod-php5
```

## 4. Le SGBD (Système de Gestion de Bases de Données)

Les SGBD qui s'utilisent naturellement (et se configurent simplement) avec django sont SQLite (libre), MySQL (libre), PostgreSQL(libre), Oracle (payant, avec une version gratuite allégée et limitée à 1 BDD). On peut en utiliser d'autres mais

la compatibilité avec django n'est pas garantie.

- SQLite est le plus léger (1 base de données = 1 fichier), mais aussi le moins performant et le moins sécurisé. Pour l'installer :

```
sudo apt-get install sqlite
```

- MySQL est un SGBD libre très utilisé pour les site web dynamiques relativement simples. Pour l'installer :

```
sudo apt-get install mysql-server
```

Pendant l'installation il vous faudra préciser le mot de passe de l'utilisateur root, qui vous permettra par la suite de vous connecter à MySQL.

Vous pouvez également installer PHPMyAdmin si vous voulez une interface graphique pour gérer vos bases de données MySQL. Pour cela :

```
sudo apt-get install phpmyadmin
```

Vous pouvez alors y accéder en allant à l'adresse «localhost» puis en cliquant sur PHPMyAdmin (sauf si vous avez déjà une configuration particulière d'Apache2).

Si vous ne le voyez pas c'est probablement que le lien symbolique vers /var/www/html/ de phpmyadmin n'a pas été créé, il faut alors le faire à la main :

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

Il est alors assez simple et intuitif de créer et gérer des utilisateurs/bases de données.

- PostgreSQL est le SGBD le plus performant parmi les SGBD libres, même si un peu plus compliqué à gérer/configurer que MySQL. C'est probablement le meilleur choix, parmi tous ceux possibles. Pour l'installer :

```
sudo apt-get install postgresql
```

À l'installation sera créé un super-utilisateur «postgres», sans mot de passe.

Vous pouvez -devez- vous connectez en tant que postgres pour créer de nouveaux utilisateurs. Pour cela :

```
sudo -i -u postgres
```

Si vous voulez alors créer un utilisateur e-colle, tapez ensuite :

```
createuser -P e-colle
```

(Conservez bien le mot de passe)

Vous pouvez alors créer la base de données qui va avec :

```
createdb -O e-colle -E UTF8 e-colle
```

(Le « -O » est bien la lettre O pour « Owner », et non le chiffre 0)

tapez ensuite

```
exit
```

pour ne plus être connecté en tant qu'utilisateur postgres.

Pour information, la commande `dropuser` permet d'effacer un utilisateur et la commande `dropdb` permet d'effacer une base de données. Par exemple, si vous êtes connecté en tant que postgres :

```
dropdb e-colle
```

va effacer la bdd e-colle, et

```
dropuser e-colle
```

va effacer l'utilisateur e-colle.

Par la suite, pour plus de commodité de gestion, vous pouvez installer une interface graphique : PHPpgAdmin. Pour cela :

