

juin 2022

Mes notes à propos de la création d'une webapp à l'aide de Ruby On Rails. Elles sont découpées en fichiers relativement indépendants car la création d'une application n'est pas un processus linéaire...

Postgresql

Je commence par un apparté ! Rien à voir avec Ruby-On-Rails. Pas directement en tout cas. Mais c'est ma première expérience avec un serveur de database (autre que sqlite dont le fonctionnement ne demande pas de configuration).

Sur archlinux :

```
pacman -S postgresql
```

CAUTION :

Mettre à jour un serveur postgresql est très délicat. Voir archWiki. Le mieux est de désactiver les mises à jour. Éditer (sudo) /etc/pacman.conf et, dans la rubrique [OPTION] ajouter :

```
IgnorePkg = postgresql
IgnorePkg = postgresql-libs
```

Configurer la base de données : l'utilisateur créé doit avoir les droits superuser pour que les tests fonctionnent !

```
sudo chown postgres:postgres /var/lib/postgres
sudo -iu postgres
createuser --interactive mazu
`superuser?y`; `create-db-ability?y`; `create-user?n`
exit
```

back to bash shell for mazu

New rails app

```
rails new my_app --database=postgresql
cd my_app
bin/setup # really ?
```

to install dependencies and create database (Have a look in `config/database.yml` : database name is `my_app_database` by default).

Add `username: db-user-name` (as was created above) for development and test..

Installer un gem

`gem install blabla` installe le gem 'blabla' dans le répertoire adéquat (j'ai une installation locale gérée par `rbenv` , donc ça va dans un sous répertoire de `/home/mazu/.rbenv/` .

Pour le déclarer en tant que dépendance de l'application, il faut ajouter la ligne adéquate dans le fichier `Gemfile` à la racine de l'application et ensuite appeler `bundle install` .

Une commande compacte (add to Gemfile and install). Il s'agit de :

```
bundle add blabla
```

Ne pas hésiter à regarder l'aide. Syntaxe : `gem help command` ou `bundle help command` . Par exemple `bundle help add` .