```
sudo apt-get install phppgadmin
```

Comme phpmyadmin, il est accessible via l'adresse localhost, puis phppgadmin. Attention vous ne pouvez pas vous y connecter en tant qu'utilisateur postgres. C'est pour cela qu'on est obligé de créer au préalable, en ligne de commande d'autres utilisateurs pour s'y connecter (éventuellement un super-utilisateur pour pouvoir tout gérer ensuite depuis phppgadmin)

- Si vous voulez utiliser un autre SGBD (Oracle ou autre), toutes les infos sont disponibles sur le site de django :

<https://docs.djangoproject.com/en/1.9/ref/databases/#using-a-3rd-party-database-backend>

## 2. Configuration de e-colle

Les étapes décrites par la suite sont à faire dans l'ordre où elle sont présentées.

### 1. Copie des fichiers et configuration de settings.py

Décompressez l'archive e-colle dans le répertoire de votre choix, pas nécessairement le répertoire prévu par Apache2 (ou tout autre logiciel pour faire serveur), on pourra faire un lien symbolique par la suite.

Dans la suite on notera CheminVersEColle le chemin absolu vers l'intérieur du répertoire e-colle.

Il faut configurer le fichier settings.py qui se trouve à CheminVersEColle/ecolle/settings.py

Éditez le fichier avec votre éditeur de texte favori. Il faut alors préciser les points suivants :

- `DEFAULT_ADMIN_PASSWD` (le mot de passe par défaut à la création de l'utilisateur Admin).
- `DEFAULT_SECRETARIAT_PASSWD` (le mot de passe par défaut à la création de l'utilisateur Secrétaire).
- `EMAIL_ADMIN` (l'email de l'admin. On peut laisser vide)
- `EMAIL_SECRETARIAT` (l'email du secrétaire. On peut laisser vide)
- `IP_FILTRE_ADMIN`. Mettre `True` si on veut que la partie admin ne soit accessible que depuis certaines adresses IP pour des raisons de sécurité, `False` sinon.
- `IP_FILTRE_ADRESSES`. Dans le cas où `IP_FILTRE_ADMIN` vaut `True`, il faut préciser dans un tuple les adresses IP autorisées sous forme d'expressions régulières. Par exemple `'^127\.\0\.\0\.\1$'` pour l'adresse 127.0.0.1, ou encore `'^192\.\168\.\d{1-3}\.\d{1-3}$'` pour toutes les adresses IP de la forme 192.168.xxx.xxx
- `IMAGEMAGICK`. `True` si on veut utiliser ImageMagick pour convertir en jpg la première page d'un document pdf d'un programme. `False` sinon.
- `ADMINS`. Un tuple de couples (nom,email) des administrateurs du site.
- `ALLOWED_HOSTS`. La liste des noms de domaine autorisés pour e-colle. Démarrer par un point pour autoriser les sous-domaines. Par exemple `'e-colle.fr'` autorisera tous les noms de domaine du type `'xxxxxxxxxx.e-colle.fr'`
- `SECRET_KEY`. Une clé secrète de 50 caractères ASCII. À modifier.
- `DATABASES`. Il faut y préciser plusieurs champs
  - `ENGINE` : le logiciel de SGBD utilisé ('django.db.backends.mysql' pour MySQL, 'django.db.backends.postgresql' pour PostgreSQL, 'django.db.backends.sqlite3' pour SQLite)
  - `NAME` : le nom de la base de données ('e-colle' est mis par défaut)
  - `USER` : l'utilisateur ayant les droits de la base de données précisée par

NAME. ('e-colle' est mis par défaut)

- PASSWORD : le mot de passe de l'utilisateur précisé juste avant.
- HOST : l'adresse IP du serveur de BDD. 'localhost' si c'est en local.

Enfin, si les crédits ECTS sont activés et que l'on veut incruster dans les pdf directement le tampon/signature du proviseur/proviseur adjoint, il suffit de mettre une image proviseur.jpg ou proviseur.png et/ou proviseuradjoint.jpg ou proviseuradjoint.png dans le répertoire resources/.

## **2. Création de la base de données**

- MySQL

Si le SGBD que vous avez choisi est MySQL, le plus simple est de créer l'utilisateur e-colle et la base de donnée e-colle avec l'interface graphique fournie par PHPMyAdmin. Dans votre navigateur internet, allez à localhost/phpmyadmin/, connectez-vous avec le login root et le mot de passe défini lors de l'installation.

Cliquez sur l'onglet User, puis en bas cliquez sur add User.

Entrez e-colle comme nom d'utilisateur et cliquez sur le générateur de mot de passe pour avoir un mot de passe robuste. Copiez ce mot de passe dans le champs PASSWORD de DATABASES dans settings.py.

Cochez la case « Create database with same name and grant all privileges » puis validez (Go tout en bas à droite)

Cliquez ensuite sur la base de donnée e-colle apparue dans la liste de gauche, puis onglet Operations. En bas à droite, partie Collation, sélectionnez dans la liste déroulante utf8\_general\_ci (Il s'agit de l'encodage utilisé pour ordonner / comparer les chaînes de caractères, on choisit donc utf8, insensible à la casse), puis validez.

Votre base de données est prête.

- PostgreSQL

Si le SGBD que vous avez choisi est PostgreSQL, suivez les [instructions données précédemment](#) dans l'installation et la configuration de PostgreSQL pour créer l'utilisateur et la base de données e-colle. Là encore, le mot de passe attribué à l'utilisateur e-colle doit être copié dans le champs PASSWORD de DATABASES dans settings.py. Votre base de données est alors prête.

Il est important de noter que dans le cas de PostgreSQL, il peut être nécessaire d'effectuer régulièrement des VACUUM (une sorte de défragmentation) sur certaines tables, surtout celles qui subissent beaucoup d'effaçages/modifications.

- SQLite

Si le SGBD que vous avez choisi est SQLite, il n'y a rien à faire, puisqu'il n'y a aucune authentification nécessaire pour accéder à la base de données. (La sécurité est donc moins bonne qu'avec MySQL ou PostgreSQL). Il faut d'ailleurs laisser les champs USER, PASSWORD et HOST vides dans DATABASES du fichier settings.py, et mettre le chemin vers le fichier d'enregistrement dans NAME.

### 3. Configuration de la base de données

Dans un terminal, allez dans le répertoire e-colle, puis lancez la commande :

```
python3 manage.py makemigrations accueil
```

Tapez ensuite :

```
python3 manage.py migrate
```

Une fois cette étape finie, toutes les tables de la base de données ont été créées.

### 4. Configuration du site dans Apache2

Il reste à configurer le site dans Apache2.

- Lien vers le répertoire e-colle

Il faut commencer par créer un lien symbolique du répertoire /var/www/html vers le répertoire e-colle. Pour cela :

```
sudo ln -s CheminVersEColle /var/www/html/e-colle
```

où 'CheminVersEColle' est à remplacer par le chemin absolu du répertoire e-colle

- Gestion des droits.

Apache2 doit avoir les droits en lecture et exécution sur tous les fichiers/répertoires de e-colle, en plus des droits en écriture sur le répertoire '/media'.

Si vous voulez aussi avoir accès à ces fichiers sans vous connecter en tant que root, le plus simple est de créer un groupe 'web' (ou autre nom de votre choix), auquel vous, ainsi que l'utilisateur www-data (apache2) serez affilié. Pour cela :

```
sudo groupadd web
```

Une fois le groupe web créé, il faut affilier l'utilisateur www-data ainsi que vous-même au groupe web :

```
sudo adduser vous web
```

où « vous » est à remplacer par votre nom d'utilisateur, puis :

```
sudo adduser www-data web
```

Il faut alors vous rendre vous, ainsi que le groupe web, propriétaires du répertoire e-colle. Rendez-vous dans le répertoire parent de e-colle, puis :

```
sudo chown -R vous:web e-colle
```

où « vous » est à remplacer par votre nom d'utilisateur.

Donnez tous les droits en lecture et en exécution à vous-même et au groupe sur tout le répertoire e-colle. À partir du répertoire parent de e-colle :

```
sudo chmod -R 750 e-colle
```

De cette manière vous avez tous les droits (écriture/lecture/exécution) sur le répertoire et ses descendants, et le reste du groupe web, dont www-data, a tous les droits en lecture/exécution. Il reste à donner les droits en écriture au groupe web sur le sous-répertoire media (celui où on stocke les téléversements) :

```
cd e-colle
```

```
sudo chmod -R 770 media
```

- Gestion du protocole SSL (conseillé)

Si vous voulez crypter l'envoi et la réception des données (si vous ne le faites pas, les mots de passe vont passer en clair sur internet), il faut utiliser le protocole SSL (https), qui combine un cryptage + une certification du site. On va auto-certifier le site (sinon il faut payer un organisme extérieur pour le certifier), donc les navigateurs internet afficheront une erreur de certificat en arrivant dessus.

On commence par activer le mod ssl de apache2 :

```
sudo a2enmod ssl
```

puis on redémarre apache2 :

```
sudo service apache2 restart
```

On installe openssl :

```
sudo apt-get install openssl
```

puis on crée une clé de cryptage et un certificat :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -out /etc/apache2/ecolle.crt -keyout /etc/apache2/ecolle.key
```



(la directive -days 365 est la durée de validité du certificat en jours, vous pouvez indiquer une durée plus longue si vous ne voulez pas avoir à le recréer chaque année)

Vous devez alors répondre à plusieurs questions (Des exemples parlants sont précisés à titre indicatif, à vous d'adapter) :

- Country Name : FR
- State : France
- Locality Name : Cayenne
- Organisation Name : Ma Prepa
- Organisation Unit Name : Ma Prepa
- Common Name : e-colle.mondomaine.org (nom de domaine)
- Email Address : [monadresse@monsite.com](mailto:monadresse@monsite.com) (email de l'Admin)

Pour plus de sécurité, rendez le fichier contenant la clé inaccessible :

```
sudo chmod o-rw /etc/apache2/ecolle.key
```

- Configuration du virtualhost.

Éditez le fichier e-colle.conf (ou e-collessl.conf si vous utilisez le protocole SSL) présent à la racine du répertoire e-colle, et modifiez les directives ServerName et ServerAdmin en conséquence. Sauvegardez. Si vous accédez au serveur via son adresse IP, mettez son adresse IP en ServerName. Sinon mettez un nom de domaine qui pointe vers l'adresse IP de votre routeur/box (voir point suivant : [Obtention d'un nom de domaine](#))

Coupez ensuite le fichier e-colle.conf (ou e-collessl.conf), puis collez-le dans le répertoire /etc/apache2/sites-available (pour le cas du ssl, renommez-le e-colle.conf). Pour le faire en ligne de commande, depuis l'intérieur du répertoire e-colle :

```
sudo mv e-colle.conf /etc/apache2/sites-available/e-colle.conf
```

ou, le cas échéant :

```
sudo mv e-collessl.conf /etc/apache2/sites-available/e-colle.conf
```

Il ne reste plus qu'à activer le site :

```
sudo a2ensite e-colle
```

- Sécurisation de Apache2

Si ce n'est pas déjà fait, il est conseillé de sécuriser un minimum apache2, en particulier, interdire l'accès au serveur (c'est-à-dire l'accès à l'intérieur du répertoire où sont stockés les fichiers des sites) aux adresse ip inconnues

Pour cela il faut éditer les fichiers 000-default.conf (pour le http) et default-ssl.conf (pour le https) se trouvant dans /etc/apache2/sites-available et rajouter avant la balise de fermeture </VirtualHost> les directives suivantes :

```
<Directory /var/www/html>  
<limit GET POST>  
Require ip 127.0.0.1  
</limit>  
</Directory>
```

si vous voulez que seul l'ordinateur serveur ait accès à son répertoire de fichiers des sites. On peut rajouter avant </Limit> d'autres adresses ou plages d'adresses, par exemple 192.168 pour toutes les adresses IP commençant par 192.168. (qui sont forcément locales)

Le <limit GET POST> est là pour interdire les protocoles http autres que GET et POST (en particulier bloquer le protocole PUT, peu utilisé, mais qui peut poser des problèmes)

Si vous voulez rendre le répertoire /var/www/html totalement inaccessible via le protocole http ou https, remplacer Require 127.0.0.1 par Require all denied.

- Obtention d'un nom de domaine

Si vous disposez d'une adresse IP fixe (établissement scolaire par exemple) vous pouvez vous dispenser d'avoir un nom de domaine et accéder directement à e-colle via l'adresse IP du serveur. Si vous l'installez depuis une adresse IP flottante (type box) un nom de domaine est obligatoire pour pouvoir suivre votre adresse IP. De nombreux sites proposent des noms de domaines (ou sous-domaines) gratuitement, avec des fonctionnalités limitées et un nombre de noms de domaines limité (~3). Par exemple no-ip, dyndns, etc.... Rendez-vous donc sur votre moteur de recherche préféré pour l'obtention d'un nom de domaine.

- Configuration de votre box/routeur

Pour rediriger les requêtes entrantes sur votre box/routeur vers l'ordinateur qui fait serveur, il faut aller dans les paramètres de votre routeur, et dans la partie NAT/PAT, il faut rediriger les requêtes entrantes http (port 80) et https (port 443) vers votre serveur (via son adresse IP ou son nom). Il est également conseillé de configurer le pare-feu pour qu'il ne laisse passer que les requêtes entrantes sur les ports 80 et 443 (sauf si vous en utilisez d'autres)

### 3. Création de l'admin et du secrétariat

Allez sur le site via son nom de domaine, rajoutez /init dans la barre d'adresse à la fin du nom de domaine puis validez. Les utilisateurs Admin et Secrétariat sont alors créés avec les valeurs par défaut du fichier settings.py.

Il est recommandé (par soucis de sécurité) d'aller désactiver ensuite cette adresse en éditant le fichier ecolle/urls.py et commenter la ligne 7 puis sauvegarder :

```
url(r'^init$', 'accueil.views.init',name="init")
```

### 4. Personnalisation de e-colle

Vous pouvez bien entendu personnaliser e-colle bien au-delà des quelques paramètres du fichier settings.py. Par exemple tout le CSS est contenu dans un seul fichier (public/css/style.css), donc vous pouvez facilement modifier les couleurs, les images, les polices, etc .... Vous pouvez aussi, pour les pythoniciens, modifier les vues, rajouter vos propres vues dans les fichiers views.py adéquates pour rajouter des fonctionnalités si le cœur vous en dit (pensez, si vous ajoutez une vue, à ajouter le fichier de template qui va avec et l'url qui va avec, dans le fichier urls.py du répertoire concerné).

### 5. Exécution en local (développement)

À des fins de test, vous pouvez exécuter en local e-colle, en passant la variable DEBUG à True dans settings.py, et en lançant la commande

```
python3 manage.py runserver
```

dans le répertoire e-colle. E-colle sera alors accessible à l'adresse localhost:8000. Attention ce serveur ne doit jamais être utilisé autrement qu'en local, de même que la variable DEBUG ne doit valoir True qu'en local.

### 6. Sauvegardes de la base de données

Il est important de faire des sauvegardes régulières de la base de donnée.

Django permet de faire cela de manière transparente vis-à-vis du SGBD, à l'aide de la commande dumpdata. La commande

```
python3 manage.py dumpdata > ecolle.json
```

exécutée dans le répertoire e-colle crée le fichier ecolle.json, qui contient toutes les données de la base de données. Pour les recharger dans la base, il suffit de lancer la commande

```
python3 manage.py loaddata ecolle.json
```

Par soucis de simplicité un script shell est joint (dans le répertoire backup) pour créer des sauvegardes de la base de données. (Il faut l'éditer pour y préciser quelques chemins). S'il est exécuté quotidiennement, il fait une sauvegarde des 7 derniers jours, des 4 derniers samedis et des 1<sup>er</sup> de chaque mois.

L'idéal est de planifier l'exécution de ce fichier dans la crontab.

```
crontab -e
```

pour ouvrir la crontab, puis on ajoute la ligne :

```
0 0 * * * CHEMIN/sqlbackup.sh
```

pour une exécution quotidienne à minuit du script (on remplace CHEMIN par le chemin absolu vers sqlbackup.sh). On sauvegarde ensuite la crontab.

## 7. Quelques liens utiles

- [Lien vers la documentation de Django](#)
- [lien vers la documentation de MySQL](#)
- [lien vers la documentation de PostgreSQL](#)
- [lien vers la documentation de SQLite](#)
- [lien vers la documentation de Apache 2](#)
- [lien vers la documentation de Mathjax](#)